Master's thesis

NTNU
Norwegian University of Science and Technology

Aksel Andrè Wiik Martinsen

Real time monitoring of hatching process for copepods

Master's thesis in Industriell kybernetikk
Supervisor: Morten Omholt Alver
June 2023

NTNU
Norwegian University of
Science and Technology

Aksel Andrè Wiik Martinsen

Real time monitoring of hatching process for copepods

Master’s thesis in Industriell kybernetikk
Supervisor: Morten Omholt Alver
June 2023

Norwegian University of Science and Technology

NTNU
Norwegian University of
Science and Technology

Acknowledgements

This master's thesis was carried out at the Norwegian University of Science and Technology (NTNU) in Trondheim as a part of a two-year master's program in Industrial Cybernetics at the Department of Engineering Cybernetics. The thesis was written in cooperation with a company called CFEED and with Morten Omholt Alver as a supervisor. I thank Morten for his valuable guidance, support, and input during the work. Maren R. Gagnat, from CFEED, has given me valuable insight into the hatching process of copepods and helped me set up the experiments. Bjarne Scholz, Sondre Kristenstuen and Raja Mansingh Rathore, also from CFEED, have been helping me at the production facilities and supported me with performing the experiments. I want to thank CFEED and their employees for welcoming me to their facilities in Vanvikan.

The work was done during the spring of 2023 and is not a continuation of the project work performed in the autumn of 2022. It has been a steep learning curve doing this project, as I needed to gain prior knowledge about machine vision and limited knowledge about machine learning before starting the work at the end of January. ChatGPT has been used for grammar and spelling, and no factual information has been provided from the service.

Executive summary

The company CFEED produces copepod eggs of the species *Acartia tonsa* for use as live feed in the production of marine species. Monitoring the hatching rate is essential to the production and quality assurance of the copepod eggs. Today, this is performed with manual counting tests, which are time-consuming and physically straining for the human observer, but also exposed to inaccuracies due to factors such as observer patience and experience in deciding the distinct stages of *Acartia tonsa* development. In this thesis, a sensor will be investigated to monitor the hatching process using machine vision and machine learning to classify the proportion of eggs and hatched copepods at the nauplii stage.

A monochrome camera was employed for image acquisition, capturing high-resolution images of the copepods. Annotation and pre-processing of the images were conducted to generate datasets. The generated datasets were used to train VGG networks and evaluate the performance of the trained network. An accuracy of 99% was obtained on a validation set during training. The trained network struggles to classify images from other distributions, which is necessary for a hatching sensor to monitor a hatching process.

The research has highlighted the importance of dataset quality for the performance of the trained model, which factors including the quality of the annotations and the variability in the images can influence. Several variables related to the quality of the images, hatching tank design, and camera setup have influenced this project's results. Several challenges were identified and need to be addressed in future studies, such as investigations related to depth of field, handling of image noise, network architecture, and hyperparameter tuning.

In conclusion, the test setup of the hatching sensor can classify with satisfactory accuracy. The setup needs further improvement to make a functional prototype.

Sammendrag

Selskapet CFEED produserer copepod-egg av arten *Acartia tonsa* for bruk som levende fôr i produksjonen av marine arter. En vesentlig del av både produksjon og kvalitetssikring av copepod-egg er å overvåke klekkeprosessen. I dag utføres dette med manuelle telleprøver, som er tidkrevende og fysisk belastende for mennesket som utfører den. Telleprøvene er utsatt for unøyaktigheter grunnet faktorer som tellerens tålmodighet og erfaringsnivå med å skille forskjellige stadier av utviklingen til copepodene. I denne oppgaven vil en sensor bli undersøkt for å overvåke klekkeprosessen ved bruk av maskinsyn og maskinlæring for å måle andelen av egg og klekkede copepoder i nauplii-stadiet.

Et monokromt kamera ble benyttet for bildeinnhenting for å ta høyoppløselige bilder av copepodene. Annotering og forbehandling ble utført for å generere forskjellige datasett. De genererte datasettene ble brukt for å trene et VGG-nettverk og evaluere ytelsen til det trente nettverket. En nøyaktighet på 99% ble oppnådd på et valideringssett under treningen. Det trente nettverket sliter med å klassifisere bilder fra andre distribusjoner, noe som er nødvendig for å tilfredsstille kravene til en klekkesensor for å overvåke en klekkeprosess.

Arbeidet i denne oppgaven har fremhevet betydningen av kvaliteten til datasettene for den opptrente modellens ytelse, noe som påvirkes av ulike faktorer, inkludert kvaliteten på annotasjonene og variabiliteten i bildene. Det er flere variabler relatert til bildekvaliteten. Design av klekketanken og kameraoppsettet har påvirket resultatene i dette prosjektet. Flere utfordringer ble identifisert og må adresseres i fremtidige studier, som undersøkelser relatert til dybdeskarphet, håndtering av bildestøy, og justering av nettverksstruktur og hyperparametre.

Konklusjonen er at testoppsettet av klekkesensoren klarer å klassifisere med tilfredsstillende treffsikkerhet. Det er behov for ytterligere forbedringer for å lage en funksjonell prototype.

Table of Contents

List of Tables

List of Figures

Abbreviations

Abbreviation	Description
AI	Artificial Intelligence
ANN	Artificial Neural Networks
CNN	Convolutional Neural Networks
FPS	Frames Per Second
Gigabit Ethernet	GigE
ML	Machine Learning
NN	Neural Networks
mL	mili-litre
RGB image	*Red, green, blue* image
Rectified Linear Unit	ReLu
Visual Geometry Group	VGG

1

Introduction

1.1 Background

Copepods play a significant role in aquaculture as a healthier and more optimal feed source for marine species, particularly during the larval rearing phase. Studies have demonstrated that feeding cod larvae with intensively cultivated copepod nauplii, specifically *Acartia tonsa*, for the first 28 days post-hatching leads to improved survival, growth, and larval quality compared to using rotifers as a food source (Hansen, 2011). Consequently, copepods have become a valuable resource in the aquaculture industry and have been commercially produced by companies such as CFEED. An essential aspect of production and quality assurance of the eggs is to monitor the hatching rate and percentage. This thesis is written in collaboration with CFEED. The input on the challenges related to the hatching process is from conversations and e-mails with employees from CFEED.

The production methods of copepod eggs are labour-intensive and prone to human error. The process of breeding copepods involves manually conducting daily hatching tests, which require examining samples under a microscope to count and identify the developmental stages of 200 to 250 individual copepods. This task is time-consuming and physically straining for the human observer and exposed to inaccuracies due to the observer's patience and experience in deciding the distinct stages of *Acartia tonsa* development.

Maintaining an overview of the hatching process is essential. The hatching rate and percentage obtained from manual counting samples today are crucial for understanding production volumes and inventory levels and ensuring the quality of the copepods delivered to the customers. To address these challenges, this master thesis presents a hatching sensor, including a novel approach that utilises a Visual Geometry Group (VGG) (Simonyan and Zisserman, 2014) neural network for the classification of eggs, copepods and undesired items such as air bubbles. A monochrome camera was utilised to obtain high-resolution images of the copepods. These images were improved using image pre-processing methods. This study investigates using a machine learning (ML) network trained on the provided images. This approach seeks to reduce the need for manual labour, increase operational efficiency and minimise human error in the copepod classification process.

1.2 Problem description

1.2.1 Hatching-sensor

This master thesis aims to develop a concept for a hatching sensor that employs a monochrome camera from FLIR to monitor the hatching process in producing copepod eggs. The focus will be on utilising a light source, camera settings, and optics to ensure high-quality image acquisition for ML classification.

- **Camera settings:** To obtain images suitable for machine vision analysis, the camera settings must be adjusted to account for factors such as exposure and aperture. These settings will be tuned to get sharp images and minimise noise, allowing for further image processing and use in a ML network.
- **Light source:** The selection of an appropriate light source is crucial for capturing sharp images. A light source should provide consistent and uniform illumination to minimise shadows and reflections, ensuring the visibility of the particles during the hatching process.
- **Image acquisition and labelling:** Images of the hatching process will be acquired using a tuned camera to capture images at various stages of the hatching process. The acquired images will be labelled and used to train and validate the ML algorithm.

1.2.2 Machine vision algorithm

- **Image processing:** The images will be processed to facilitate use in a neural network. It includes techniques used in digital image processing. The images will be labelled according to the developmental stage of the copepod, providing ground truth for the algorithm to be trained on.
- **Train a machine learning algorithm:** A VGG network will be trained with the processed images. The training process will involve adjusting hyperparameters such as the number of learning rates, epochs and batch size to achieve the best possible accuracy. The trained model will be able to classify between egg and nauplii stages, providing real-time monitoring of the hatching rate and percentage.

1.3 Structure of the report

The report is organised as follows:

- In chapter 2, relevant theoretical background is presented.
- In chapter 3, the hardware of the hatching-sensor is described.

- In chapter 4, the building-blocks of the machine vision algorithm is presented.
- In chapter 5, various experiments and tests of the image acquiring and machine vision algorithm are presented.
- In chapter 6, the project results are discussed and further work is presented.
- In chapter 7, a conclusion is made.

2

Theory

This chapter will review and present the relevant theory used in this master thesis.

2.1 Copepods

2.1.1 Acartia Tonsa

Copepods are the natural initial feed for marine organisms and are considered to possess an optimal nutritional profile. This results in increased survival, improved quality, and faster growth compared to traditionally cultivated plankton such as rotifers and Artemia (Hansen, 2011). *Acartia tonsa* is a copepod species used in aquaculture due to its optimal nutritional profile for larval rearing. The life cycle of the copepods consists of several developmental stages. Acartia tonsa undergoes six naupliar stages and five copepodite stages before adulthood, totalling twelve distinct developmental stages (Hansen et al., 2010). Copepod eggs are relatively small, with an average length of 80 micrometres. Upon hatching, the nauplii are, on average, 110 micrometres in size (Leandro et al., 2006). An image of an egg to the left and newly hatched nauplii to the right can be seen in Figure 2.1.

2.1.2 Breeding

Although copepods exhibit superior nutritional values compared to rotifers and artemia, only a few commercial hatcheries utilise them as live feed (Drillet et al., 2011). Cultivation of copepods has presented significant challenges, and wild harvests are subject to unpredictable factors, such as weather conditions, handling, and storage. The species of *Acartia Tonsa* has shown the highest cultivation potential (Drillet et al., 2006) and is mass cultivated by the company CFEED AS. To ensure optimal growth, it is essential to maintain proper water circulation in the production tank (Jepsen, 2014). Aeration should be incorporated in the hatching tanks to prevent sedimentation of the eggs and the copepods

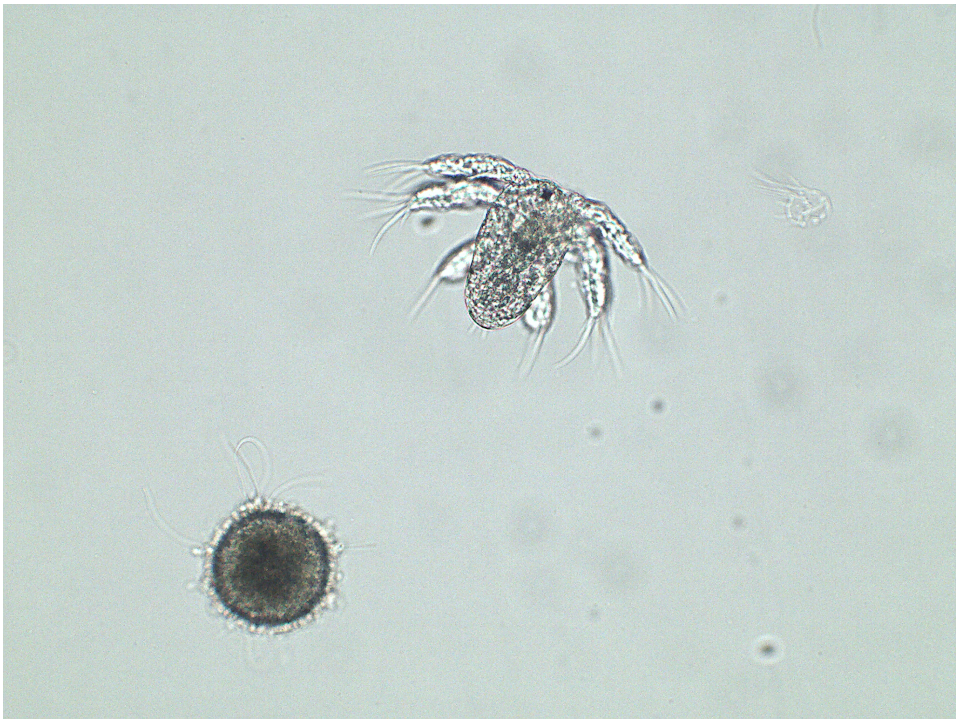


Figure 2.1: An image from a microscope of an egg and a nauplius acquired through a microscope by CFEED.

from clustering. During their life cycle, *Acartia Tonsa* copepods primarily feed on algae, which provide essential nutrients for their growth and development. The growth conditions play a crucial role in their cultivation success (Jepsen, 2014). Maintaining a temperature of 26 °Cenables the completion of the hatching process within 24 hours (Appendix B).

2.2 Photography

The theory in this section is based on Solomon and Breckon (2011) unless other is stated.

2.2.1 Photography and optics

Pixels

A pixel is the smallest unit that makes up a digital image. Pixels are images' building blocks, representing each point's colour and light intensity. The resolution of a digital image is often expressed in terms of the number of pixels along its width and height—an image with a resolution of 2448x2048 results in a total of 5 013 504 pixels. A higher number of pixels results in a more detailed image but requires more storage space and processing power to handle the data. By manipulating the pixels through processing techniques, images can be enhanced and analysed to extract valuable information.

Exposure time

In photography and digital imaging, exposure time, also known as shutter speed, is a parameter that determines the amount of light captured by the camera sensor during image acquisition. Exposure time is typically measured in fractions of a second. Short exposure time ranges from 1/4000th of a second (0.2ms), and long exposure time such as several seconds. The choice of exposure time depends on the lighting conditions, the subject being photographed, and the desired outcome of the image. A short exposure time captures fast-moving objects effectively and reduces the chances of motion blur. Balancing exposure time with other camera settings, such as aperture, is crucial for obtaining desired image quality in different environments.

Aperture

Aperture refers to the opening camera's lens size that allows light to pass through and reach the image sensor. The gap impacts the depth of field, which is the distance range in a scene that appears sharp and in focus. A larger aperture results in a shallower depth of field, meaning that only a small portion of the scene will be in focus, while the rest will appear blurred. Vice versa for smaller apertures making more of the scene appear sharp and in focus. Aperture is typically expressed as an f-number, for instance, f/1.8, f/2.8, f/4 (Potmesil and Chakravarty, 1982). It represents the lens's focal length ratio to the diameter of the aperture opening. A low f-number corresponds to a larger aperture opening, while a high f-number indicates a smaller aperture. By adjusting the aperture, photographers can control the amount of light reaching the sensor and manipulate the depth of field to achieve the desired outcome.

Gain

Gain is an essential parameter in photography that affects the image quality, particularly in low-light conditions or when capturing fast-moving subjects. In photography, gain refers to the amplification factor applied to each pixel value. It can improve the brightness of an image. In dimly lit environments or with fast shutter speeds, the gain can improve the imaging. However, increasing the gain can introduce noise to the image, which can affect the clarity and detail of the captured subjects.

2.2.2 Colour models

The purpose of a colour model (also called colour space or colour system) is to facilitate the specification of colours in some standard, generally accepted way. A colour model defines a coordinate system in which each colour corresponds to a unique point within a specific subspace (Gonzalez et al., 2022). Image classification for specific objects is widely used with a ML algorithm (Pak and Kim, 2017). The experimental findings by Bui et al. (2016) demonstrated that utilising greyscale images for classification led to higher accuracy across various types of classifiers compared to employing RGB images.

Grey-scale

A greyscale image is a visual representation in which each pixel's value corresponds to a single sample, conveying the quantity of light. This type of image encompasses solely the intensity information (Sonka et al., 2013). Greyscale images, categorised as grey monochrome or black-and-white, only consist of diverse shades of grey. Each pixel's brightness or darkness is represented by a discrete numerical value, varying from 0, indicative of black, to 255, representative of white (Johnson, 2006). This spectrum of grey shades allows for a nuanced depiction of image features.

RGB

For RGB colour models, each colour appears in its primary spectral components of red, green and blue. This model is based on a cartesian coordinate system (Gonzalez et al., 2022). By combining the three components with different intensities ranging from 0, (absence of colour) to 255 (maximum intensity of colour) Pratt (2007). Images represented in the RGB colour model consist of three component images, one for each colour. In Figure 2.2 [1], a representation of an RGB image is of the left, with the greyscale equivalence.

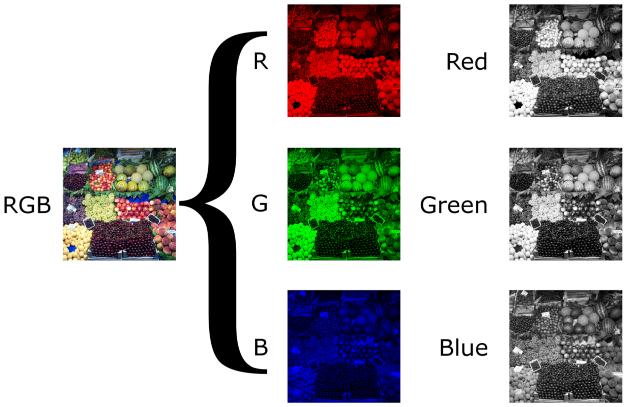


Figure 2.2: Copmosition of RGB from three greyscale images.

[1] Accessed: 10.05.23. Available at: `https://en.wikipedia.org/wiki/Grayscale#/media/File:Beyoglu_4671_tricolor.png`.

2.3 Machine vision and machine learning

2.3.1 Image processing

The theory presented in this section is based on Solomon and Breckon (2011) unless other is stated. Image processing involves manipulating and analysing digital images to improve their quality or extract valuable information. It can be divided into two categories: pre-processing and post-processing. Pre-processing enhances the image quality, while post-processing focuses on extracting relevant information or features from the processed images.

Thresholding

Thresholding is a technique in which a specific threshold value is chosen to categorise pixels into separate regions. Pixels with values greater than the threshold are assigned to one region, while those below the threshold are assigned to another (adjacent) region. This process creates a binary image *b(x, y)* from an intensity image *I(x, y)* based on a criterion:

$$b(x,y) = \begin{cases} 1, & \text{if } I(x,y) > T \\ 0, & \text{otherwise} \end{cases} \tag{2.1}$$

where T represents the threshold value (Solomon and Breckon, 2011). Thresholding is used to segment or separate objects of interest from the background in a digital image. Simplifying the image makes it easier to identify and analyse the objects of interest.

Image subtraction

The method involves subtracting one image's pixel values from another's corresponding pixel values. This operation can result in negative pixel values, which will be clamped to zero. The result of the operation is a third image highlighting the differences between the two input images. By subtracting one image from another, the information about differences becomes visible.

Image segmentation

In computer vision, segmentation is the generic process of subdividing an image into regions or objects. The goal is to localise desired objects or regions and attach labels. Two primary routes are used to detect the objects: edge/boundary methods or region-based methods, either by looking for sharp differences between groups of pixels or assigning pixels to a given region based on their degree of similarity. Blob detection is used for identifying and segmenting distinct regions in an image where a set of pixels differ in properties compared to the surroundings. The *blob* refers to a region where some properties are constant or vary within a predetermined range. It could be a set of pixels in an area connected, sharing some common properties. An image segmentation technique can detect specific shapes, which can be valuable for further analysis or processing.

2.3.2 Machine learning

The theory in this section is based on Goodfellow et al. (2016), unless other is stated.

A computer program is said to learn from ***experience E*** *with respect to some class of* ***task T*** *and* ***performance measure P****, if its performance at tasks in* ***T****, as measured by* ***P****, improves with experience* ***E****.* -Mitchell (1997)

Machine learning (ML) refers to the study of computer algorithms that improve automatically through experience and is a branch of *artificial intelligence* (AI).

Dataset

The first step in a ML project is to obtain knowledge about the problem domain and to gather or generate relevant data. A common practice in ML is setting aside some data for validation. A data-generating process will result in a dataset, $\mathcal{D}$. The training set, $\mathcal{D}_{\text{train}}$, is the data available for model training (learning). The validation set, $\mathcal{D}_{\text{val}}$, is the data used for model validation, which is used to tune parameters, select features, and make other decisions regarding the learning algorithm. A holdout set, $\mathcal{D}_{\text{test}}$, is used to evaluate the performance of the algorithm without making any decisions regarding what network or parameters to use. If the holdout set is used to adjust the learning algorithm or the parameters, data leakage/snooping occurs (Yaser, 2012), which is undesirable. A balanced dataset is crucial for the network to learn the features of all the classes effectively and to avoid bias of classes which are over- or underrepresented. To ensure a balanced dataset, the represented classes should have even distribution of samples.

Learning task and performance measures

The learning task for a ML network is often to make a model to predict, gain insight, optimise or control. A ML task refers to the specific prediction determined by the problem and the available data. The two most common are:

- **Regression**: used to predict a numerical value y given a data point x.
- **Classification**: used to specify which of k categories a data point x belongs to.

A performance measure is a designed, quantitative measure used to evaluate the performance of a ML algorithm on a learning task. Performance measures are often defined as an error where the goal is to minimise errors during training. For classification, the model accuracy is often used as a performance measure.

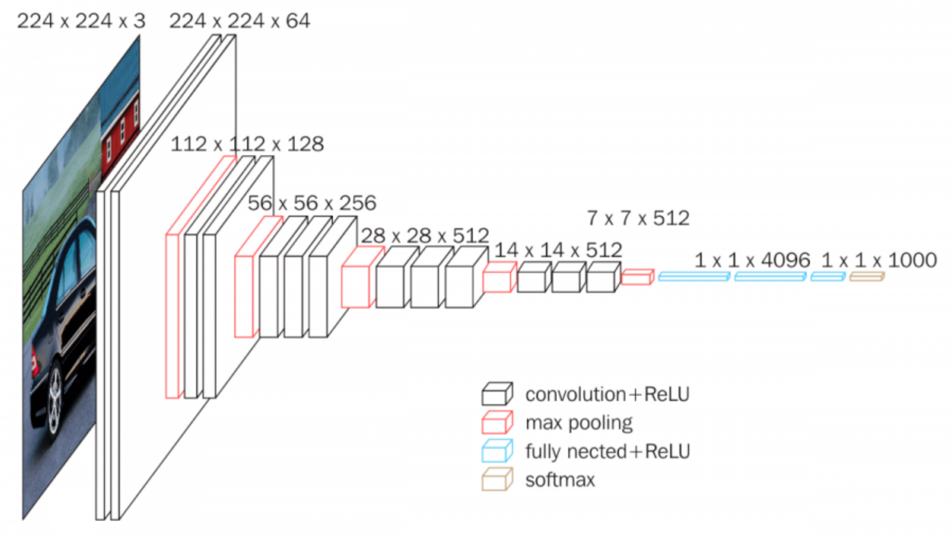
Loss function

In ML, loss functions measure the model performance by quantifying the difference between predicted and actual output (Rosasco et al., 2004). Cross-Entropy loss is commonly used for multi-class classification problems (Hui and Belkin, 2020). This loss function, also called logarithmic loss, compares the probability of the predicted classes with the actual label. A score/loss is calculated that penalises the probability based on how far it is from the actual expected value.

2.3.3 Neural networks

ML has undergone a significant transformation using Artificial Neural Networks (ANN). These computational models, inspired by the human brain (Agatonovic-Kustrin and Beresford, 2000), have outperformed traditional AI methods in standard ML tasks (Lecun et al., 1998). Among the most notable architectures of ANNs is the Convolutional Neural Networks (CNN). It is utilised for complex image-based pattern recognition tasks.

Visual Geometry Group (VGG) (Simonyan and Zisserman, 2014) is a CNN for image classification. The architecture uses small convolutional filters (3 x 3) and deep network depth. The authors released a series of VGG networks with different layer lengths, from 11 to 19 (VGG-11, -13, -16 and -19). Figure 2.3 illustrates the architecture of a VGG neural network. It consists of 13 convolutional layers, 5 max-pooling layers and three connected layers.



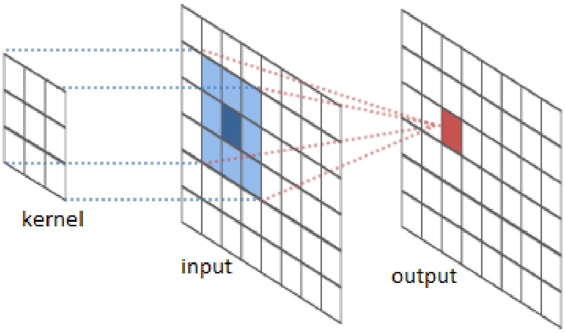
.

Figure 2.3: An illustration of a VGG16 architecture (Sugata and Yang, 2017).

The structure of a CNN can be separated into two parts. The first convolution/pooling layer(s) breaks up the image into features and analyses them. The fully connected takes the output from the convolution/pooling and predicts the best label to describe the images.

Convolutional layers

Convolutional layers consist of a filter, also called kernel or convolution matrix, passing over the image, scanning a few pixels at a time and creating a feature map which predicts the class to which each feature belongs. As the convolution matrix slides over the image, the dot product of the kernel and the image are computed at every spatial position. The convolutional layer is connected to an activation function. This operation is illustrated in Figure 2.4[2].

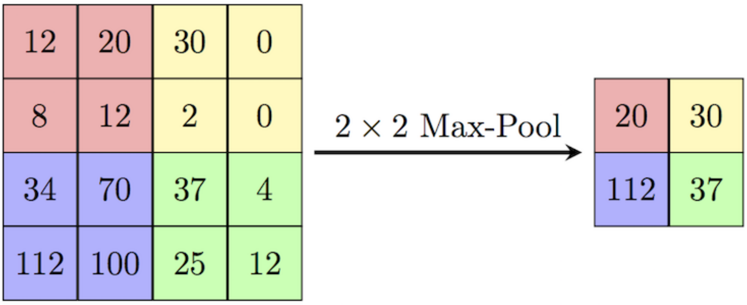


.

Figure 2.4: An illustration of the convolution operation.

Max pooling layers

Max pooling layers downsample the information in each feature extracted from the convolutional layer. It partitions the image into sub-rectangles and only returns the maximum value of the inside of that sub-region (Ozeki and Okatani, 2015). Figure 2.5 [3] illustrates a 2 x 2 max pooling. A pooling layer acts as a form of regularisation by gradually decreasing the spatial dimension and reducing the number of computations. Regularisation is the act of tuning or selecting model complexity to avoid overfitting. An overfitted model has trained with too much data, so it starts to learn from the noise and inaccurate data entries in the dataset.



12	20	30	0
8	12	2	0
34	70	37	4
112	100	25	12

20	30
112	37

.

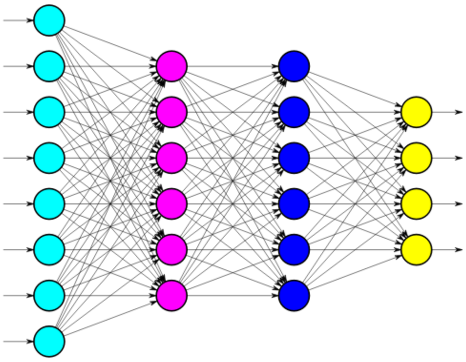
Figure 2.5: An illustration of a 2 x 2 max pooling.

[2] Accessed: 10.05.23. Available at: https://towardsdatascience.com/beginners-guide-to-understanding-convolutional-neural-networks-ae9ed58bb17d.

[3] Accessed: 10.05.23. Available at: https://computersciencewiki.org/index.php/Max-pooling_/_Pooling.

Fully connected layers

In a fully connected layer, also called a dense layer, all the neurons in the previous layer are connected to every neuron in the next layer. The fully connected layers often use activation functions to determine the output of the neurons. Figure 2.6 illustrates a dense layer.



.

Figure 2.6: An illustration of a fully connected neural network.

Activation functions

Activation functions are applied to the input to have an output with a given characteristic to improve the output (Nair and Hinton, 2010). Rectified Linear Unit (ReLu) is a threshold at zero, such that it ensures positive values, yielding $f(x) = max(x, 0)$. The softmax function is often used as the final activation function in a CNN (Bishop, 2006). It normalises the network's output, creating a probability distribution over predicted output classes, based on Luce's choice axiom (Luce, 1977).

Batch size and epochs

Batch size and the number of epochs are hyperparameters in training a NN. The batch size refers to the number of data samples that the network processes before the internal parameters of the network are updated. It is a crucial determinant of how effectively the network can learn from the data, striking a balance between computational efficiency and the accuracy of the learning process. The number of epochs is the total number of iterations where the network goes through the entire dataset. A high number of epochs could lead to better learning up to a point, but beyond this, it may result in wasted computation or overfitting to the training data.

Generalisation is a central challenge in ML. It refers to the ability of a learning algorithm to perform well on new, previously unseen data drawn from the same distribution used to train the network. During hyperparameter tuning, there is a sweet spot for choosing the model that does not over-or underfit.

Evaluation metrics

To evaluate the training of a NN several metrics can be used to evaluate the performance of the model.

Classification accuracy is often referred to as the accuracy of the model. It is calculated with the number of correct predictions divided by the total number of input samples. It can provide misleading results if the dataset is unbalanced. If most samples in a dataset belong to the same class (or classes), they could overshadow the samples from less frequent classes. Accuracy is the most common evaluation metric, as it is the most intuitive.

Learning curves plots show changes in learning performance over time (epochs) in terms of experience. A model's optimal loss function value for a training set against the same loss function value evaluated on a validation dataset using the same parameters which obtained the optimal function (Mohr and van Rijn, 2022) is a learning curve. The metric evaluates the training process and whether the network benefits from more training. If a network is over- or underfitting, this could be observed in a learning curve by comparing the convergence of the training- and validation loss. Studying learning curves and tuning hyperparameters based on the output is valuable for choosing the best model.

Confusion matrix is a matrix that provides a visualisation of a ML performance. It is used in supervised learning algorithms where the outputs are known and are compared with the predicted outputs. This matrix includes information about actual and predicted classifications made by the model. Each row of the matrix corresponds to the actual classes of the objects, while each column corresponds to the predicted class. In a confusion matrix for a multi-class classification task with classes A_1, A_2, ..., A_n, the entry in the *i-th* row and *j-th* column, represented as N_ij, indicates the number of samples that belong to the actual class A_i, but are predicted as class A_j (Deng et al., 2016).

3

Hatching-sensor

This chapter covers the various physical components that constitute the hatching sensor. A review of two light sources, the setting and the parameters used for the camera are conducted through testing. The chapter delves into the aspects of image acquisition. The goal is to develop and evaluate a prototype hatching-sensor acquiring images to make a dataset for training the machine vision algorithm. The test setup of the sensor can be seen in Figure 3.1.

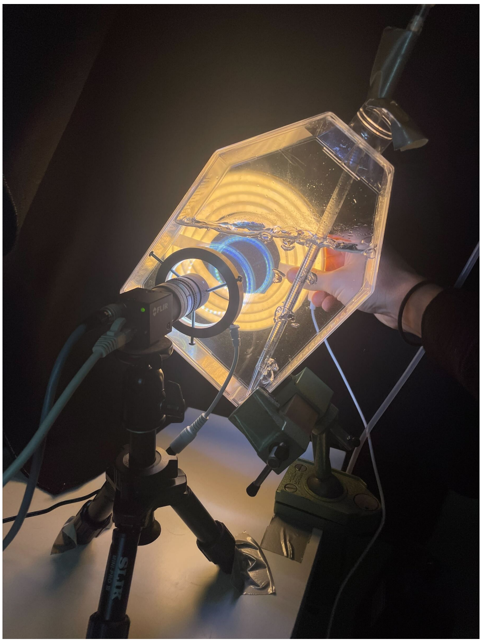


Figure 3.1: The hatching-sensor setup utilising a camera, aeration and illumination.

3.1 Components

3.1.1 Camera

The FLIR BFS-PGE-50S5M-C PoE GigE Blackfly® S Monochrome Camera [1] provides grey-scale images. The camera delivers images with 2448 x 2048 resolution. The frame rate is 24 frames per second (fps). The Gigabit Ethernet (GigE) interface enables high-speed data transfer from the camera to a computer. To achieve the correct focus, a Pentax 25mm lens was used with two extension rings (50mm each). The camera was employed to capture the images.

SPINVIEW was used to manage image acquisition. This software can adjust camera settings, such as exposure time, aperture, and gain, to test and improve the image quality under different experimental conditions. By tuning these variables, consistently clear and well-lit images of the copepods were obtained. Figure 5.8 provides a screenshot of the settings which can be adjusted in the software.

3.1.2 Illumination

A black fabric created a ”light tent” for accurate and consistent lightning. The purpose was to provide a controlled environment with consistent lighting conditions. Various variations

[1] Accessed: 28.04.23. Available at: `https://www.flir.com/products/blackfly-s-gige/?model=BFS-PGE-50S5M-C&vertical=machine+vision&segment=iis`.

of indirect lighting were tested. Two different light sources and their placement were tested.

- LED Ring Light[2]
- IKEA Myrvarv LED lightstrip[3]

The goal was to obtain a sharp image with clear contours of both eggs and nauplii.

3.1.3 Aeration

The hatching tanks need to be stirred to prevent the eggs from clustering and sedimentation to the bottom of the tank. Aeration makes the water move, which is also vital for successful hatching. A pump with a serological pipette was used to create aeration. Eheim Air Pump 100[4] was used in the project. The maximum amount of air it can produce is 100 litres/hour. An experimental test was conducted to examine the impact of aeration on the homogeneity of the mixture. Details of this Test can be found in section 5.5.

3.1.4 Biology lab equipment

Biological equipment is necessary to carry out a hatching process and subsequent testing. Table 3.1 contains the required components to conduct a hatching process of copepods.

Item	Description
Disinfected seawater	The environment in which the hatching process will take place.
A mixture of copepod eggs	Copepods is stored in a high seawater concentration.
NUNC™ EasyFlask™	Serves as the actual hatching tank. A flat surface is advantageous for achieving the correct focus for image acquisition.
Pipette	This is necessary to accurately measure the appropriate volume of the egg mixture in millilitres.
Pump and serological pipette	Prevent the eggs from sedimentation.
Stand for the NUNC flask	Support structure for holding the NUNC flask. The flask is angled to prevent sedimentation of the eggs.

Table 3.1: Equipment necessary to perform a hatching process of copepods.

3.2 Testing

Several tests have been performed in order to gain insight. Trial and error have been a part of the project. The development process has evolved progressively, involving multiple iter-

[2] Accessed: 28.04.23. Available at: `https://www.adafruit.com/product/4433`.

[3] Accessed: 28.04.23. Available at: `https://www.ikea.com/us/en/p/myrvarv-led-light-strip-flexible-dimmable-90487191/`.

[4] Accessed: 28.04.23. Available at: `https://greenaqua.hu/en/eheim-air-100.html`.

ations of hatching experiments. The hatching process of copepods can be seen in Appendix B. A sample of 3.5mL copepod eggs was given from CFEED for testing. The density was approximately 400 000 eggs/mL mixed with approximately 11.5mL disinfected seawater. The concentration was

$$\frac{400000 eggs/mL * 3.5mL}{15mL} = 93333 eggs/mL. \tag{3.1}$$

in the sample. In Test I, II, III and IV (section 5.1, section 5.2, section 5.3 and section 5.4respectively), this sample was used mixed with different amounts of disinfected seawater, which gave different densities. In the hatching tanks, the density varies from batch to batch in a range of 10-1000 eggs/mL, according to CFEED. For testing purposes, an initial density range for the eggs was established to be between 100 and 1000 eggs/mL. The hatching process performed in Trondheim was done in a NUNC flask [5], and the setup can be seen in Figure 3.1. The NUNC flask was used because of its flat surface and availability of the flask. The hatching process in Vanvikan was performed in a soda bottle according to CFEEDs hatching manual (Appendix B). A photograph in Figure 3.2 shows the hatching processes and fetching a sample. The photo was acquired during the testing in section 5.5.

[5] Accessed: 28.04.23. Available at: `https://www.thermofisher.com/order/catalog/product/156499`.

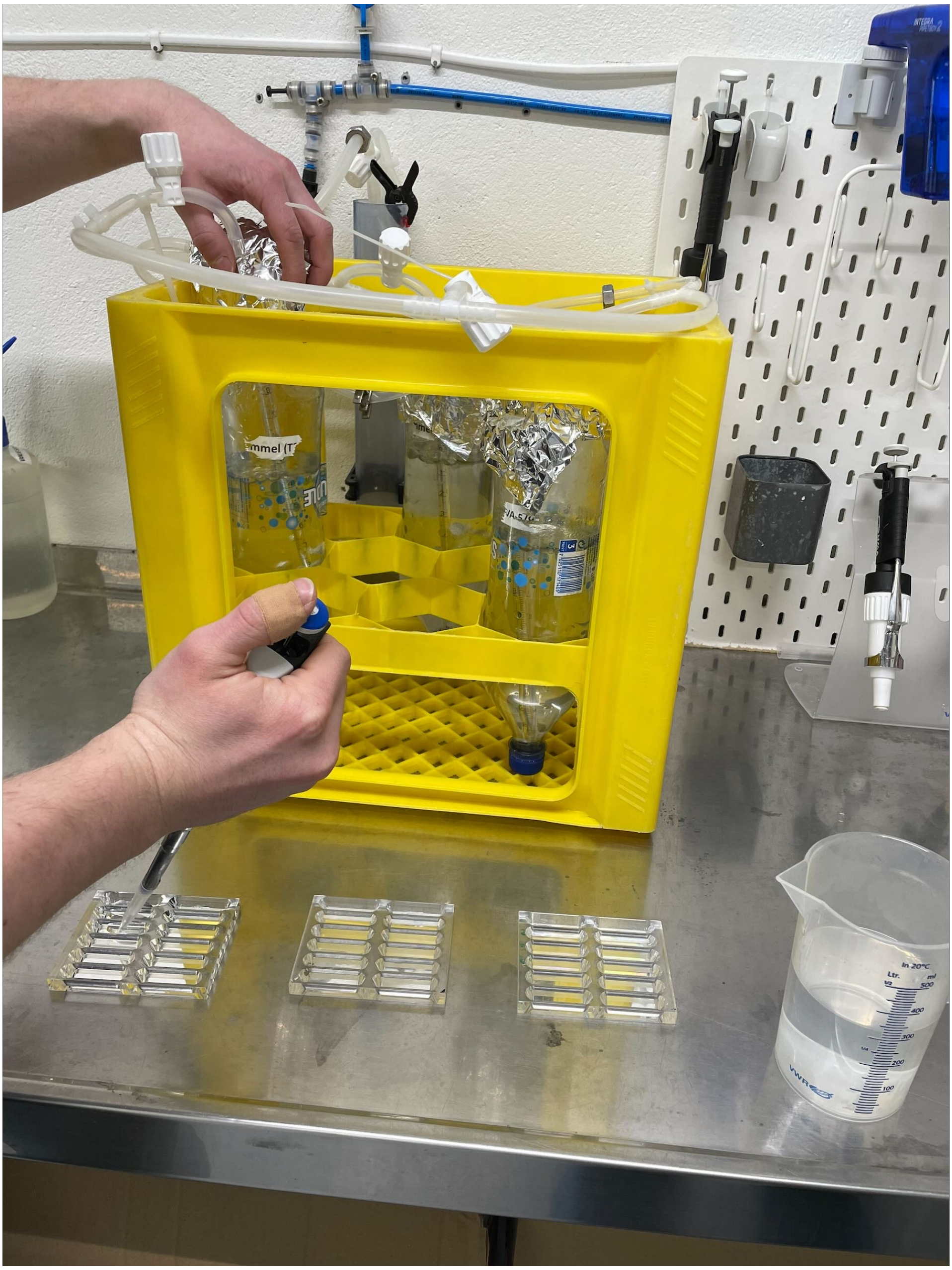


Figure 3.2: A photo of the hatching process performed at CFEEDs production facilities.

3.3 Images

There are some general considerations regarding the image format for machine vision algorithms. Particle detection relies on the accuracy of the pixel. Therefore it is best to use a lossless format that preserves the original pixel values of the image. The camera captures 23 FPS when using .bpm-file. The .bpm-file is of size 2448x2048 pixels and is the result of the image acquisition. The process and the images acquired can be viewed in chapter 5.

4

Machine vision algorithm

This chapter covers image labelling and enhancement, methods and techniques employed to improve image quality for further analysis, and the machine vision algorithm, including a deep neural network.

4.1 Image processing

The images generated by the hatching sensor are in the .bmp format and are ready for further processing, allowing the ML algorithm to be trained. The pipeline can be viewed in Figure 4.1.

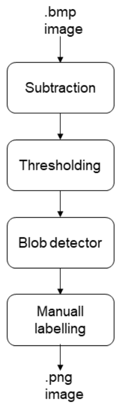


Figure 4.1: Flow diagram of the image processing.

4.1.1 Pre-processing

To detect and remove stationary noise, the previous image is subtracted from the current image using OpenCV's subtract function, cv.subtract. Figure 4.2 illustrates the process of subtracting the second image from the first image, and the third image is the result. The image is brightened to get better visualisation. Next, the cv.threshold function is applied to set a binary threshold for the image. All pixels with values above 50, **thresLevel**, are set to 255. Pixels with values less than or equal to 50 are set to the minimum intensity (0). The result is a binary image suitable for detecting particles in areas of interest. Figure 4.3 illustrates the result of thresholding.

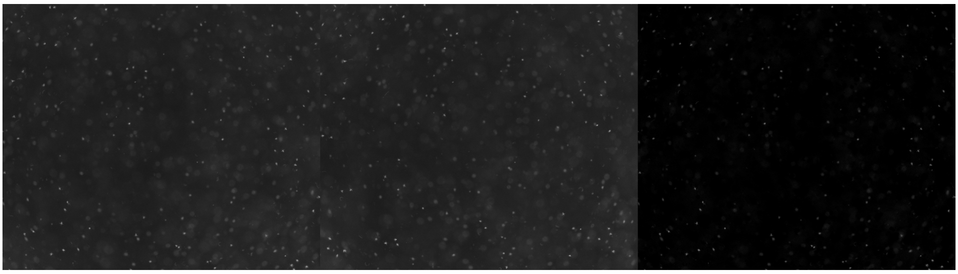


Figure 4.2: Image subtraction of the second image on the first image, resulting in the third image.

Figure 4.3: thresholding on the image from the subtraction.

4.1.2 Blob Detection

The binary image is used to detect particles that may be of interest. OpenCV's blob detector[1] is used. The function identifies a group of connected pixels called a "blob." The number of pixels determines the area size of a blob. By specifying **params.minArea** and **params.maxArea**, blobs that are too small or too large relative to the size of eggs or nauplii, will be filtered out section 2.1.

4.1.3 Annotating

The result of the image processing presents individual particles so that a user can label the images. Each particle is presented, the user determines what it is, and the image is saved as a .png-file and a .csv-file containing the labelling information. Figure 4.4 presents the images to the user. The labelled images form a dataset used to train the VGG. The dataset was split into 70% for training and 30% for validation. The annotator converts the 1-channel greyscale .bmp images to 3-channel RGB .png images. The conversion is done prior to feeding them to the network. The VGG network expects 3-channel RGB images as input. When a greyscale image in the .bmp format (1-channel) is converted to an RGB .png image (3-channel), it retains its original information while expanding from a single channel to three channels. Each of the three colour channels in RGB will contain the same

[1] Accessed: 08.05.23. Available at: `https://learnopencv.com/blob-detection-using-opencv-python-c/`.

intensity values as the original greyscale image. The result is an equivalent representation of the original greyscale image in the RGB colour space.

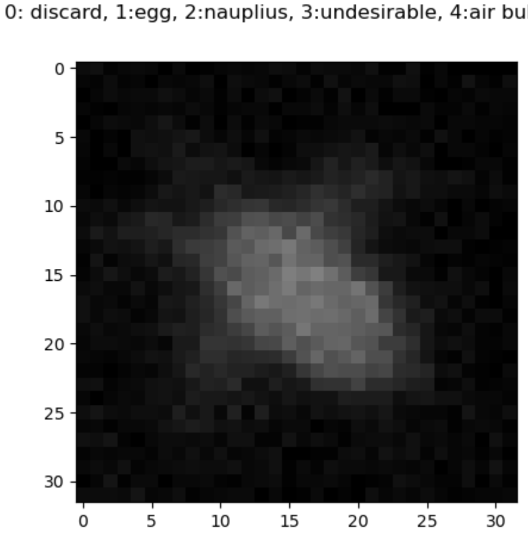


Figure 4.4: The user is presented with a particle withdrawn by the blob detector and is manually labelled with the keyboard input 0, 1, 2, 3 or 4.

4.2 Training process

The implementation of a VGG-network [2] makes it possible to change between the structures VGG11, -13, -16 and -19. The cross-entropy-loss and the accuracy were the performance measures used in the training process. Learning curves and confusion matrices were used to evaluate the performance. The learning task was to predict and gain insight into a hatching process. Experimenting with the number of epochs and the batch size and testing different VGG architectures was the main focus. This thesis does not focus on hyperparameter tuning and optimising with the architecture of the network. The following hyperparameters were used as a recommendation from the origin of the network:

- Validation split = 0.3
- Random seed = 42
- Learning rate = 0.001
- Momentum = 0.9

[2] Accessed: 28.04.23. Available at: `https://www.analyticsvidhya.com/blog/2021/06/build-vgg-net-from-scratch-with-python/`.

The data generating process was images acquired by the hatching sensor, resulting in a dataset, $\mathcal{D}$ The dataset was split into $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$ from the same distribution. Figure 4.5 illustrates how the network was trained, evaluated and tuned on the training and validation set. Test sets containing independent and unseen data were obtained to test the trained model and monitor the egg-nauplii ratio throughout hatching.

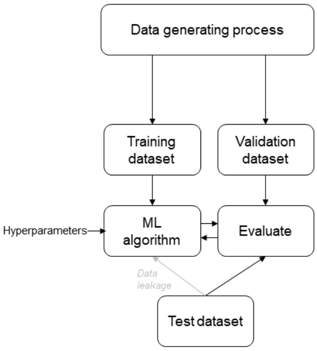


Figure 4.5: The pipeline of the training process of the network.

The training and test processes with the images acquired by the hatching sensor can be viewed in section 5.7, section 5.8, section 5.9 and section 5.10.

5

Experimental testing

This chapter will cover the experimental testing to develop the hatching sensor and the machine vision algorithm. Camera settings and light sources have been the main focus in the implementation of the sensor.

5.1 Test I

An initial test was conducted to familiarise with the hatching process and the equipment involved. The hatching occurred in a small NUNC flask (175mL), with the ring light attached to the camera lens. The setup can be seen in Figure 5.1. The small NUNC flask was used to minimise the loss of copepod eggs. CFEED provided a 15mL sample for testing purposes. 1mL of eggs was used and provided a concentration of *86 372eggs / 176mL = 490* eggs/mL.

Various camera settings were tested to familiarise with how they affected the quality of the images. The exposure time, gain and aperture were the main focus. Acquiring images while there is continuous aeration is desirable, as mentioned in section 2.1. However, it was observed that bubbles occupied the entire image frame during continuous aeration, so this approach was quickly ruled out using the small NUNC flask. The size and shape of the flask affect the water circulation. In subsequent image acquisition, images were taken approximately 2-5 seconds after stopping the aeration to see the particles more clearly. Figure 5.2 presents images with different shutter speeds to investigate the boundary at which particles appear sharp or blurred. The four images have a gain of 0dB.

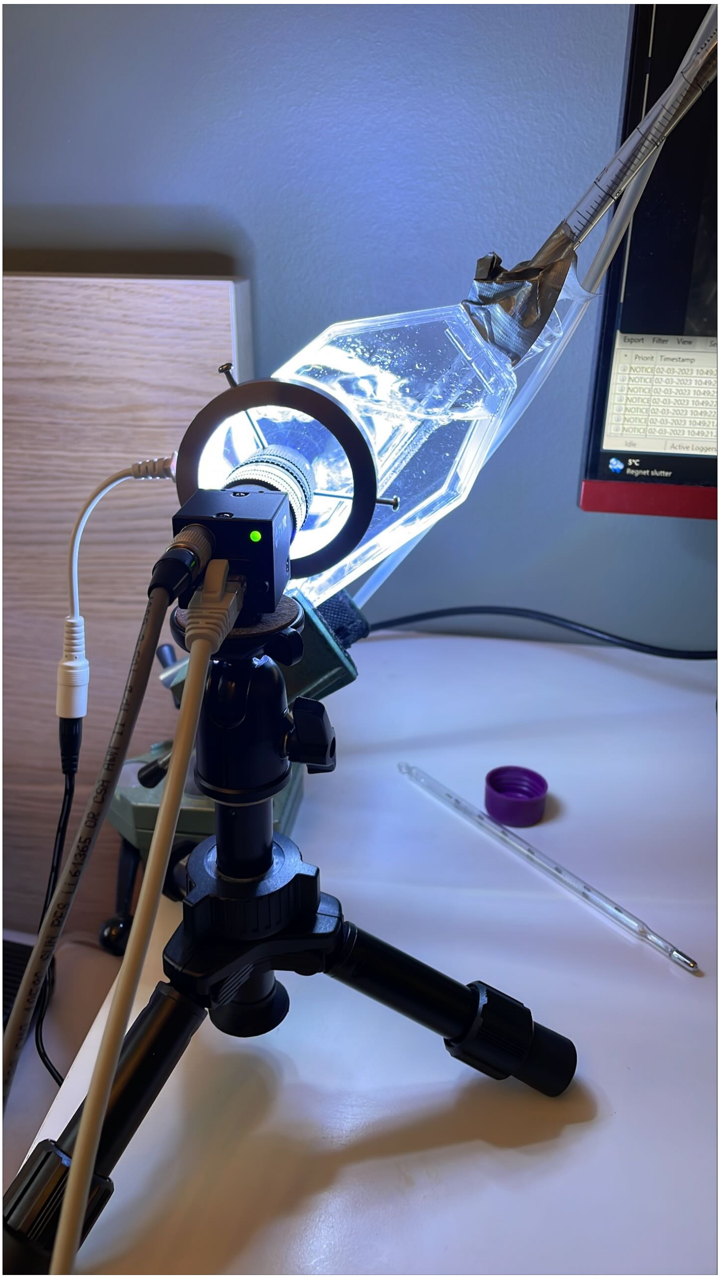


Figure 5.1: Initial hatching-sensor setup.

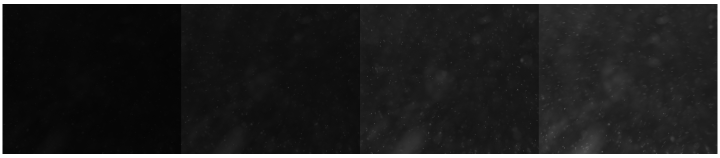


Figure 5.2: Four images acquired with exposure time of 1.506ms, 2.000ms, 3.993ms and 7.005ms respectively.

It can be observed that the sharpness of the particles gets worse with a higher exposure time, and more of the particles become visible. A longer exposure time, higher gain, or lower aperture could be applied to get a more visual representation of the particles.

5.1.1 Main findings

- The small NUNC flask was too small to acquire images with aeration. The geometry of the hatching tanks influences the water movement and thereby influences the hatching process.
- Make a setup where the surrounding light conditions are the same each time.

5.2 Test II

The setup in Figure 5.1 was tested using a large NUNC flask (855mL). In addition, a light tent was used to reduce light pollution. The density of the eggs was approximately 200 eggs/mL. Initially, the ring light was placed on the camera's extension rings, at the outer edge of the lens, and on the opposite side of the NUNC flask. Placing the ring light at the outer edge of the camera lens yielded the best results and was used for subsequent tests. The other two options showed glare on the images, as shown in Figure 5.3.

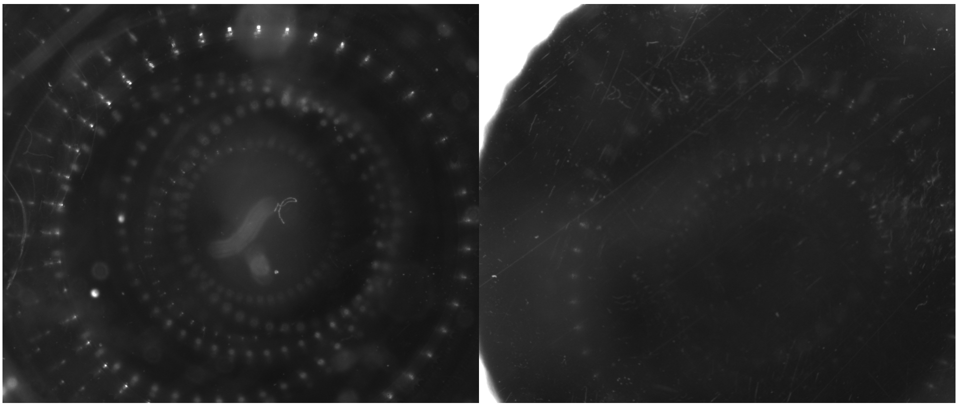


Figure 5.3: Two images acquired with the ring light placed on the opposite side of the NUNC-flask and the camera's extension rings, respectively.

Images were taken at 0 hours and after approximately 48 hours at room temperature. The images were captured with and without aeration (10 and 20 seconds pause) in the respective intervals of [0.2, 4]ms and [2, 10]ms. SPINVIEWS automatic gain tuning was set to 17.9 dB. The aperture was the same for the two first images. A smaller aperture for the two last images was chosen to reduce the light intensity, thereby reducing noise. The images were acquired at the start of the hatching process (0 hours), so the particles are either eggs, dust or air bubbles. The result can be seen in Figure 5.4. It is possible to see particles in all four images, but the clarity varies. The second image shows that noise is amplified when using the same aperture but with a longer exposure time. In images three and four, the particles start to become blurry. More light will be introduced for the next test to improve visibility and get a more precise contour.

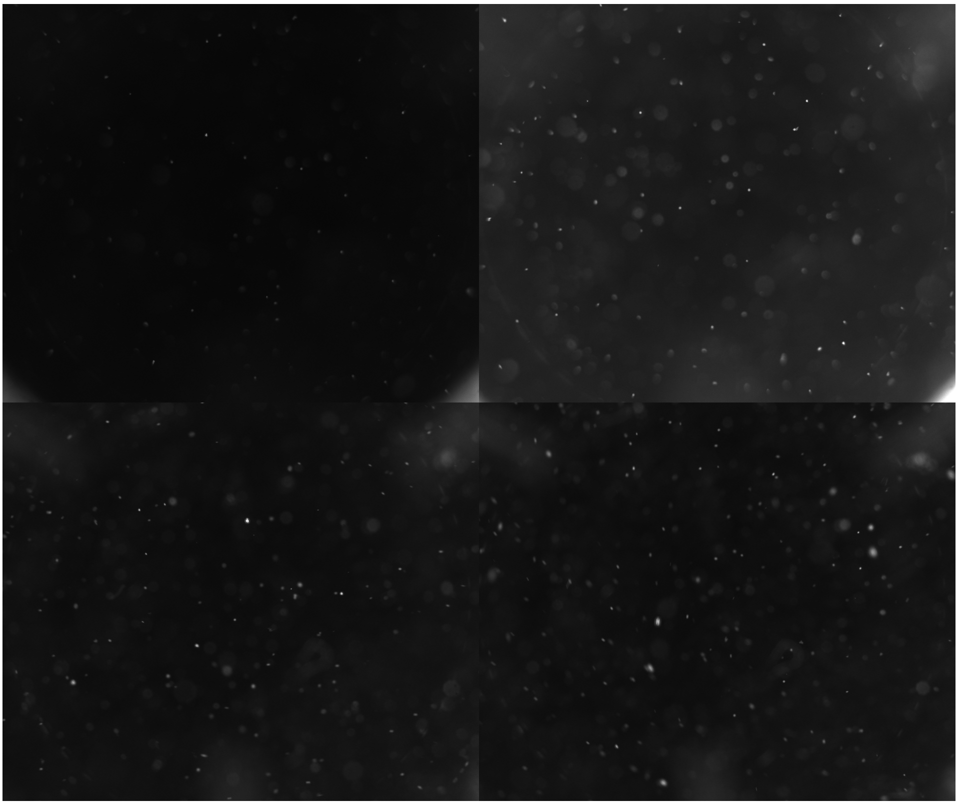


Figure 5.4: Four images acquired with different aperture and exposure time of 0.2ms, 1ms, 2ms and 3.0ms respectively and continuously aeration.

Images of nauplii were acquired with different shutter speeds. It is possible to see the contours of what appears to be nauplii. The same observations were made here. In Figure 5.5, it can be observed that the particles become sharp, even with a considerably longer shutter speed of 8ms. The challenge with stopping the aeration is whether the mixture is homogeneous or not. The mixture must be homogeneous to determine a representative hatching rate and percentage.

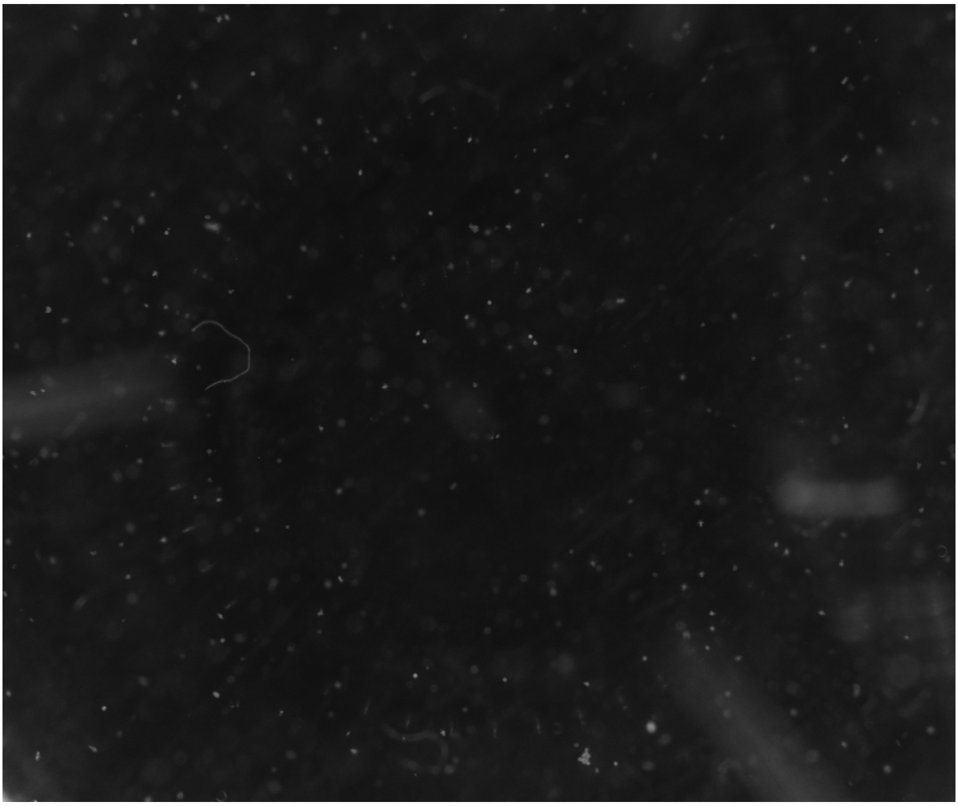


Figure 5.5: One image acquired with an exposure time of 8.0ms and 20 seconds without aeration.

5.2.1 Main findings

- Use the automatic gain set by SPINVIEW to 17.9dB.
- Increase the illumination to visualise the contour of the particles by using an additional light source.
- The 850mL NUNC flask can be used with continuous aeration without disturbing the acquired images.
- The images become clearer, particularly of nauplii, when the aeration has been stopped. Stopping the aeration can affect whether the mixture is homogeneous or not.

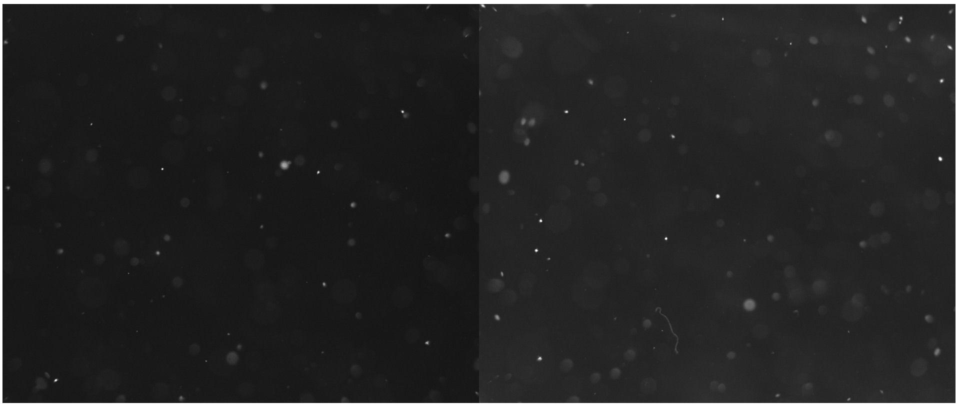


Figure 5.6: Two images acquired to demonstrate the two light sources.

5.3 Test III

In this test, the focus is on the image quality of individual particles. The objective was to test the placement of both light sources and finalise a test setup. Comprehensive testing of the placement of the two light sources was performed. Various configurations of the ring light and the light strip were tested. Figure 5.6 compare the led strip in the back with- and without the ring light in the front. The best result was acquired with both light sources. It resulted in the final setup in Figure 3.1. To determine an exposure time for the setup, images in the interval of [0.1, 1]ms were conducted. It was challenging to decide what would give the best result. There is a trade-off for what will achieve the best result. In Figure 5.7, some selected images with different shutter speeds can be seen. It can be observed in the images that the amount of noise increases when the aperture and exposure time is increased. An exposure time of 0.2ms was chosen for further use. The choice was based on the amount of noise and the clarity of the individuals' contours. The final settings of the camera can be seen in Figure 5.8, a screenshot from SPINVIEW.

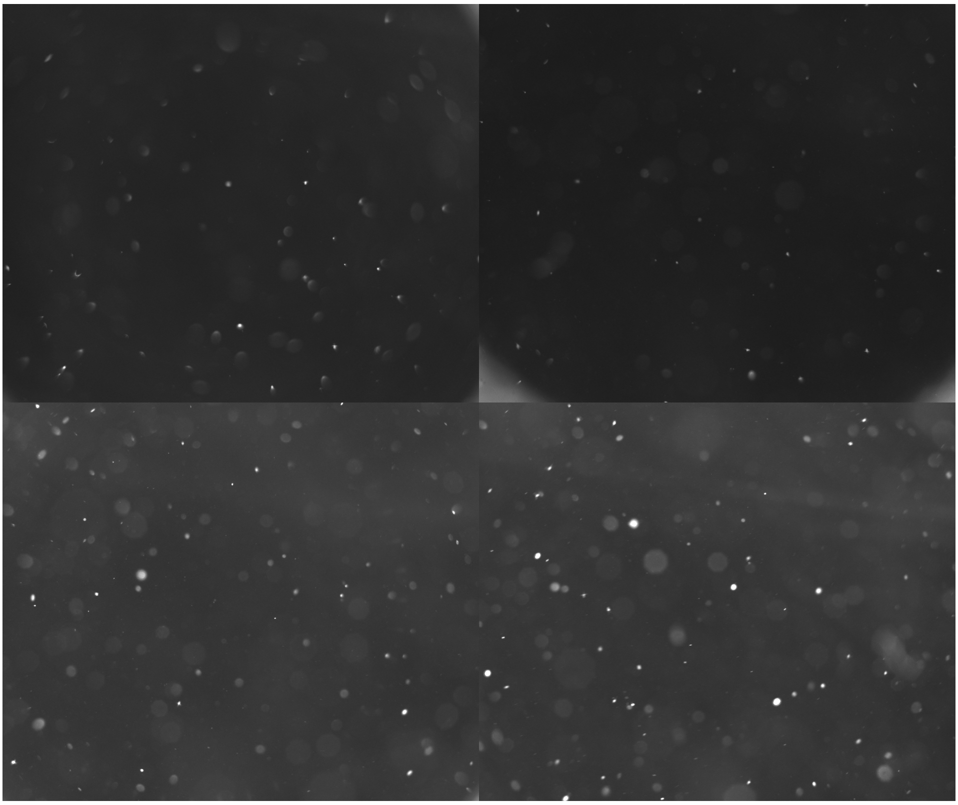
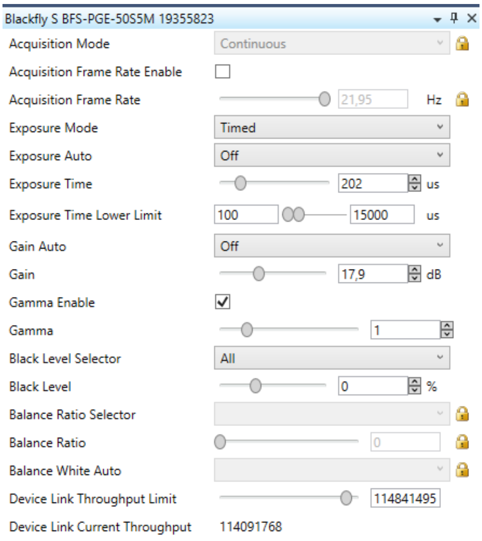


Figure 5.7: Four images acquired with exposure time of 0.1ms, 0.2ms, 0.6ms and 1ms respectively.



Blackfly S BFS-PGE-50S5M 19355823

Acquisition Mode	Continuous	
Acquisition Frame Rate Enable	□	
Acquisition Frame Rate	21,95	Hz
Exposure Mode	Timed	
Exposure Auto	Off	
Exposure Time	202	us
Exposure Time Lower Limit	100 – 15000	us
Gain Auto	Off	
Gain	17,9	dB
Gamma Enable	☑	
Gamma	1	
Black Level Selector	All	
Black Level	0	%
Balance Ratio Selector		
Balance Ratio	0	
Balance White Auto		
Device Link Throughput Limit	114841495	
Device Link Current Throughput	114091768	

Figure 5.8: The camera settings from SPINVIEW.

5.3.1 Main findings

- The exposure time will be set to 0.2ms.
- Led strip behind the NUNC flask and ring light mounted on the lens's edge will be the light sources' positions.
- A set up for the hatching sensor to achieve a similar image quality for all testing.

5.4 Test IV

This test aimed to collect test data to create a dataset that could be used for training the machine vision algorithm with the setup in Figure 3.1. The camera's aperture was set to approximately 6. The egg density was approximately 600 eggs/mL. As mentioned in section 2.1, a hatching process takes about 48 hours at room temperature. Images were taken with shutter speeds of 0.2 ms, 0.4 ms, and 0.6 ms. Only 0.2 ms was used. Five hundred images were taken with each of the three shutter speeds to gather enough data. This was done 12 times over 51.5 hours. The acquired images formed the first Dataset used for training the neural network. Figure 5.9 presents a set of images after using the annotator. The images in the figure were acquired after 0 hours. The majority of the images consist of eggs. Some images exhibit multiple eggs within a single frame.

A few of the images appear as noise. Instances of air bubbles are also observed. Figure 5.10 presents a set of images from approximately 21 hours into the hatching process. Eggs, nauplii, noise and air bubbles can be observed. Figure 5.11 presents a set of images acquired after approximately 48 hours. The hatching process should be done at this stage. Fewer eggs and several nauplii can be observed, which indicates hatching. The images in the three figures have been processed with the annotator, and no discarding was done. Some images of noise, dust and air bubbles are present. Approximately 13 000 - 15 000 images of particles were acquired using the annotator on 500 images.

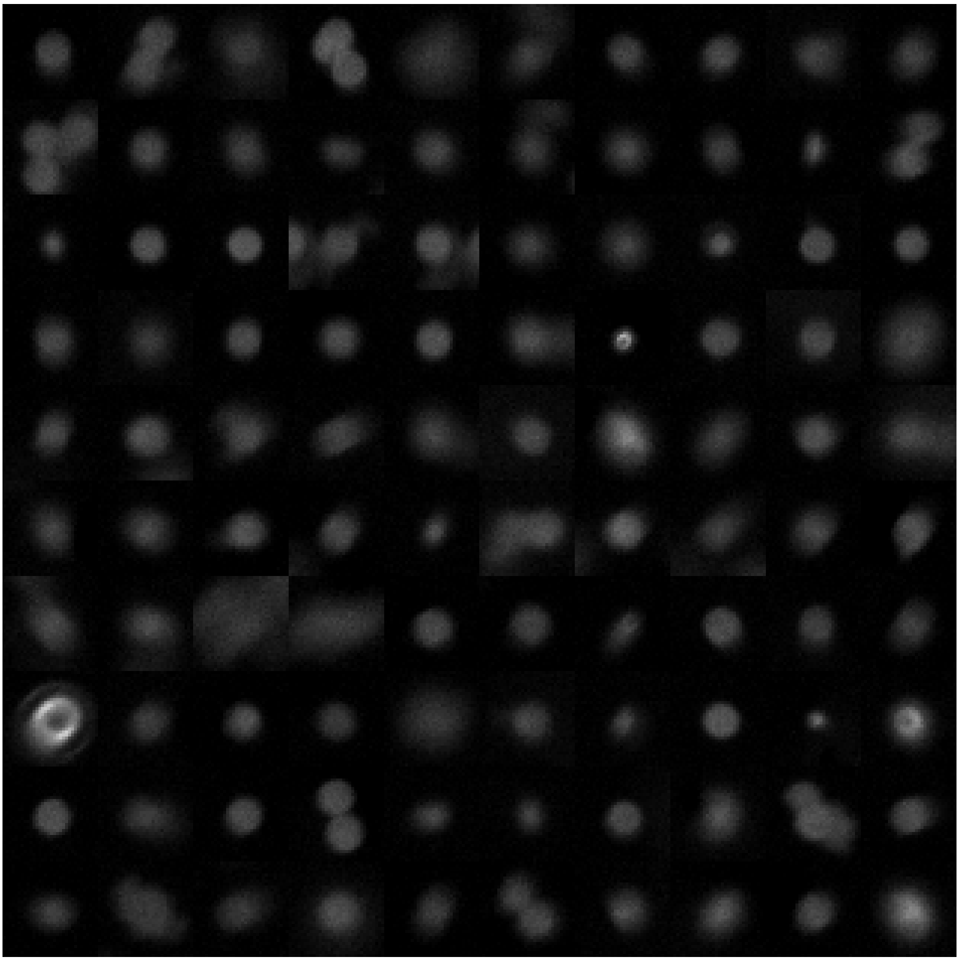


Figure 5.9: A sample of images acquired approximately 0 hours into the hatching process. Eggs, noise and air bubbles can be observed.

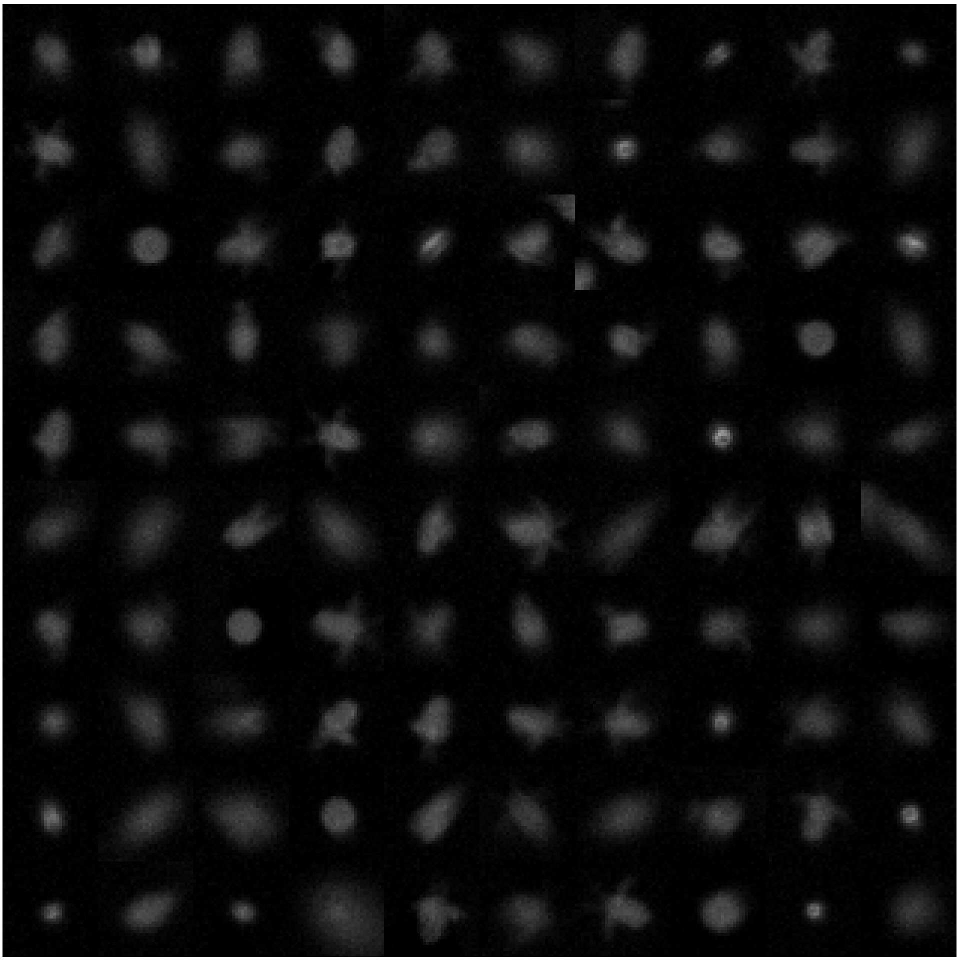


Figure 5.10: A sample of images acquired approximately 21 hours into the hatching process. Eggs, nauplii, noise and air bubbles can be observed.

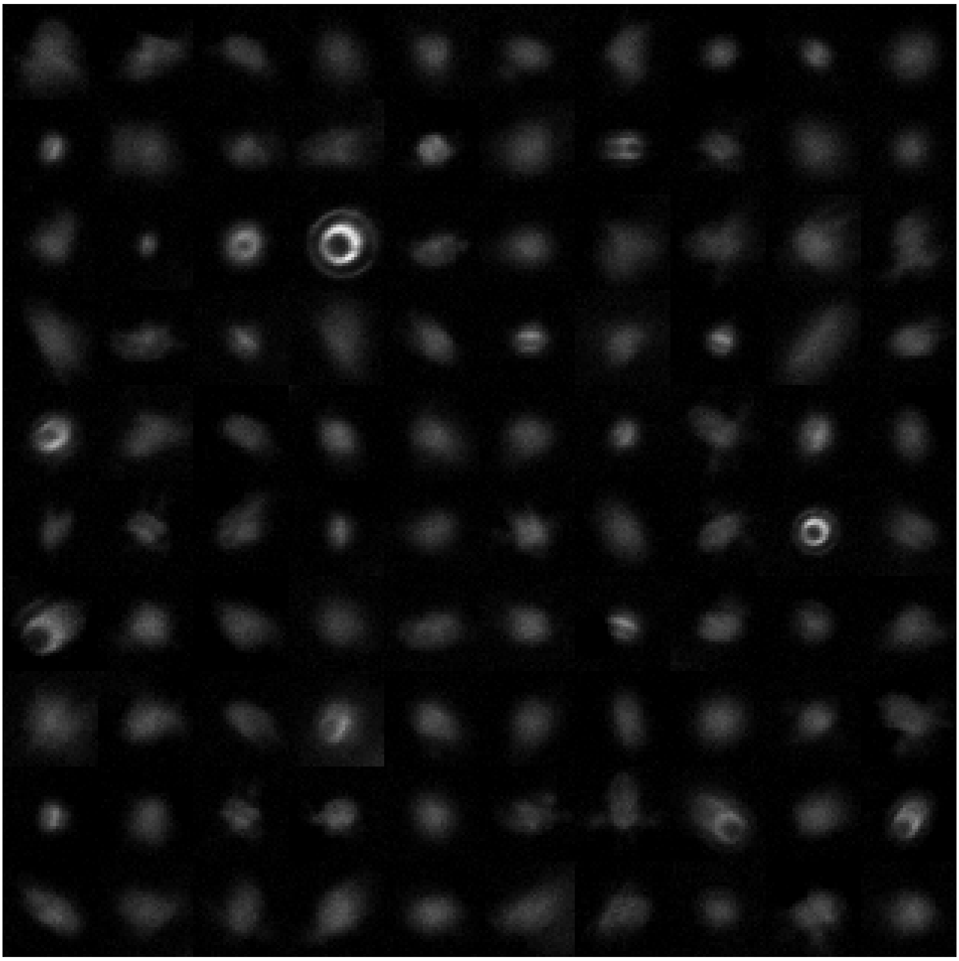


Figure 5.11: A sample of images acquired approximately 48 hours into the hatching process. Nauplii, noise and air bubbles can be observed.

Particles observed along the bottom of the NUNC flask indicate sedimentation, which can adversely impact the hatching process. A photo of these sediments can be seen in Figure 5.12.

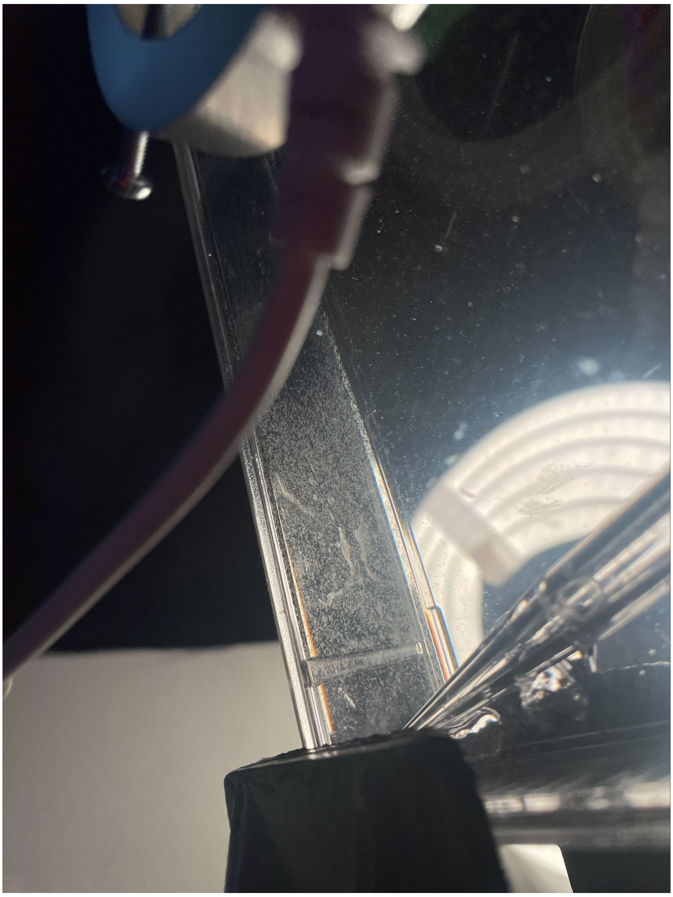


Figure 5.12: A photo of sedimentation during the hatching process.

5.4.1 Main findings

- Collecting images to create datasets.
- Perform a full hatching process.

5.5 Test V - Homogeneity test

The goal was to test how long the aeration could be turned off without resulting in a non-homogeneous mixture. Turning off the aeration before taking the image could improve image quality because of the smaller movement of the individuals. Two visits to CFEEDs production facilities were necessary to obtain the knowledge to perform the test and the actual execution of the test. A mixture was prepared in which the density of eggs and nauplii was predetermined. Then, counting samples were taken: six parallels while the aeration was on and ten parallels while trying 3, 6, 10, 15 and 20 seconds without aeration. Considering the changes in variance and measured density, an opinion can be made regarding the duration for which aeration can be switched off without compromising the homogeneity of the mixture. Parts of the data collected can be seen in Appendix A.

The highest and lowest values have been removed from the Dataset, as it is common practice to reduce noise from randomly high or low samples. The standard deviation in the Dataset is relatively consistent, generally around +/- 5, which can be observed in Figure 5.13. The percentage deviation from the count with aeration mainly remains within +/- 5% up to 15 seconds after the bubbling has stopped. Figure 5.13 presents the deviation from counting. Some percentage deviation is expected when taking samples, but deviations exceeding 5% may significantly impact the hatching percentage, especially if values diverge in opposite directions. Based on the collected data, there is potential for collecting images while stopping the aeration. More than relying on a single dataset for analysis may be required, as the resulting values could be subject to random variations. Three separate tests would have been preferable to enhance the experiment's validity.

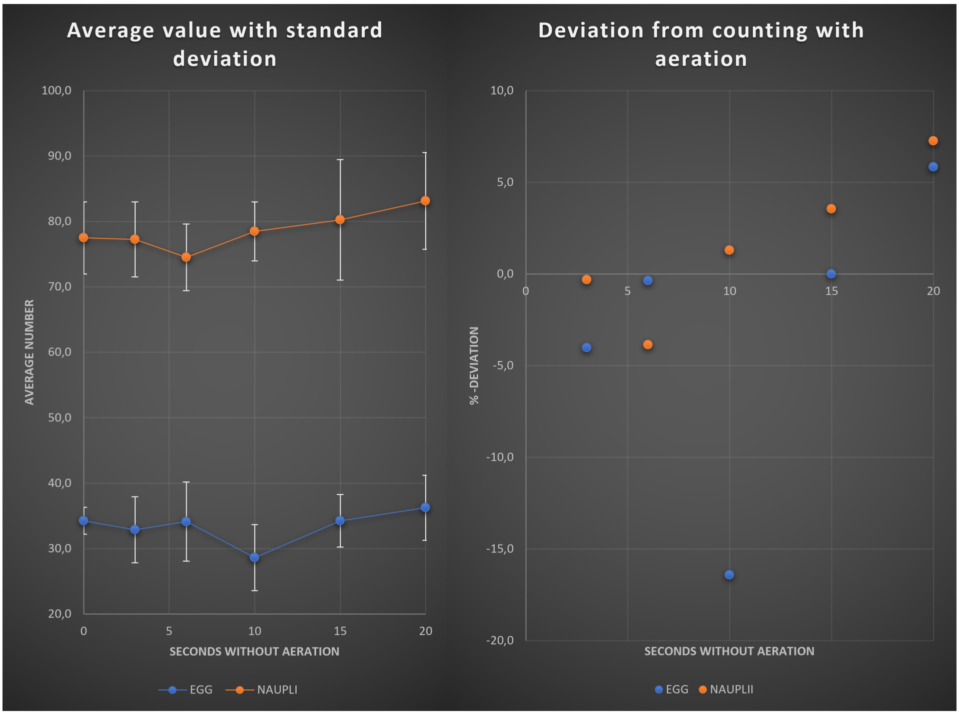


Figure 5.13: Deviation from counting with aeration and the average value of eggs and nauplii with standard deviation.

5.5.1 Main findings

- Firsthand experience is why it is desirable to have a hatching sensor replacing manual counting samples.
- There is potential to maintain a homogeneous mixture even when the aeration is stopped for several seconds.
- A more extensive testing of how the aeration must be provided to increase the validity.

5.6 Test VI - Shell or nauplii

It has proven most challenging to distinguish between empty eggshells and nauplii. This sometimes makes labelling the images difficult, as there is uncertainty about what it is. To investigate the visual differences between eggshells and nauplii, attempts were made to take pictures of only shells and only nauplii. A hatching process was set, and after about 48 hours, the experiment was conducted. First, the aeration was turned off for about 1 hour, allowing the eggs to sediment. In Figure 5.14, the eggs can be observed in the cone in the two pictures. The mixture was expected to consist of only shells and nauplii. The aeration was put back in, and the rotifer floss was put in. After approximately 3 hours, the rotifer floss was cleaned. The cleaned mixture was expected to only consist of shells. In Figure 5.14, in the picture to the left, the brown colour on the rotifer floss are particles which are attached to the white surface. The rotifer floss was cleaned in the picture to the right, and the mixture of just expected shells was made.

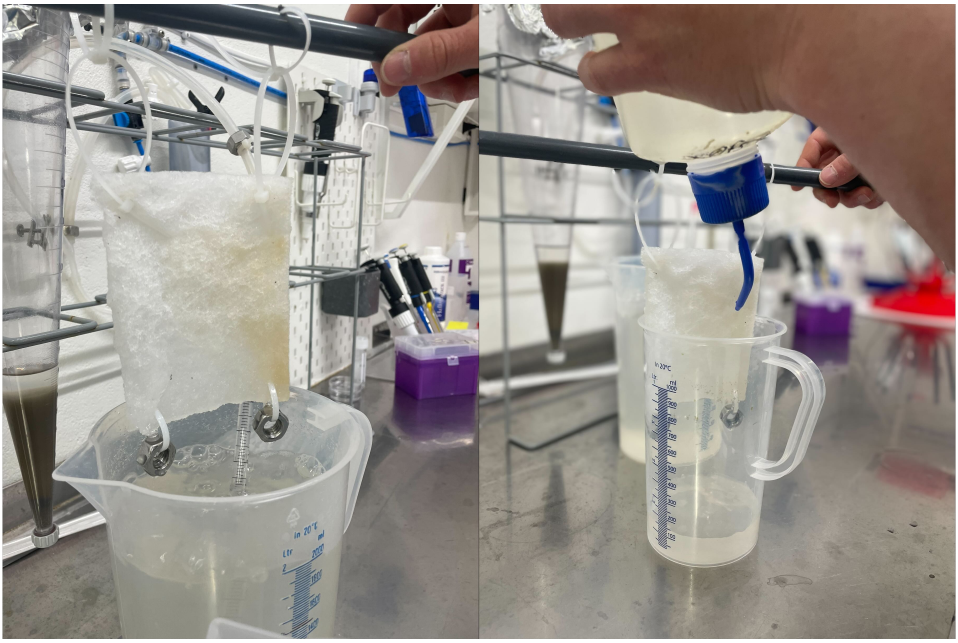


Figure 5.14: Pictures from using rotifer floss to get mixtures of only shells and nauplii.

The mixture of just nauplii and shells was poured into a NUNC flask, and 1000 images were acquired with the hatching setup in Figure 3.1. A sample of all the images acquired from the mixture of just shells and just nauplii can be seen in Figure 5.15 and Figure 5.16, respectively. The figure of ”just” eggs contains different classes. Eggs, nauplii, noise and air bubbles can be observed in addition to shells, so it is not easy to separate them. The images of nauplii are, on the other hand, more successful. The majority of the images appear to have a distinct nauplii shape.

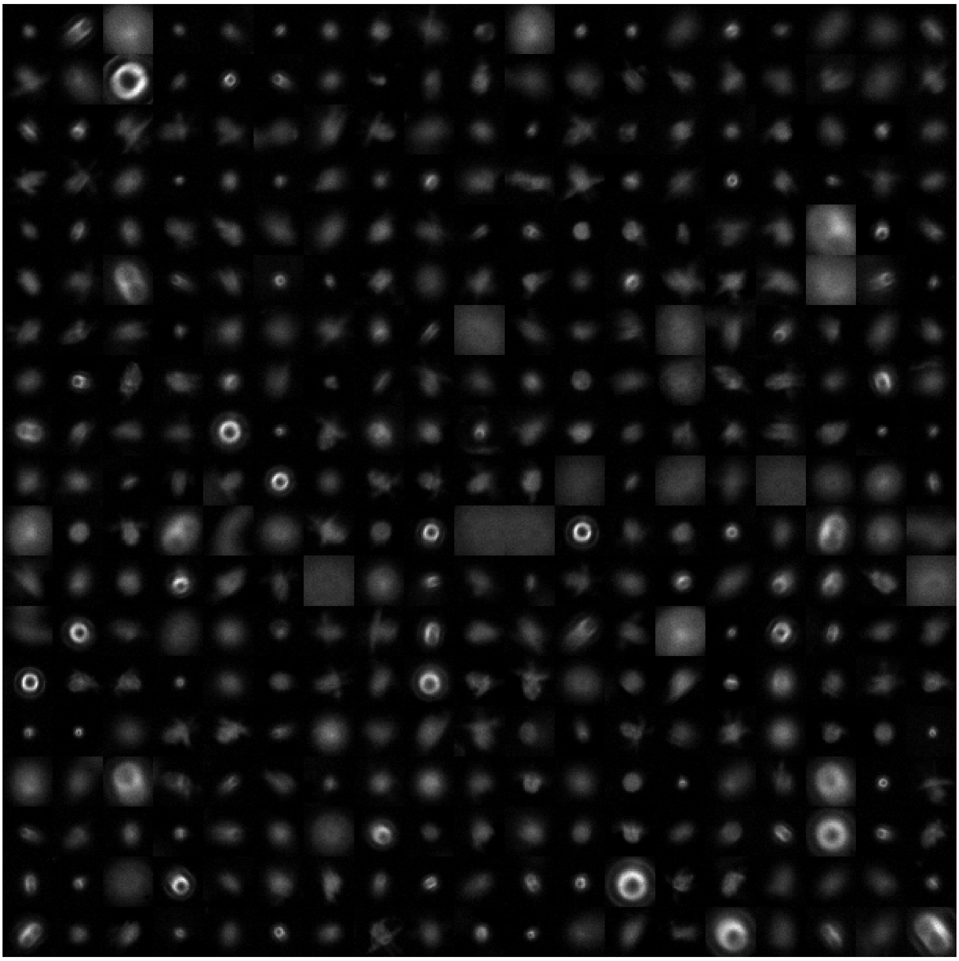


Figure 5.15: Images of the mixture of ”just” shells.

A comparison of the two was made to investigate how to distinguish between eggs and shells. In Figure 5.17, a sample of images and the brightened images is provided. The first 50 images on the first five rows are shells. The following 50 images, on the last five rows, are eggs. A difference can be observed between the two. The eggs stand out. However, observing a clear trend in the images of shells is more complex. Five images of what could be empty shells can be seen in Figure 5.18.

5.6.1 Main findings

- Challenging to distinguish shells.
- Images of distinct nauplii shape was acquired.

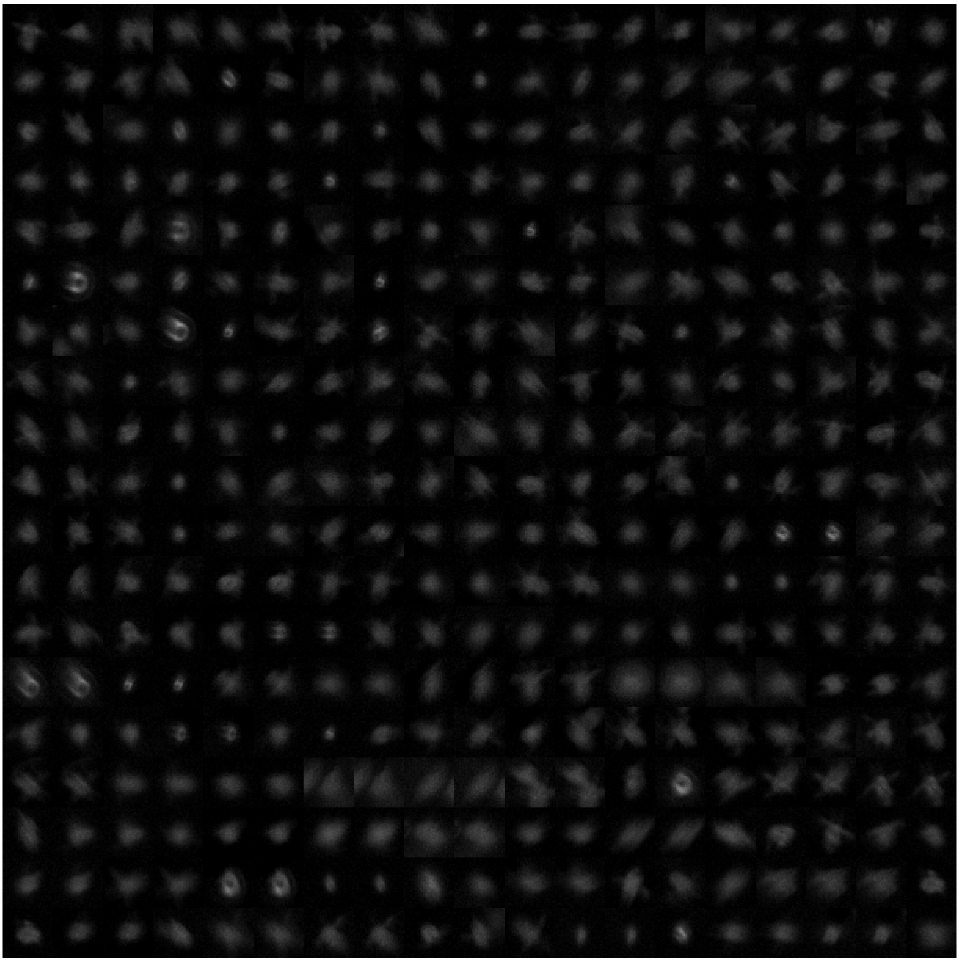


Figure 5.16: Images of the mixture of ”just” nauplii.

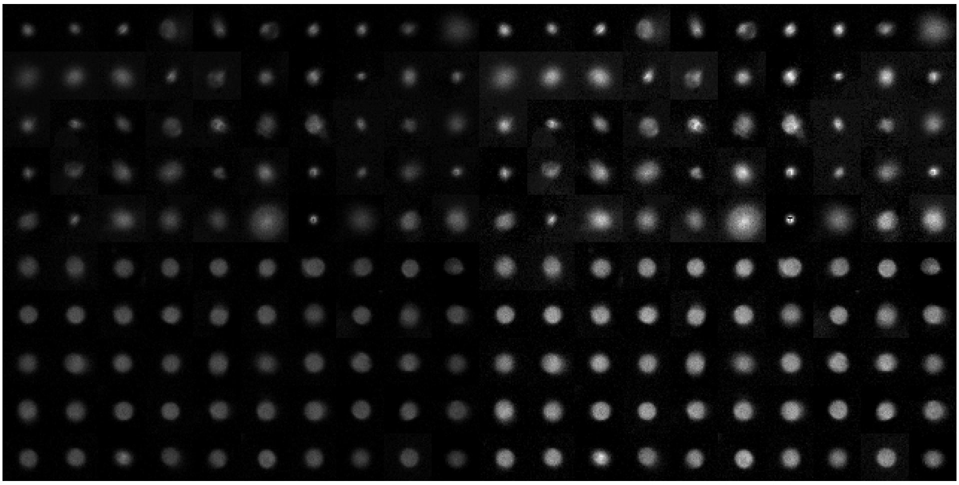


Figure 5.17: A comparison of eggs and shells, including a brightened version, where the first five rows are shells, and the next five columns are eggs.

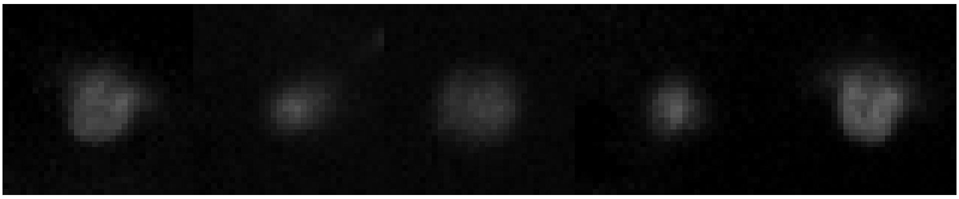


Figure 5.18: Five images of shells.

5.7 Dataset I

A dataset was made with the images acquired from Test IV (section 5.4). 434 images were annotated, with approximately the same number of images for each class - eggs, nauplii and unwanted. The labelled images can be seen in Figure 5.19.

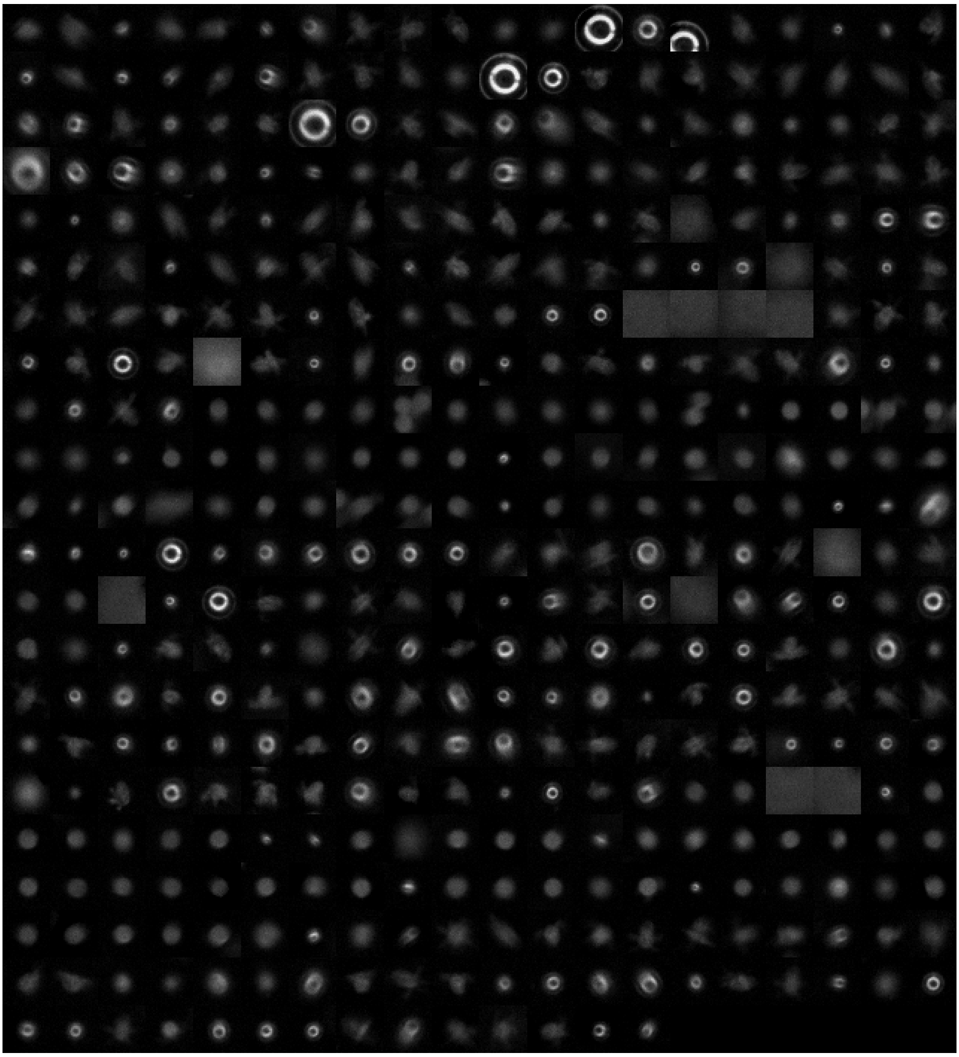


Figure 5.19: Dataset I consists of three classes - egg, nauplii and unwanted/noise.

The labelling was done by assuming eggs to be perfectly circular, nauplii to have a nauplii-like shape and the unwanted class with the rest. The last class consisted of air bubbles, noise, and an image of several eggs. VGG11, -13, -16 and -19 were tested with different epochs and batch sizes on the Dataset. The results of parts of the training executed can

Architecture	Epochs	Batch Size	Accuracy	Running time (min)
VGG11	30	16	84.5%	37
VGG11	30	32	86.8%	34
VGG11	35	16	89.9%	44
VGG11	40	16	93.8%	50
VGG11	500	16	93%	626
VGG13	4	64	33%	7
VGG13	30	16	81.4%	55
VGG13	35	16	88.3%	66
VGG13	40	16	92%	78
VGG16	25	16	94.6%	58
VGG16	30	16	90%	67
VGG16	35	16	95.3%	79
VGG16	40	16	94.6%	92
VGG16	100	16	93.8%	219
VGG19	30	16	93%	76
VGG19	35	16	91.5%	91
VGG19	40	16	90.7%	108
VGG19	100	16	93%	242

Table 5.1: Accuracy for different combinations of epochs and batch sizes for VGG11, VGG13, VGG16, and VGG19, including running times.

be seen in Table 5.1. When increasing the batch size, the running time decreases, which can be observed for VGG11, which was expected. VGG16 and -19 provide accuracy exceeding 90% in all training instances.

Learning curves are presented in Figure 5.20, with VGG11, -13, -16 and -19, respectively, to evaluate the network's performance during training. The same trend for the four networks can be observed. A tendency of convergence of the validation loss can be seen. A more noisy curve can be observed in the learning curves for the validation loss due to fewer images in the validation set.

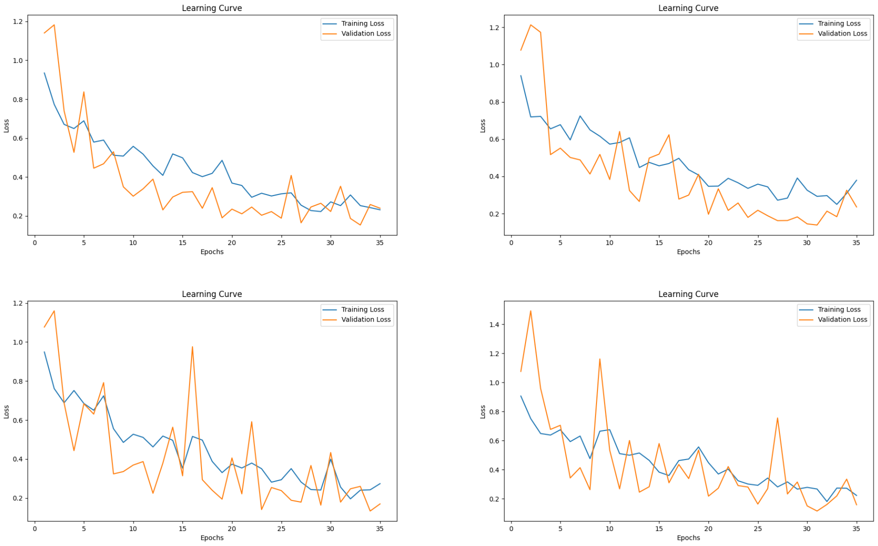


Figure 5.20: Learning curves of VGG-11, -13, -16 and -19 with 35 epochs and batch size 16.

To illustrate the model of overfitting, a learning curve with a large batch size (500) was made. In Figure 5.21, the validation loss curve is increasing, and the training loss is converging. An observation like this is a typical sign of a model overfitting.

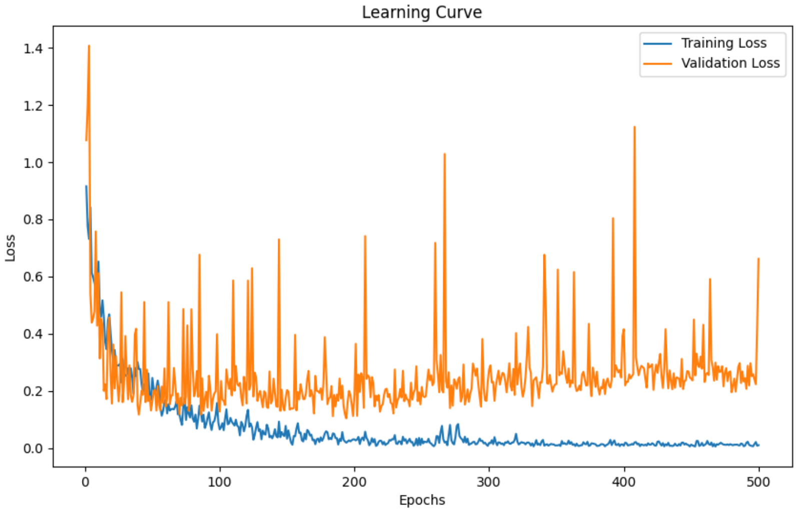


Figure 5.21: Learning curve illustrating overfitting of a VGG11-network with 500 epochs.

There were errors in the labelling in the first Dataset. The algorithm predicted the image to be unwanted, but the label was nauplii. A prediction of unwanted when the label was egg also occurred. This is a weakness in the Dataset, as the network needs to be trained on correct information.
Based on the results from Table 5.1, the learning curves, and the running times, the VGG16 structure was chosen as the ”best option”, with a batch size 16. The network structures perform well, with small margins between them. The different layers of the VGG16 network are illustrated using Netron and can be shown in Appendix C.

5.8 Dataset II

A new dataset was made to improve the labelling and training. The knowledge acquired from performing the training with Dataset I was a valuable experience in improving the Dataset. Three classes were utilised during the annotation process - eggs, nauplii, and air bubbles. Dataset II consists of 500 images of each, totalling 1500 images. Images of eggs and air bubbles were taken from the image capture process described in Test VI (section 5.4), and the images of nauplii came from Test VI (section 5.6). It was desirable to ensure correct labelling to ensure the network trained on correct data. The images constituting Dataset II can be viewed in Figure 5.22.

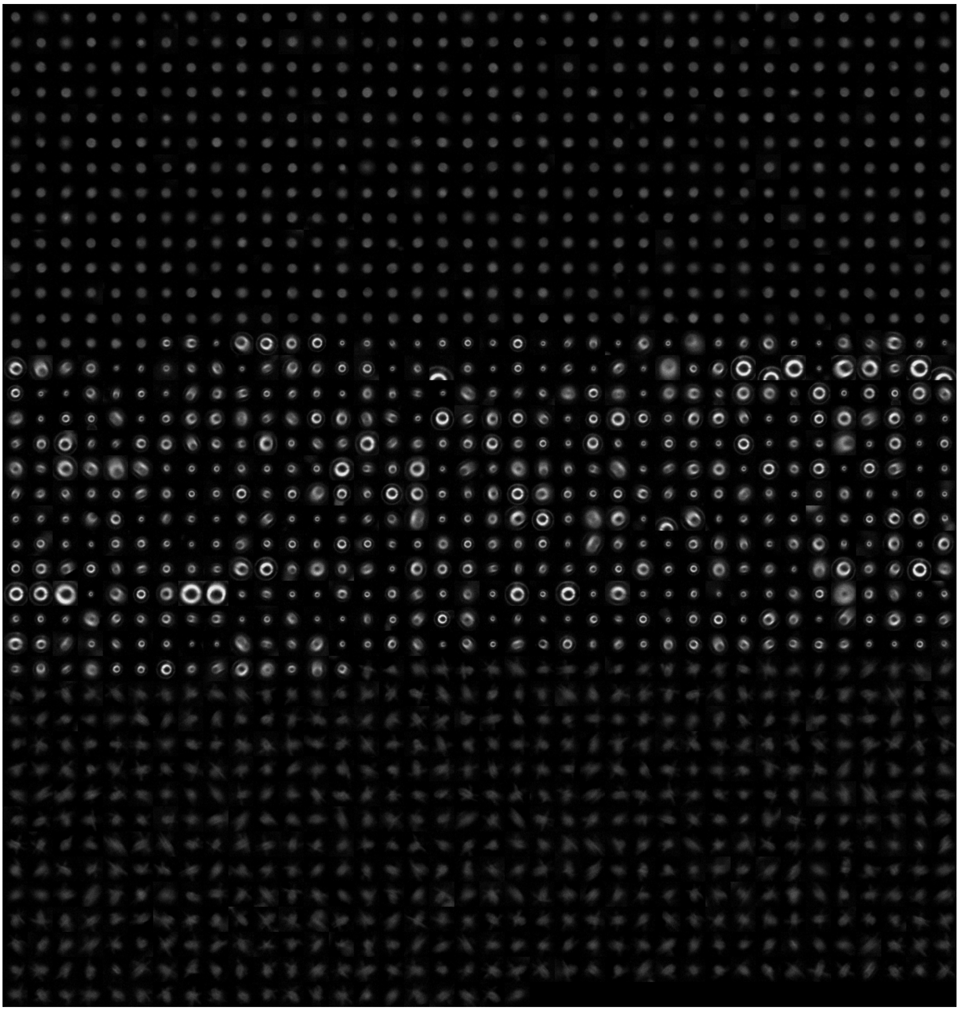


Figure 5.22: Dataset II consists of three classes - egg, nauplii and air bubbles.

A VGG16 architecture was utilised with a batch size of 16 in all the training processes for Dataset II. The only variable that changed across the runs was the number of epochs. The outcomes from the training process can be viewed in Table 5.2. It can be observed from the results that the network is capable of effectively learning when the images are accurately labelled.

From Figure 5.23, it can be observed that the validation loss converges more quickly, which may suggest that the network is learning at a faster rate. The observation implies that fewer epochs are needed compared to Dataset I.

Epochs	Accuracy	Running time (min)
5	95.11%	39
7	97.55%	55
10	98.67%	79
30	99.56%	234
35	99.56%	266

Table 5.2: Accuracy's for different epochs with VGG16 network including running times, with Dataset II.

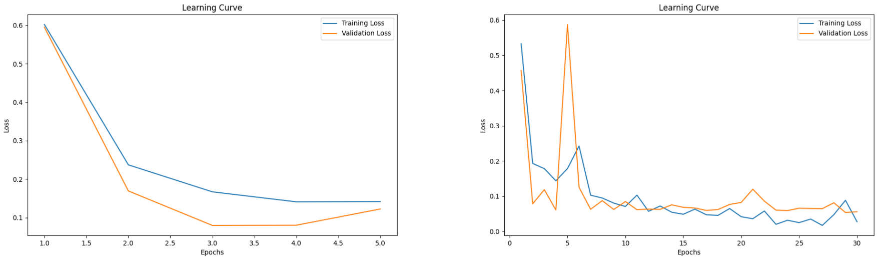


Figure 5.23: Learning curves of the training process with Dataset II using VGG16 and epochs of 5 and 30 respectively.

5.9 Dataset III

A class to handle noise is necessary to add to create a dataset capable of dealing with everything the blob detector extracts from the original images from the hatching sensor. Dataset III is built upon Dataset II but includes a fourth class - noise. This class was meant to contain everything that is neither egg, nauplii, or air bubble. Two hundred images of noise were annotated, leading to a slightly imbalanced dataset. The images of the noise added to the Dataset can be seen in Figure 5.24.

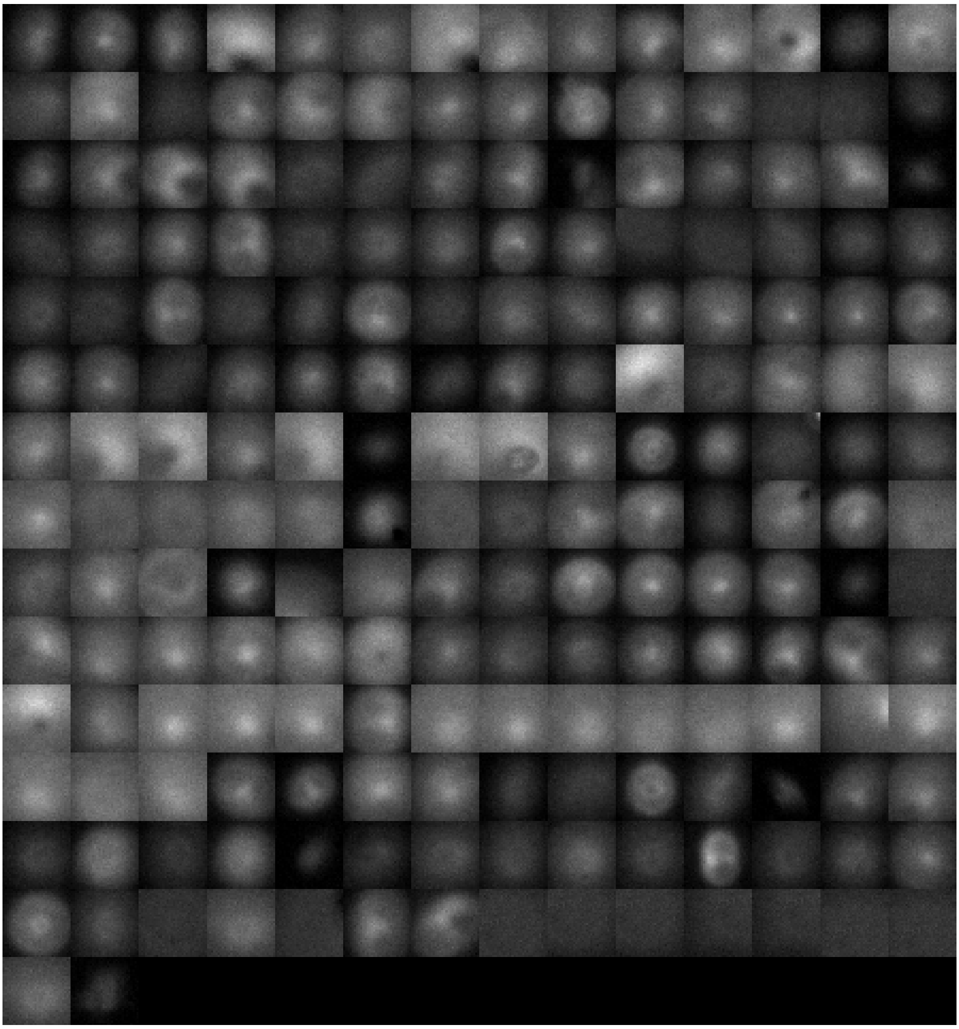


Figure 5.24: Images of the fourth class consisting of noise.

The training with Dataset III can be seen in Table 5.3. The accuracy is decreasing compared to Dataset II, but this is expected as a new class is involved. Some of the images of the noise look like air bubbles, which can confuse the model when predicting the class.

A confusion matrix from training with 10 epochs can be seen in Figure 5.25, where 0 is egg, 1 is nauplii, 2 is noise, and 3 is air bubbles. The matrix implies confusion for the network regarding classifying nauplii when the actual label was noise and air bubbles. This was a common trend observed in most of the confusion matrices. In Figure 5.26, learning curves from the training with 10 and 15 epochs can be seen. Interpreting the learning curves, epochs of 10-12 is reasonable.

Epochs	Accuracy	Running time (min)
2	94.89%	18
3	96.66%	27
7	96.66%	59
10	98.62%	82
12	96.27%	103
15	97.84%	132
25	98.62%	220
30	98.04%	267

Table 5.3: Accuracy's for different epochs with VGG16 network including running times, with Dataset III.

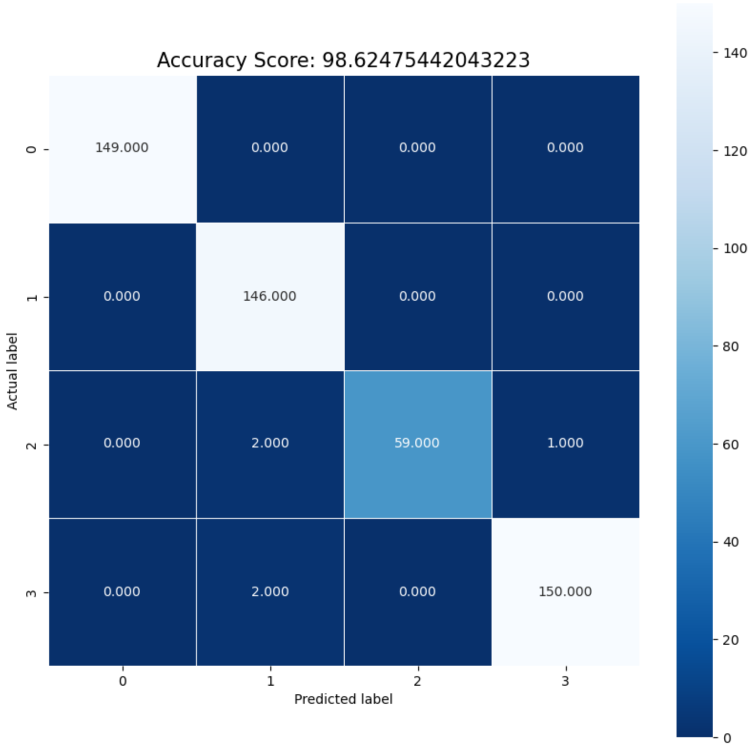


Figure 5.25: A confusion matrix of the training process with Dataset III, VGG16 and 10 epochs.

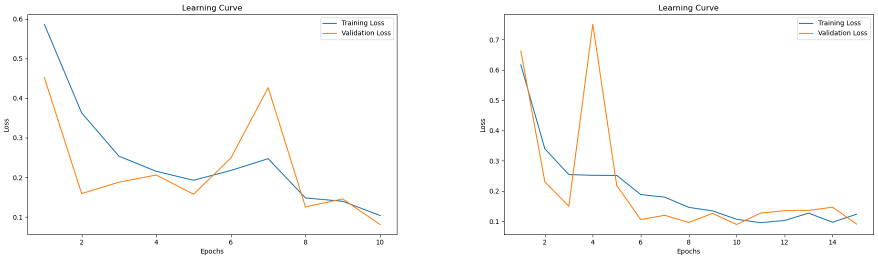


Figure 5.26: Learning curves of the training process with Dataset III using VGG16 and epochs of 10 and 15 respectively.

A lower learning rate was tested to improve the trained network's performance. A decrease in the learning rate allows the network to make finer adjustments to its weights, enabling the network to learn more complex patterns in the data accurately. The final trained model featured a change of the learning rate to 0.00025 and the number of epochs to 35. In Figure 5.27, the learning curve of the training process can be seen. Smaller spikes in the validation loss can be observed due to a lower learning rate. The training of the network achieved an accuracy of 99.02%, and the classification of the various classes can be viewed in Figure 5.28. It can be observed that the network is classifying all the instances of eggs correctly, and one nauplius is misclassified.

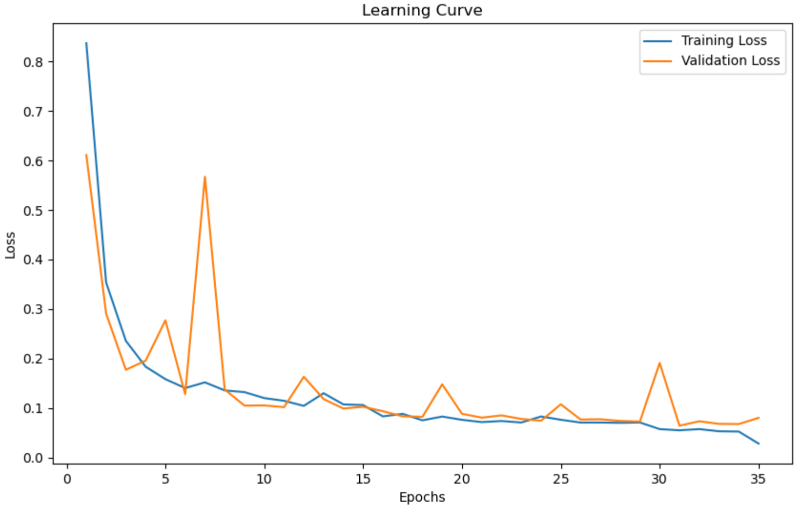


Figure 5.27: Learning curve of the training process with Dataset III using VGG16 and epochs of 10 and learning rate of 0.00025.

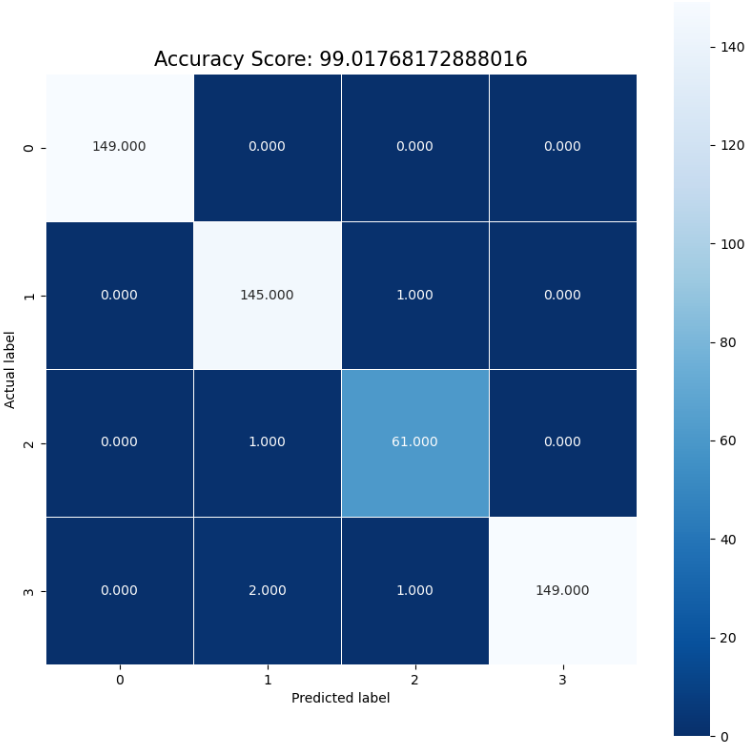


Figure 5.28: A confusion matrix from the training process with Dataset III used for testing the trained network.

5.10 Final test

Classifying the particles throughout a hatching process was desirable to test the trained model. Test sets were created to use entirely unseen data on the trained network. Images from Test 4 (section 5.4) and Test 6 (section 5.6) were utilised to generate these test sets. For each test, 100 images were annotated to represent a counting sample. The annotated particles in these datasets were labelled conservatively to avoid inaccuracies.

The trained model was first tested on the test data at 0 hours into the hatching process. Figure 5.29 shows an accuracy of 80.80% and 3 out of 78 eggs were misclassified.

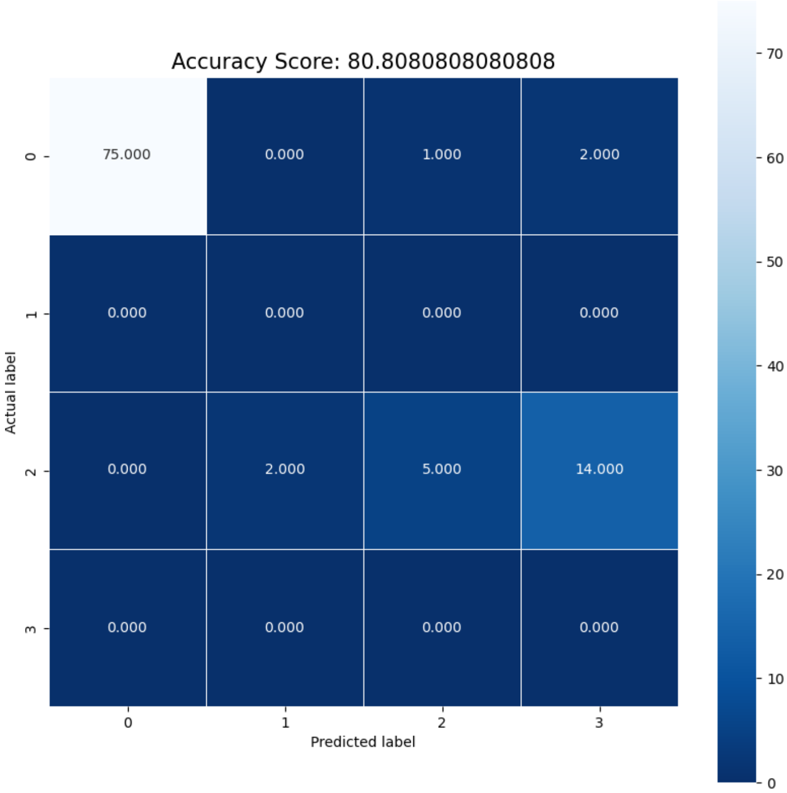


Figure 5.29: A confusion matrix from the classifications from 0 hours into the hatching process.

Twenty-six hours into the hatching process, eggs and nauplii were expected in the mixture. Figure 5.30 shows an accuracy of 40.4% and 20 out of 44 nauplii misclassified, a more significant proportion of misclassifications compared to eggs from 0h. The observed quantity of eggs being significantly less than nauplii after 26 hours causes an invalid representation of the hatching rate.

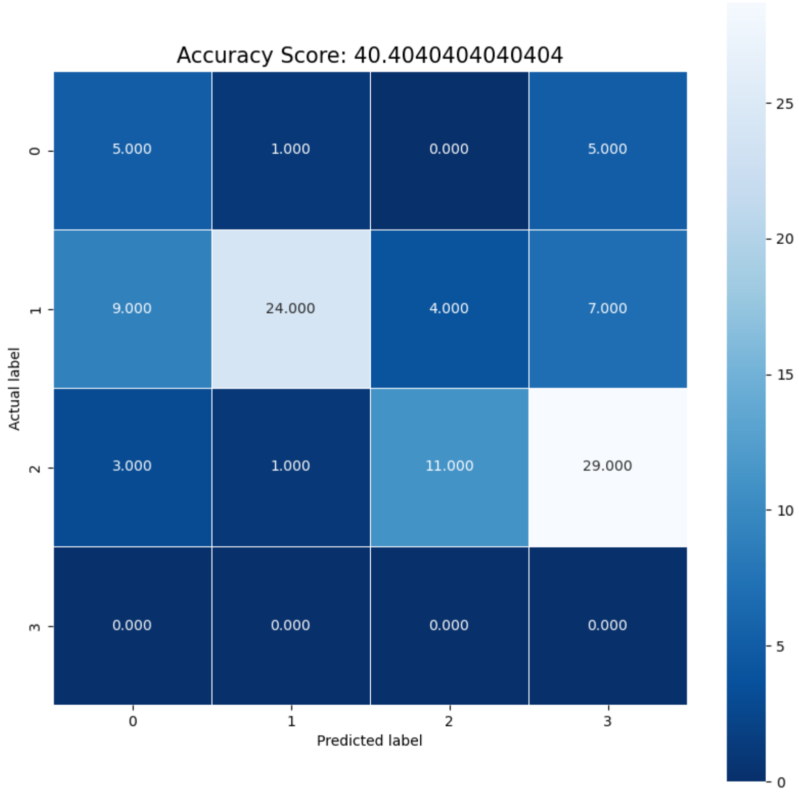


Figure 5.30: A confusion matrix from the classifications from 26 hours into the hatching process.

Two misclassified nauplii can be seen in Figure 5.31. The network is labelling nauplii as both egg and unwanted.

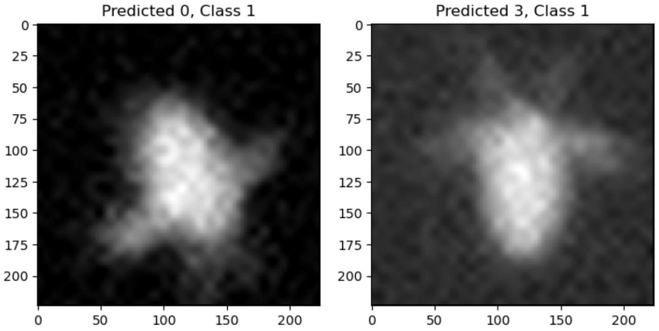


Figure 5.31: Misclassification of two images of nauplii after 26 hours. 0 is egg, 1 is nauplii, and 3 is unwanted.

Given the poor classification of nauplii, the data distribution was investigated. A test set consisting only of nauplii from Test IV (section 5.6) was created to examine how the same trained network performs on these images, which come from the same distribution the network was trained on. Figure 5.32 displays an accuracy of 86.87%, with 7 out of 87 nauplii misclassified. The accuracy is higher on the test set where the data has the same distribution compared to the test dataset from 26 hours into the hatching process.

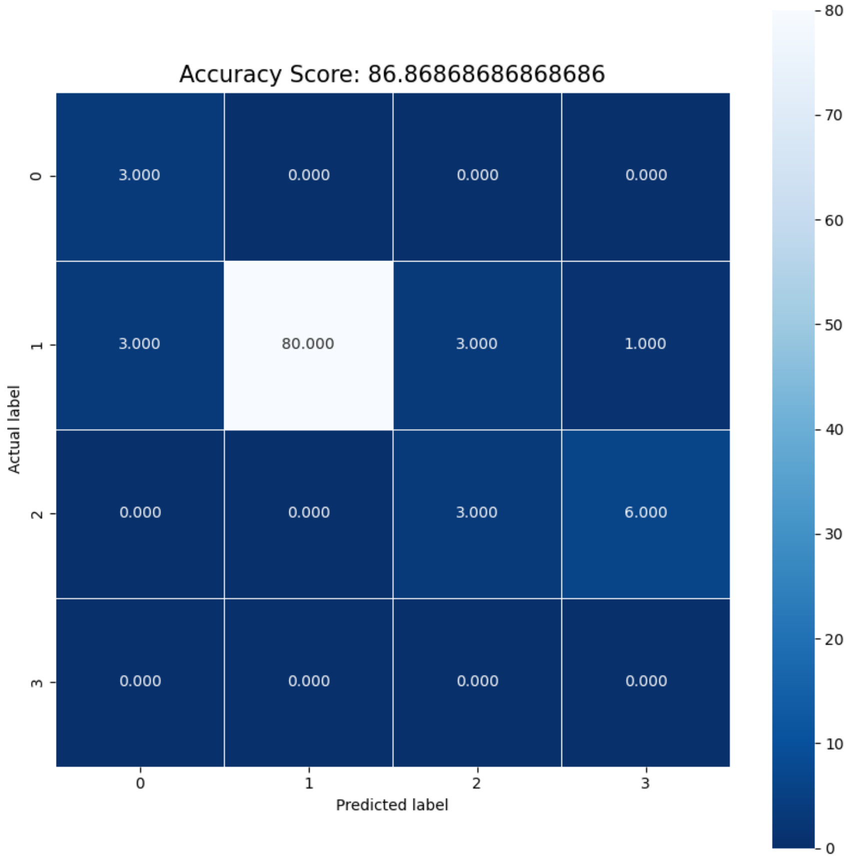


Figure 5.32: A confusion matrix from the classifications on nauplii from Test VI (section 5.6).

In Figure 5.33, a comparison of nauplii from different distributions can be seen. The left image is from testing done in Test VI (section 5.4), and the right image is from testing done in Test IV (section 5.6). A visual inspection of the two images presents minor differences in the image quality. The inspection indicates that the network is sensitive to differences in image quality.

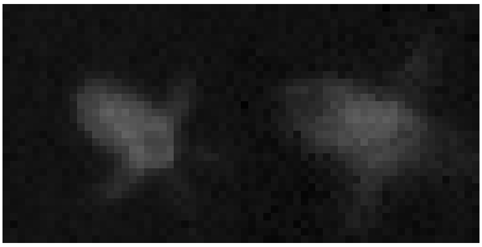


Figure 5.33: A comparison of images of nauplii from Test IV (section 5.4) and Test VI (section 5.6) respectively.

6

Discussion

6.1 Discussion

6.1.1 Image acquisition

The motion within the hatching tank during image acquisition significantly influences the quality. The exposure time and aperture needed depend on the movement of the particles. Water movement is necessary to create an environment conducive to copepod hatching. Several factors influence the water movement, impacting the quality of the acquired images.

In this project, aeration was used to make water movement. The degree of aeration was not quantitatively measured during testing. No tests were conducted to examine the intensity of the aeration. The turbidity influences the water circulation, which could lead to motion blur, resulting in blurry images, the air bubbles' size, and the hatching process's outcome.

When using aeration in combination with a NUNC flask as a hatching tank, sedimentation was observed in Figure 5.12. The tank's geometry influences the water circulation. Appendix B describes how to use a soda bottle upside down to prevent sedimentation during hatching, which is better suited to avoid sedimentation. A soda bottle has a curved surface, while a NUNC flask has a flat surface better suited for image acquisition. The depth of field provided by the NUNC flask presents a challenge associated with image acquisition. The flask's thickness, combined with the camera, led to a situation where not all particles within the tank could be in focus at the same time. This challenge results in certain particles falling out of focus. Ensuring that the camera and the hatching tank are parallel is essential to achieve the correct focus in the image.

Test V (section 5.5) investigated the effect of reduced motion in the tank and how it affects the homogeneity of the mixture. A homogeneous mixture is crucial when capturing images to ensure a representative depiction of the egg-to-nauplii ratio. Findings from this test suggest a potential to pause aeration during image capture, which may lead to longer exposure times and improved image quality. More than relying on a single dataset for analysis may be required, as the resulting values could be subject to random variations. Ideally, utilising three separate datasets would provide a more robust foundation for concluding how stopping the aeration affects a homogeneous mixture.

6.1.2 Annotating

An essential step in a ML process is the creation of a robust dataset. This project involved annotating images based on personal knowledge of copepods. Many images were discarded during the creation of the datasets. The discarding was done to avoid training the network with incorrect information.

Classifying nauplii proved challenging due to their orientation within the hatching tank. Nauplii could be misidentified as noise or undesired particles as dust, and vice versa. Egg shells were challenging to get a clear understanding of the shape. Therefore, dataset 2 was created based on particles identifiable as eggs, nauplii, or air bubbles. This ensured that the network did not train on inaccurate data. A hatching sensor must be capable of managing noise. Hence, in Dataset 3, a class for unwanted items was introduced. The intention behind this class was to contain anything that was not an egg, nauplii, or air bubble. The sensor should be able to classify everything that the blob detector captures. Annotation errors can occur, resulting in network training on incorrect data.

Motion blur and depth of field present challenges in accurate annotation and classification. It can be observed that some of the particles at the edges of the images are blurry, which makes it challenging. The lack of clarity in these images makes it particularly difficult to identify the actual objects. There are instances where the depth of field of the particle results in being out of focus. This applies to all particles so that the model can classify noise as an air bubble.

If clusters of eggs frequently appear, it might be necessary to count them accurately. In such cases, one could create categories for ”two eggs” and ”three eggs”. However, if these scenarios occur more often, it becomes easier to train such classes effectively. Alternatively, these clusters of eggs could be categorised within the ”one egg” class to at least account for one of the eggs. The accuracy of detecting these cases depends on the number of examples available during training. If these instances are rare, it might be wisest to overlook them. A better-suited hatching tank should reduce the amount of double and triple eggs.

Refining the parameters used by the blob detector could have narrowed down the particles selected by the annotator. This could have reduced the variation in image quality by excluding out-of-focus particles.

6.1.3 Machine vision

Hyperparameter tuning varies depending on the specifics of the task and the data being used. No extensive hyperparameter tuning has been conducted in the work of this thesis. It is common to start with a basic set of parameters as a starting point. The model performs with accuracies that can be used for a hatching sensor, so the focus was not to tune the model with the optimal hyperparameters. To improve the model's performance, data-augmentation techniques, pre-trained models, optimising the architecture, and adding regularisation and hyperparameter tuning could be performed.

It proved challenging to accurately label everything that the blob detector found without the certainty of giving the correct label. This is mainly due to the orientation of the nauplii, as discussed in subsection 6.1.2. While an attempt was made to annotate everything, much guessing was involved about what certain particles could be. As a result, a test dataset featuring labels for all detected particles in the annotator was not constructed. Narrowing the blob detector, as discussed in subsection 6.1.2, could benefit the ability of the model to classify all the particles. This could be done prior to fully automating the process.

The observations from the Final Test (section 5.10) indicate that the model is sensitive to variations in image quality. Theoretically, identical image quality should be maintained with an identical setup. However, testing demonstrated that the hatching sensor setup in chapter 3 does not produce consistent quality images. Factors such as the orientation of particles and depth of field affect the image results, as previously discussed. Training the network on a broad range of data is desirable to make it as robust as possible. This approach will create a generalised model resilient to new and unseen data, resulting in a robust hatching sensor.

6.1.4 Sources of error

This project has various potential sources of error, primarily due to its biological aspect. The work performed in this master's thesis depends on knowledge and expertise related to biology, which might have influenced the testing processes, particularly those involving taking samples and microscope usage.

Counting tests demand patience, and their validity can rely on the experience level of the person conducting the test. This could have affected the results from Test V (section 5.5). Distinguishing between eggs, shells, and nauplii can pose a challenge in a counting test, especially when they are closely packed and difficult to differentiate. On the other hand, the counting tests were conducted by the same person. Hence, all particles have been evaluated equally, so the ratio should be representative.

The hatching process was carried out in a NUNC flask for most of the experimental testing, which led to sedimentation (Figure 5.12) and non-optimal conditions for the copepods to hatch. Multiple double and triple eggs stuck together occurred (Figure 5.9). Such instances would be significantly reduced by conducting a hatching process where the aeration and hatching tank is optimal.

6.1.5 Final test

The purpose of the hatching sensor was to monitor the hatching process. To test the performance of the trained model, final testing was executed.

The idea behind the annotation of Dataset 3 was to use only 100% certain labels. Therefore, the images of nauplii were collected from Test 8, while the remaining images came from Test 4. An independent test set was created using only images from Test 6 to see if it improved the quality of the nauplii classification. As it turned out, the model performed more accurately on the images from the same distribution.

Consistency in the setup was prioritised each time data was collected to maintain similar image quality across the dataset. This was done because it was essential to have data as similar as possible, to ensure that new and unseen data was as close as possible to the data the model was trained on. Based on the tests, identical results were not achieved, and the model was sensitive to differences in image quality. This may indicate that the trained model needs to be generalised.

The observed quantity of eggs being significantly less than nauplii after 26 hours causes an invalid representation of the hatching percentage. More eggs were expected at this time of the hatching process. Observing this, along with the sedimentation observed in Figure 5.12, it is apparent that the NUNC flask and aeration do not serve as an adequate hatching tank where the mixture remains homogeneous. The hatching percentage was not reliable in this test.

The test dataset should consist of 200-250 images with approximately the same number of particles as in the manual counting samples to achieve a comprehensive counting sample.

A test to experiment on how the model performed on all the particles annotated by the annotator was investigated. During the making of such a dataset, it was shown to be challenging to annotate all the particles with certainty.

6.2 Further work

Throughout this project, several ideas for additional projects have emerged aimed at improving, validating, and further developing a hatching sensor. Based on the results of this study, several recommendations for future work have been compiled.

6.2.1 Hatching sensor

A more in-depth testing to validate the technique

1. Is it possible to train an adequately performing model on real images of objects with the shape of a triangle, square, circle and out-of-focus unknown form? The ground truth can be compared in this case since the exact number of particles is known.

2. Is it possible to train an adequately performing model on real images of objects with the shape of a sphere (egg), ”open” sphere (hatched egg), dust, air bubble, and out-of-focus unknown form?

3. Can an adequately performing model be trained on real images of real eggs, shells, nauplii, dust, air bubbles, and out-of-focus unknown form?

Camera, optics and image acquisition

- Use a non-circular aperture, creating a distinct non-circular shape when out of focus to filter out particles.
- Make a limitation on maximum size or brightness to filter out objects that are too large or out of focus.
- Test with different lenses and cameras to change the depth of field.
- Implement video analysis to classify particles using motion patterns and appearance.
- Capture images from a narrower depth range to mitigate the challenges posed by the depth of field.
- Validate the homogeneity test with three separate tests.
- Conduct how the amount of aeration influences the movement of the particles.
- Use a hatching tank with a shape where the mixture is homogeneous to avoid sedimentation.

Different illumination

- : Using light with different wavelengths, such as infrared or coloured light, could reveal information not apparent with visible light.
- Direct light might provide clearer particle contours and brightfield images.

The study by Braggins (2000) could serve as a starting point for investigating alternative light sources.

6.2.2 Machine vision

- Refine the blob detector to examine what particles are drawn from the annotator.
- The network should be trained on images from several different tests to improve its ability to generalise on new unseen data.
- Utilise a pre-trained model to label more efficiently, in which the pre-trained model can propose labels.
- Use transfer learning to train the network for the hatching sensor.
- Test other network architectures such as AlexNet, GoogleNet, ResNet-18, and DenseNet-201 (Alom et al., 2018).

6.2.3 Other

- Finalise the setup for the hatching sensor soft- and hardware to fully automate the process.

- Explore the possibility of predicting factors such as hatching success based on measurements from only the initial part of the hatching process.
- Consider expanding the application of the hatching sensor to other areas within biology that currently require manual counting tests.

7

Conclusion

The aim of this thesis was to explore, develop, and evaluate image acquisition and apply a VGG16 network to form a hatching sensor. The objective was to identify and classify copepod eggs and nauplii, to be able to monitor the hatching process. The work has contained image acquisition and annotation of images to form a dataset to train the network. The findings of 99% accuracy on a validation set suggest promising results.

The research has highlighted the importance of dataset quality for the performance of the model, which can be influenced by various factors including the quality of the annotations and the variability in the images. There are several variables related to the quality of the images, hatching tank design, and camera setup that have influenced the results of this project. Several challenges were identified and need to be addressed in future studies, such as investigations related to depth of field, handling of image noise, and hyperparameter tuning.

The study has made progress in monitoring the hatching process more effectively. Further improvements and refinements are needed to fully automate the process. This could significantly reduce manual labour and improve the precision of the hatching process of copepods. Future work should focus on addressing the identified challenges.

Bibliography

Agatonovic-Kustrin, S., Beresford, R., 2000. Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. J. Pharm. Biomed. Anal. 22, 717–727.

Alom, M.Z., Taha, T.M., Yakopcic, C., Westberg, S., Sidike, P., Nasrin, M.S., Van Esesn, B.C., Awwal, A.A.S., Asari, V.K., 2018. The history began from AlexNet: A comprehensive survey on deep learning approaches `arXiv:1803.01164`.

Bishop, C.M., 2006. Pattern Recognition and Machine Learning. Springer, New York, NY.

Braggins, D., 2000. Illumination for machine vision. Sens. Rev. 20, 20–23.

Bui, H.M., Lech, M., Cheng, E., Neville, K., Burnett, I.S., 2016. Using grayscale images for object recognition with convolutional-recursive neural network, in: 2016 IEEE Sixth International Conference on Communications and Electronics (ICCE), pp. 321–325. doi:`10.1109/CCE.2016.7562656`.

Deng, X., Liu, Q., Deng, Y., Mahadevan, S., 2016. An improved method to construct basic probability assignment based on the confusion matrix for classification problem. Inf. Sci. (Ny) 340-341, 250–261.

Drillet, G., Frouël, S., Sichlau, M.H., Jepsen, P.M., Højgaard, J.K., Joarder, A.K., Hansen, B.W., 2011. Status and recommendations on marine copepod cultivation for use as live feed. Aquaculture 315, 155–166.

Drillet, G., Jorgensen, N.O.G., Sorensen, T.F., Ramlov, H., Hansen, B.W., 2006. Biochemical and technical observations supporting the use of copepods as live feed organisms in marine larviculture. Aquac. Res. 37, 756–772.

Gonzalez, R.C., Woods, R.E., Masters, B.R., 2022. Digital image processing, third edition. J. Biomed. Opt. 14, 029901.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press. `http://www.deeplearningbook.org`.

Hansen, B.W., Drillet, G., Kozmer, A., Madsen, K.V., Pedersen, M.F., Sorensen, T.F., 2010. Temperature effects on copepod egg hatching: does acclimatization matter? J. Plankton Res. 32, 305–315.

Hansen, M.H., 2011. Effects of feeding with copepod nauplii (*Acartia tonsa*) compared to rotifers (*Brachionus ibericus*, *Cayman*) on quality parameters in Atlantic cod (*Gadus morhua*) larvae. Master's thesis. Norwegian University of Science and Technology, Department of Biology. Trondheim.

Hui, L., Belkin, M., 2020. Evaluation of neural architectures trained with square loss vs cross-entropy in classification tasks `arXiv:2006.07322`.

Jepsen, P.M., 2014. Copepods as live feed - optimisation and use in aquaculture. Master's thesis. Roskilde University. Denmark.

Johnson, S., 2006. Stephen Johnson on Digital Photography. O'Reilly Media, Sebastopol, CA.

Leandro, S.M., Tiselius, P., Queiroga, H., 2006. Growth and development of nauplii and copepodites of the estuarine copepod *Acartia tonsa* from southern europe (ria de aveiro, portugal) under saturating food conditions. Mar. Biol. 150, 121–129.

Lecun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proc. IEEE Inst. Electr. Electron. Eng. 86, 2278–2324.

Luce, R.D., 1977. The choice axiom after twenty years. J. Math. Psychol. 15, 215–233.

Mitchell, T., 1997. Machine Learning. McGraw-Hill Professional, New York, NY.

Mohr, F., van Rijn, J.N., 2022. Learning curves for decision making in supervised machine learning – a survey `arXiv:2201.12150`.

Nair, V., Hinton, G.E., 2010. Rectifed linear units improve restricted boltzmann machines, in: Proceedings of International Conference on Machine Learning (ICML), pp. 807–814.

Ozeki, M., Okatani, T., 2015. Understanding convolutional neural networks in terms of category-level attributes, in: Computer Vision – ACCV 2014. Springer International Publishing, Cham, pp. 362–375.

Pak, M., Kim, S., 2017. A review of deep learning in image recognition, in: 2017 4th International Conference on Computer Applications and Information Processing Technology (CAIPT), pp. 1–3. doi:`10.1109/CAIPT.2017.8320684`.

Potmesil, M., Chakravarty, I., 1982. Synthetic image generation with a lens and aperture camera model. ACM Trans. Graph. 1, 85–108.

Pratt, W.K., 2007. Digital Image Processing: PIKS Inside. John Wiley & Sons.

Rosasco, L., De Vito, E., Caponnetto, A., Piana, M., Verri, A., 2004. Are loss functions all the same? Neural Comput. 16, 1063–1076.

Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition `arXiv:1409.1556`.

Solomon, C., Breckon, T., 2011. Fundamentals of digital image processing: A practical approach with examples in matlab. 1 ed., Standards Information Network.

Sonka, M., Hlavac, V., Boyle, R., 2013. Image processing, analysis and machine vision. 1993 ed., Springer, New York, NY.

Sugata, T.L.I., Yang, C.K., 2017. Leaf app: Leaf recognition with deep convolutional neural networks. IOP Conf. Ser. Mater. Sci. Eng. 273, 012004.

Yaser, A.M., 2012. Learning from Data: A Short Course. AMLBook.

Appendix

A Table of data collected for Test V - Homogeneity test

Tank	Eggs	Shells	Nauplii	Hatching Rate (%)
4 mL (not used)	25.67	81.00	98.00	79.25
4.5 mL tank (Original density: 276.6 eggs/mL)				
Continuous aeration	70.33	173.33	149.33	67.98
3 s without aeration	66.40	153.80	155.00	70.01
6 s without aeration	68.20	169.20	148.00	68.46
10 s without aeration	57.80	152.00	163.40	73.87
15 s without aeration	76.40	140.00	158.00	67.41
20 s without aeration	70.40	148.40	165.60	70.17
5 mL tank (Original density: 307.3 eggs/mL)				
Continuous aeration	65.00	178.00	167.00	71.98

Table 7.1: Summary of occurrences of eggs, shells and nauplii, and the hatching rates for different conditions.

B User manual - how to hatch and harvest your copepods

User manual

How to hatch and harvest your copepods

PREPARING YOUR HATCHING TANK

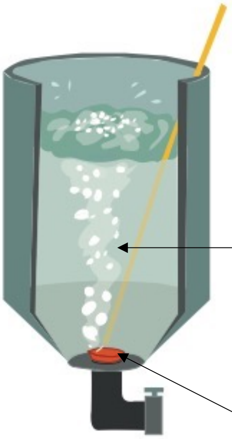
COPEPOD EGGS sediment easy, and sedimentation of eggs may lead to reduced hatching. To prevent the eggs from sedimenting, pay attention to:

- **Tank design**
- **Aeration**
- **Blocking all entrances**

Hatching time depends on the water temperature:

26 °C: 24h hatching
21 °C: 48h hatching

Choose the temperature closest to what is in your production tank



Tank design:
Use tanks with a **conical** or **rounded** bottom to ensure good circulation.

- Avoid tanks with a flat bottom

Aeration:
Heavy aeration, similar to when hatching Artemia, ensures a **good circulation** in your tank and prevent sedimentation of eggs:

- Use an open-ended tube to create **big bubbles**
- Place the tube at the lowest part of the tank

Blocking all entrances:
Eliminate **ALL** openings where eggs can sediment

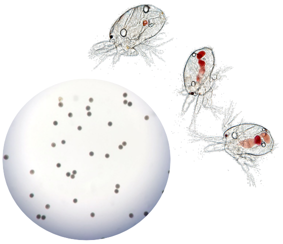
HATCHING CONDITIONS

	26 °C / 79 °F	21 °C / 70 °F
Harvest copepods after:	24 hours	48 hours
Salinity	15 – 40 ppt	15 – 40 ppt
pH	7,8 – 8,5	7,8 – 8,5
Oxygen	> 80% saturation	> 80% saturation
Light	Not necessary	Not necessary
Density	< 1000 ml^{-1}	< 1000 ml^{-1}
Water exchange	0 %	0 %

CFEED
www.cfeed.no

Page 1 of 2

1: RECEIVING AND STORING EGGS



The eggs are transported on ice in a Styrofoam box. Upon arrival:

- Check that ice is still present in the box to ensure that the cool chain has been intact.
- **Store the eggs at 1-4 °C / 34-39 °F.**
- Use the eggs within a month after arrival.
  - For longer storage, contact CFEED.

2: REMOVAL OF STORAGE MEDIUM

The eggs come in a storage solution containing clay that needs to be removed:

- Remove the bottle you would like to hatch from the cool storage.
  - If hatching only part of a bottle: Shake the bottle until the clay and eggs are evenly mixed. While it is mixed, transfer the amount of eggs you want to hatch to a separate container. Place the rest of the bottle back into cool storage.
- Pour the clay mixture into a **50 µm sieve** that will allow the clay to pass through while the eggs remain in the sieve.
- Rinse with seawater until the eggs are clean. Transfer the eggs to a bucket.

3: DISINFECTION

EQUIPMENT:

- A bucket filled with 10L temperate sea water.
- Clean eggs where the storage medium has been removed.
- Aeration for mixing the eggs and chemicals.

DISINFECTION PROCESS:

- Disinfect for 10 minutes using **4 ml NaOCl (14%)**. Aerate and stir to ensure that all eggs are in contact with the chlorine.
- Add dissolved **Na_2SO_3 (8g)** to neutralize the chlorine. Aerate and stir to ensure that all eggs are in contact with the Na_2SO_3. Leave for 10 minutes.
- **Transfer your disinfected eggs into your prepared hatching tank (see page 1 for more information).**

4: LEAVE THE EGGS TO HATCH

Hatching time is dependent on water temperature. See page 1 for more information.

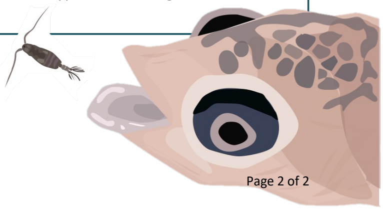
5: CONCENTRATING AND HARVESTING YOUR COPEPODS

Concentrate the hatched copepods before transferring them to your fish larval tank:

- Close off aeration for 20 minutes to let unhatched eggs sediment. **Flush quickly**.
- Transfer the copepods to you concentrator.
  - **Mesh size: < 65 µm**.
- **Aerate well** while concentrating:
  - Prevent the copepods from clogging the mesh.
  - No addition of O_2 is necessary.
- If the aeration is good the nauplii can be kept at a density of **15000 ml^{-1}** for up to **24 hours** at **5-6 °C**.
  - For longer storage time, do not exceed the limit of 500 per ml. Use the copepods within 24 hours.

Tip! Giving the copepods microalgae is a great way to **increase the visibility** for the hunting fish larvae:

- Add a small amount of **live microalgae** when the copepods are in the concentrator and wait for 10-15 minutes. The guts will be filled and the nauplii are ready to be fed to the fish.
- Contact CFFED if you would like to know more about which types of microalgae to use.



CFEED
www.cfeed.no

Page 2 of 2

C VGG16 architecture

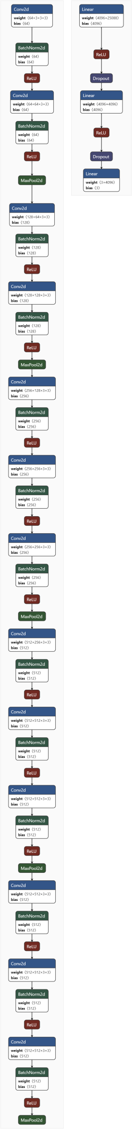


Figure 7.1: The structure of the VGG16 network used.

NTNU

Norwegian University of
Science and Technology