

TALLINN UNIVERSITY OF TECHNOLOGY  
School of Information Technologies

Moutaz Mohamed Fathy Ahmed Aboseada 184610IASM

# **UNDERWATER OBJECT DETECTION USING DEEP LEARNING**

Master's Thesis

Supervisor: Jeffrey Andrew  
Tuhtan  
PhD.

Co-Supervisor: Mohamed Walid  
Remmas  
MSc.

Tallinn 2020

TALLINNA TEHNIKAÜLIKOOL  
Infotehnoloogia teaduskond

Moutaz Mohamed Fathy Ahmed Aboseada 184610IASM

# **VEEALUNE OBJEKTITUVASTUS KASUTADES SÜVAÕPET**

Magistritöö

Juhendaja: Jeffrey Andrew  
Tuhtan  
Doktorikraad.

Kaasjuhendaja: Mohamed Walid  
Remmas  
Magistrikraad.

Tallinn 2020

## **Author's declaration of originality**

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Moutaz Mohamed Fathy Ahmed Aboseada

[16-12-2019]

## Abstract

Earth is a water planet, and fish is one of its's most valuable ecological and economic resources. Internationally, the fisheries industry for both marine and freshwater species reached 143 USD billion in 2016, producing a total of 171 million metric tonnes of fish biomass. Due to climate change and increasing urbanisation, aquatic biodiversity is declining rapidly, putting global fisheries at considerable risk. Methods to effectively monitor fish require passive underwater systems such as sonar and cameras. Due to the rapid growth of the aquaculture industry, camera-based systems are needed which can work in shallow depths or with high fish densities where sonar systems do not work well. In order to make these systems economically viable, it is necessary that cameras include automated image processing methods which can detect fish with minimum human intervention.

The main objective of this thesis is to develop and test image processing methods for image enhancement and automated fish detection. Images are taken from field studies in both freshwater river environments as well as salmon aquaculture sea cages. A new library of field imagery was created in this work based on 7,162 original images. The images included poor lighting conditions, debris, and high levels of turbidity and were all taken from field sites in freshwater and marine environments, and which are often significantly different than many of those found in the scientific literature.

Underwater object detection methods are not as developed as their terrestrial counterparts. The underwater environment is physically very different from air and requires additional algorithms to improve the imagery. Specifically, in this work, it is shown that underwater image enhancement can improve correct fish detection by increasing the mean average precision (mAP) by 5.1% and intersection over union (IOU) by 3.8%. After enhancement, deep learning networks are used to perform fish detection and identification trained on a balanced dataset with augmented images of 64,458 images. Finally, fish detection with image enhancement performance reached  $mAP = 75.11\%$  and  $IOU = 64.95\%$ , and future

improvements and limitations are provided based on the experience and findings of the thesis work.

This thesis is written in English language and is 59 pages long, including 6 chapters, 22 figures and 19 tables.

## List of abbreviations and terms

TUT	Tallinn University of Technology
YOLO	You Only Look Once
mAP	Mean Average Precision
IOU	Intersection Over Union
RGB	Red Green Blue Channels
SGD	Stochastic Gradient Descent
CNN	Convolution Neural Network
R-CNN	Region Convolution Neural Network
F4K	Fish for Knowledge
SVD	Singular Value Decomposition
CLAHE	Contrast Limited Adaptive Histogram Equalization
LAB	L: Light, a: Red/Green Value and B: Blue/Yellow Value
RELU	Rectified Linear Unit
2D	Two Dimensions
3D	Three Dimensions
GPU	Graphics Processing Unit
RTX	Real-Time Ray Tracing
CLAHS	Contrast Limited Adaptive Histogram Specification
TADA	Turn Angle Distribution Analysis
COCO	Microsoft Common Objects in Context
Pascal VOC	Pascal Visual Object Classes
R-CNN	Region-based Convolution Neural Network

## Table of contents

Author’s declaration of originality .....	3
Abstract.....	4
List of abbreviations and terms .....	6
Table of contents .....	7
List of figures .....	9
List of tables .....	11
1 Introduction .....	13
2 Background.....	17
2.1 Image Processing.....	17
2.1.1 Underwater Imagery .....	18
2.2 Object Detection Using Deep learning.....	21
3 State of the Art.....	25
3.1 Underwater Object Detection .....	25
3.1.1 Underwater Object Detection Based on Object Characteristics.....	25
3.1.2 Underwater Object Detection Based on Deep Learning .....	26
3.2 Underwater Image Restoration and Enhancement .....	27
3.2.1 Image Restoration.....	27
3.2.2 Image Restoration by Synthesising and Deep Learning.....	30
3.2.3 Image Enhancement .....	32
3.2.4 Image Restoration Compared to Image Enhancement .....	35
4 Digital Image Processing.....	36
4.1 Emphasis Homomorphic Filtering.....	38
4.1.1 Image Illumination .....	38
4.1.2 Frequency Domain of an Image .....	39
4.1.3 Filtering Image in the Frequency Domain.....	40
4.1.4 Proposed Filtering Method .....	45
4.1.5 Emphasis Homomorphic Filtering Results.....	46
4.2 White Balancing .....	48
4.2.1 White Balancing Proposed Method.....	48
4.2.2 White Balancing Results .....	49
4.3 Temporal Coherent Noise Reduction .....	51

4.4 Weights .....	52
4.4.1 Laplacian Contrast Weight .....	52
4.4.2 Local Contrast Weight.....	52
4.4.3 Saliency Weight.....	52
4.4.4 Exposedness Weight.....	52
4.4.5 Weights Results .....	52
4.5 Multicast Fusion .....	54
4.6 Image Enhancements Results .....	54
5 Underwater Object Detection .....	56
5.1 Deep Learning Network Training Settings.....	56
5.2 Dataset .....	59
5.2.1 Original Datasets Images.....	59
5.2.2 Dataset Image Augmentation .....	60
5.2.3 Final Dataset Information.....	63
5.3 Fish Detection Using YOLO Training and Results.....	63
6 Conclusion .....	70
7 Bibliography .....	72
Appendix 1 – Fish Image Dataset Examples.....	74
Appendix 2 – Underwater Image Enhancement Results.....	76

## List of figures

Figure 1. Fish production increased rapidly from 1950 to 2016 in terms of fish capturing and fish aquaculture, after [1].	13
Figure 2. Fish World trade exports in USD billions with the top 10 countries showing a comparison between 2006 and 2016, indicating that the fisheries industry is growing very fast from 86 USD billions in 2006 to 143 USD billions in 2016, after [1].	14
Figure 3. Water surface effects on light propagation as when the sea is calm, the light will create crinkles inside water, while if sea surface rough the light will diffuse, after [13].	18
Figure 4. Underwater light propagation, as incident light hits the water surface, horizontally polarised light will be reflected to outside and vertically polarised light will penetrate underwater, and it is worth mentioning that blue light usually is the most Rayleigh scattered light, after [13].	19
Figure 5. Wavelength-dependent attenuation coefficients of the Jerlov water types after [6], [14] and [15].	20
Figure 6. Illustration of underwater colour absorption, showing that red is the first to vanish at 5 meters whereas blue may persist up to 60 meters, after [5].	20
Figure 7. Comparing performance of different deep learning algorithms trained on VOC database based on mean average precision (mAP) at confidence score more than 50% versus the frame per seconds (FPS), showing that MASK RCNN + ResNext101 has the best mAP, but it was not built for quick detection, on the other hand, YOLOv3 SPP offer high mAP with higher FPS, After [17], [18].	22
Figure 8. Comparing performance of different deep learning algorithms trained on COCO database based on mean average precision (mAP) at confidence score more than 50% versus the frame per seconds (FPS), showing that MASK RCNN + ResNext101 has the best mAP, but it was not built for quick detection, on the other hand, YOLOv3 offer high mAP with higher FPS, After [17], [19] and [20].	22
Figure 9. YOLO version three structure with an example on a chub fish developed based on the configuration file and after [19], [21].	24
Figure 10. Block diagram in [5] for the proposed image restoration algorithm.	29
Figure 11. UWCNN model deep learning block diagram, after [6].	31

Figure 12. The proposed approach in [7] for image enhancement, as input image will be subjected to processing in two paths for colour correction and sharpening, then output image will be acquired by fusion.....	34
Figure 13. The proposed algorithm for underwater image enhancement, after [7] and [8], as input image will be subjected to processing in two paths for colour correction and sharpening, then output image will be acquired by fusion. ....	37
Figure 14. The image in the frequency domain where the low frequencies are in the centre of the image. Frequencies tend to increase as the distance from the centre increases. For every frequency, the magnitude in dB for the rate of change in the spatial domain. ....	40
Figure 15. Butterworth bandpass filter where a higher cut-off the frequency $\sigma H = 20$ , and the lower cut-off frequency $\sigma L = 4$ is on the left, and on the right with the same settings is the emphasis Butterworth bandpass Filter where $\alpha = 0.5$ and $\beta = 1.5$ , showing frequency range with different amplification variation. ....	42
Figure 16. Gaussian high-pass filter where cut-off frequency = 20 is on the left, and on the right with the same settings is the emphasis Gaussian high-pass filter where $\alpha = 0.5$ and $\beta = 1.5$ , showing frequency range with different amplification variation. ....	42
Figure 17. Butterworth high-pass filter where cut-off frequency = 20 and filter order 2 is on the left, and on the right with the same settings is the emphasis Butterworth high-pass filter where $\alpha = 0.5$ and $\beta = 1.5$ , showing frequency range with different amplification variation. ....	42
Figure 18. Emphasis homomorphic filtering proposed algorithm block diagram.....	46
Figure 19. Learning rate adjustment per iterations showing the different values of learning rate starting by burn-in stage from iteration zero to 300 <sup>th</sup> values are increased from 0 to 0.001 and at the 1500 <sup>th</sup> iteration changed to 0.0001 till iteration 2,500 <sup>th</sup> changed to 0.00001 for better tuning of weights. ....	64
Figure 20. Average loss during training per iterations for training with and without image enhancement, showing the average loss from 0 to 0.5 for better representations. ....	65
Figure 21. Mean average precision (mAP) per iterations from the 1206 <sup>th</sup> iteration to 5000 <sup>th</sup> iteration, comparing mAP for training with image enhancement increased by 5% more than training without image enhancement.....	66
Figure 22. Intersection over Union (IOU) per iterations from the 408 <sup>th</sup> iteration to 5000 <sup>th</sup> iteration, showing IOU for training with image enhancement has increased IOU by 3%. ....	66

## List of tables

Table 1. An example of processing an input image using the algorithm in [5], where the input image dominant colour is green. Using original settings in this algorithm will provide a failed restored image. For successful restoration need to modify the algorithm to have the green as the dominant colour and modify the stretching thresholds of the red colour.....	30
Table 2. Applying an emphasis homomorphic filtering with different filters of Gaussian high-pass filters and Butterworth high-pass and bandpass filters, each column represents one filter type settings, input and output image with their histogram, Light channel in LAB colour space, frequency domain. ....	44
Table 3. Results of emphasis homomorphic filter using the bandpass Butterworth filter to eliminate non-uniform illumination. ....	47
Table 4. White balance using traditional simplest colour balance and proposed new white balanced, showing that the simplest colour balance output appears redder than the new white balance. As the new white balance algorithm is more optimized per channel and for every image.....	50
Table 5. Temporal coherent reduction output using contrast limited adaptive histogram equalization (CLAHE).....	51
Table 6. Weights results for every path, for colour correction path and sharpening path, will calculate the weights for Laplacian contrast, local contrast, saliency and exposedness weights.....	53
Table 7. Digital image enhancement proposed results showing input image and the final output image. ....	54
Table 8. Results of image enhancement using the proposed algorithm and algorithms in [5], [7], showing input and output image for each algorithm.....	55
Table 9. Fish object detection training settings. ....	58
Table 10. Images types of fish in the dataset without image enhancement and with image enhancement, samples used in training like images that have good fish details, overlapping fish and fish like the background.....	60

Table 11. Image processing operations configuration to introduce image augmentation. .....	61
Table 12. Image augmentation will be used for every image in the datasets to generate four images. Every image will be the result of shown different image processing operations. ....	62
Table 13. Image augmentation results of example original image.....	62
Table 14. Fish images datasets used for training and testing for YOLO training with and without image enhancement. ....	63
Table 15. Example of fish detection from the test images in dataset shows that fish detection using image enhancement have higher confidence score and better detection of the fish (better IOU compared to ground truth).....	67
Table 16. Example of fish detection on images not from the training dataset or test datasets, it is from a different video set, fish detection using image enhancement have higher confidence score and better detection of the fish (better IOU compared to ground truth). ....	68
Table 17. Best training results for both training with and without image enhancement.	69
Table 18. Images types of fish in the dataset without image enhancement and with image enhancement, samples used in training like images that have a good fish details, foggy fish shape, overlapping fish, fish like the background, parts of fish. ....	74
Table 19. The proposed image enhancement algorithm results on different scenes showing input and output image.....	76

# 1 Introduction

Fish are both an ecological and economic resource, and there is an urgent and growing need to monitor their quantity and health status, especially considering the rapid decrease in aquatic biodiversity. Primarily the adverse effects of climate change and urbanisation have put stress on fisheries worldwide. Camera-based monitoring of fisheries is the most widely-used method to observe fish and are installed in oceans, sea, rivers and aquaculture facilities to observe and study fish and evaluate their health and welfare.

According to the Food and Agriculture Organization of United Nations (FAO), the fish industry in 2016 reached 171 million tons (see Figure 1) to be traded with USD 143 billion as fish and fish products, which reflect the importance of fish industry worldwide. The fish industry grew from 86 USD billion in 2006 to 143 USD billion in 2016, as presented in Figure 2. In fact, some countries like Cabo Verde, Greenland, Iceland exports of fish is essential to their economy as it exceeds 40% of the total merchandise trade value. [1]

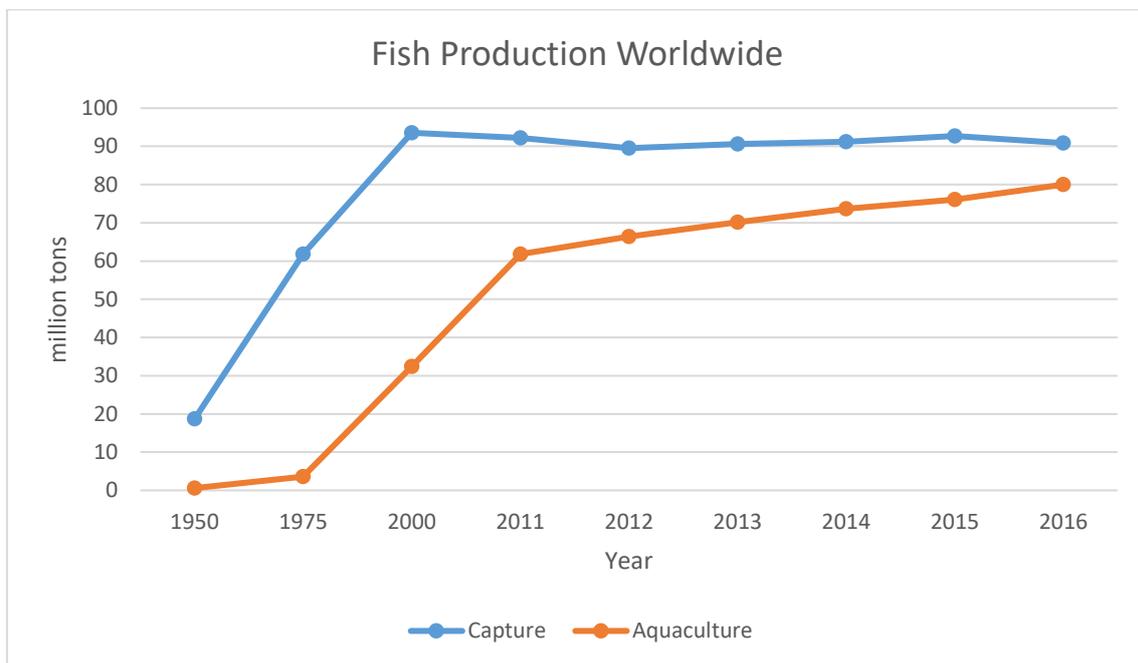


Figure 1. Fish production increased rapidly from 1950 to 2016 in terms of fish capturing and fish aquaculture, after [1].

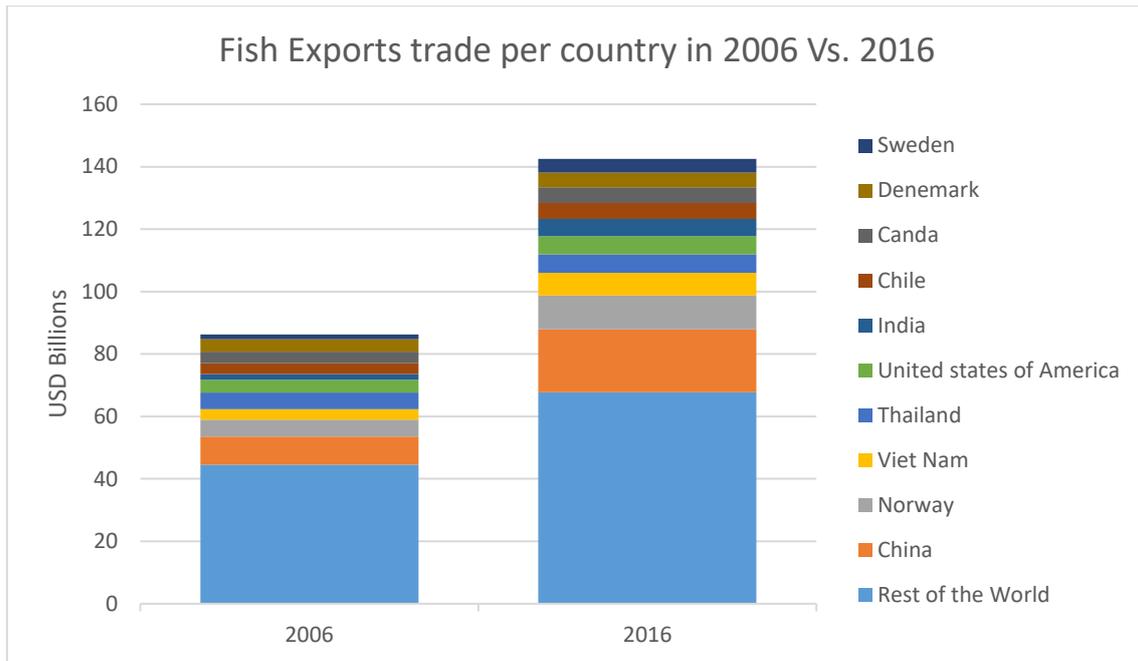


Figure 2. Fish World trade exports in USD billions with the top 10 countries showing a comparison between 2006 and 2016, indicating that the fisheries industry is growing very fast from 86 USD billions in 2006 to 143 USD billions in 2016, after [1].

Fish detection is essential for studies of types, population, behaviour, movement, migration and survival of hydraulic structures, most frequently dams and weirs. Underwater object detection is not limited to the fisheries industry, but are also needed for underwater pipeline inspections, marine environment protection and observation, diving assistance and the rapidly growing underwater robotics industry.

Underwater object detection faces many challenges, from limited visibility and colour reduction to non-uniform illumination resulting in objects appearing less shiny and not in their actual colour. Different sea states change the light propagation through the water surface, and various water types attenuate different light wavelengths, and many different species of fish can have very similar body shapes and colouring, increasing the difficulty of underwater classification.

A literature review on underwater object detection at early stages highlights that imagery was being processed based on contour, colour and whole shape matching that requires a particular setup, for example, the approach in [2] has a proper and uniform lighting and image quality, which is not suitable for real-life underwater object detection. Improvement in this field was achieved with the use of neural networks using customised

convolution neural networks to learn the object features and therefore identify this object type on a different image set.

Yet the proposed algorithms in [3], [4] use underwater images without applying digital image enhancement on dataset to improve the image quality degraded by underwater imagery problems, as they depend on dataset of images captured in uniform lighting conditions, so the used imagery to some extent should have a proper and uniform illumination to clearly show fish features, therefore trained model may not keep same performance in real-life situations such as noisy images, turbid water, partial images of a fish and overlapping fish bodies.

While digital image processing literature review for underwater imagery is rich with different approaches from image restoration algorithms that are aimed to restore the ground truth of the image, as proposed by authors in [5]. Some papers developed algorithms to restore images, and others used deep learning as introduced in [6]. Yet image restoration is dependent on many factors defining the characteristic of this water model at a specific depth and lightning conditions.

Researchers in [7] and [8] focused on image enhancement algorithms to improve the quality of the image regardless of the ground truth, aiming to reach a general principle to resolve the underwater effect on photography, so one given approach may be suitable for multiple water types and different depths.

The objective of this thesis is to develop an image processing workflow for underwater object detection by applying digital image processing on underwater imagery to improve image quality and use deep learning ability to work in real-time. As raw underwater images will result in lower performance than if it is processed with digital image processing algorithm, increasing the quality of the underwater image.

From the literature review, there was no approach found which has applied advanced digital image enhancement or restoration algorithm to improve underwater imagery *before* processing the imagery using deep learning. Generally, the author also noticed the lack of realistic underwater datasets to support the studies, and it was observed that most of the deep learning approaches used available datasets of high-quality images with nearly uniform illumination. Considering field studies with underwater imagery, such conditions

are less frequent, and therefore in this work, we have opted to use actual field imagery provided by commercial camera systems in freshwater and marine environments.

Regarding digital image enhancement, the author aims to resolve underwater imagery problems by eliminating non-uniform illumination, colour casting and reduced contrast and sharpness, which is partially addressed in existing literature, but not to the extent desired for real-world applications.

Therefore, this research focuses on digital image processing that resolves previously mentioned underwater imagery, to be used for real-time deep learning network training, aiming to improve performance of underwater object detection, compared to train without image enhancement

This thesis has the following two major objectives:

1. Provide an automated image processing workflow for underwater fish detection, which first enhances underwater images and then applies a deep neural network for detection of imagery with and without fish.
2. Provide a new annotated underwater image dataset taken from field videos captured by different cameras and for different types of fish, including augmentation.

## **2 Background**

This purpose of this chapter is to introduce background information regarding the difficulties faced when applying digital image enhancement in underwater imagery, and object detection using deep learning, in order to select a suitable real-time deep learning algorithm.

### **2.1 Image Processing**

Digital image processing is the modification of a digital image using a computer. One of the early usages of digital image processing was in the newspaper industry in the early 1920s, to transfer images between London and New York through a submarine cable. Further image processing advancements introduced in the 1960s, aimed for improving image quality, thus serving many applications like satellite imagery, medical imaging and character recognition. Typical image processing techniques are image enhancement, restoration, encoding and compression [9].

Image enhancement is the processing of an image, so the resulting image is more suitable than the original image for a specific application [10]. While image enhancement is mostly a subjective process, image restoration is an objective process aimed to recover or reconstruct a degraded image, by using prior knowledge of degradation phenomena a model of degradation can be constructed, and by applying the inverse process of that model on the degraded image, the original image can be recovered [11].

Here it is worth mentioning that the visual evaluation of image quality is highly subjective, therefore defining a standard for good image is elusive to achieve and by extension comparing algorithms performance. On the other hand, machine perception can provide a means for comparing algorithms performance by judging the improvement in the application performance. As an example, this thesis uses digital image enhancement to improve the ability of underwater object detection. Ideally, this will lead to improvements in underwater object detection performance [12].

### 2.1.1 Underwater Imagery

Underwater imaging is challenging due to water's physical interaction with light, including absorption, diffusion, reduced visibility, lowered contrast, non-uniform lighting, bright artefacts, noise, blurring, and diminished colour.

The underwater environment can introduce diffusion effects and non-uniform lighting. Specifically, light propagation underwater is creating light crinkle patterns, and surface irregularities caused by waves often diffuse light, as shown in Figure 3. The sun's position, time of day and seasons also affect the amount of light reflected, introducing poor visibility. Visibility is as well affected by particles and sediments in underwater environments. As illustrated in Figure 4, the penetrated light is vertically polarised, which can make objects less shiny and horizontal polarization is reflected, also showing that blue colour is Rayleigh scattered more than other colours [13].

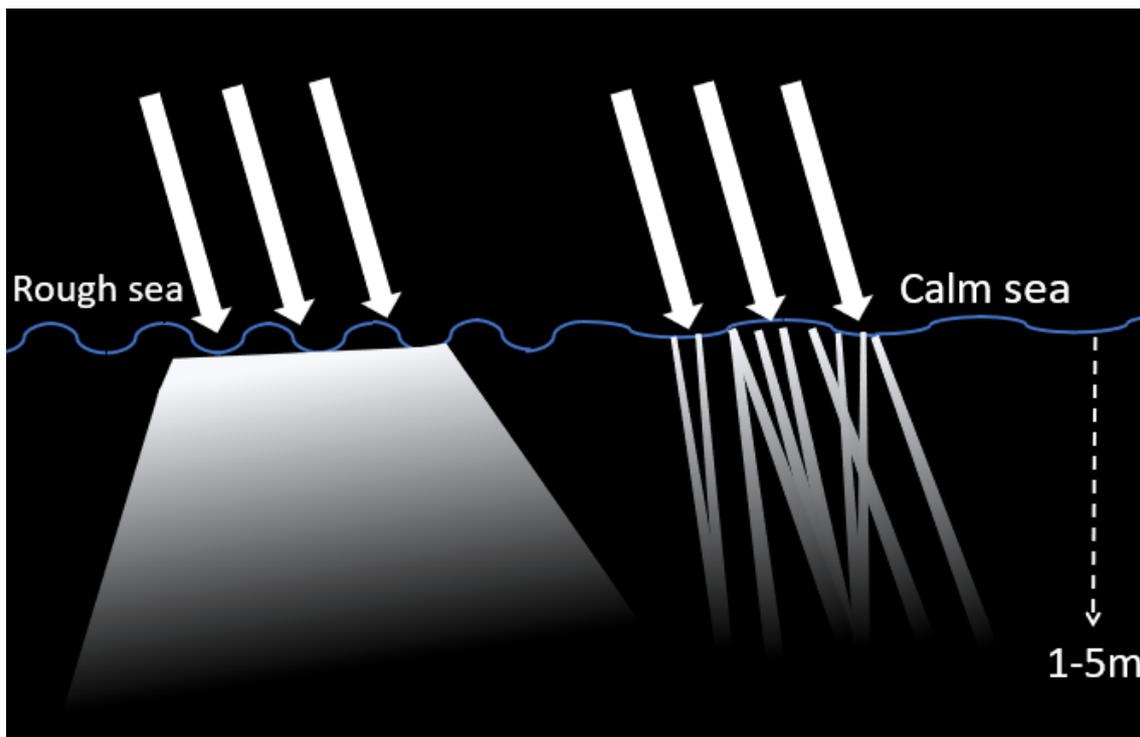


Figure 3. Water surface effects on light propagation as when the sea is calm, the light will create crinkles inside water, while if sea surface rough the light will diffuse, after [13].

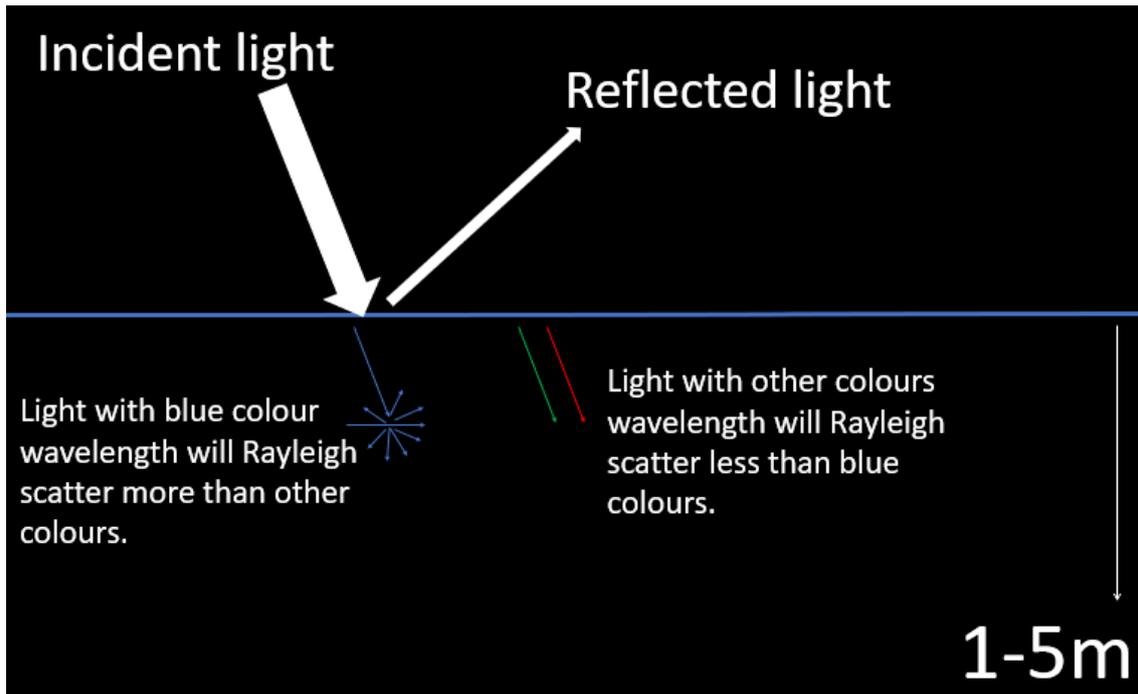


Figure 4. Underwater light propagation, as incident light hits the water surface, horizontally polarised light will be reflected to outside and vertically polarised light will penetrate underwater, and it is worth mentioning that blue light usually is the most Rayleigh scattered light, after [13].

As stated in [14], ocean water can be classified based on inherent or apparent optical properties. Inherent optical properties study the medium effect on a light beam where attenuation is caused by light scattering and light absorption. Apparent optical properties also include the medium's effect on a light beam and the study of the geometric structure of illumination. Jerlov water types classification is the first quantitative classification scheme and is based on apparent optical properties.

Jerlov water types are sub-classified into coastal and open ocean water, which is further divided into groups, where coastal water types are assigned to groups 1-9 and open ocean water types cover groups II, III, IA, IB. An overview of a subset of the Jerlov water types as presented in Figure 5, showing wavelength-dependent light attenuation coefficients [6], [14] and [15].

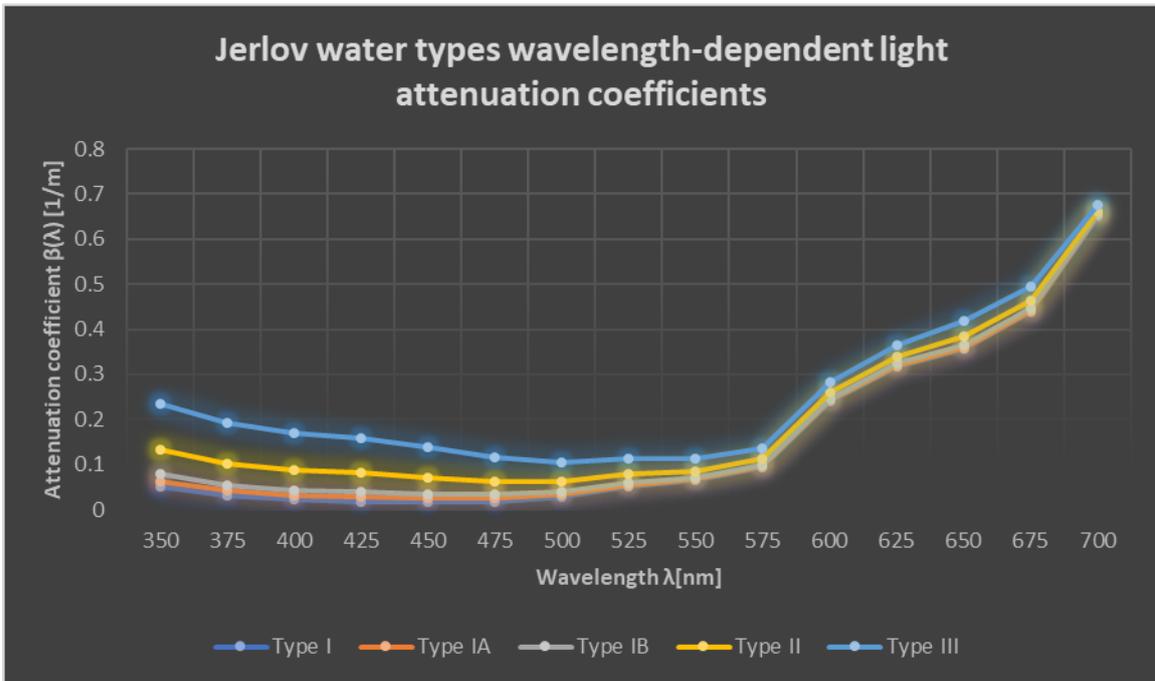


Figure 5. Wavelength-dependent attenuation coefficients of the Jerlov water types after [6], [14] and [15].

Visible light wavelengths of 600nm, 525nm and 475nm, often translate to red, green and blue respectively. As shown in Figure 5, red suffers from the highest attenuation. An example of multiple colours is shown in Figure 6 in open water, where the red colour is the first to diminish, and blue is last [6].

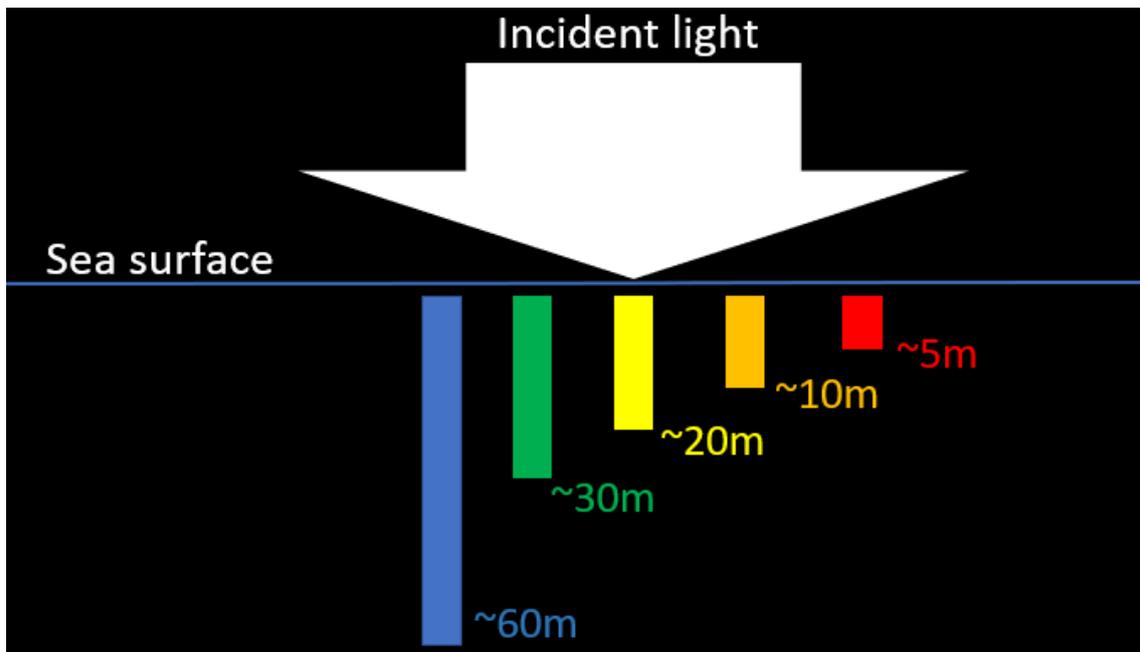


Figure 6. Illustration of underwater colour absorption, showing that red is the first to vanish at 5 meters whereas blue may persist up to 60 meters, after [5].

## 2.2 Object Detection Using Deep learning

Object detection is the ability to correctly recognise an object, determine its location and dimensions in the image, and perform semantic or instance segmentation [16]. Previous research was based on algorithms aimed at identifying an object based on its characteristics from shape, colour, contour matching, but this is not robust for real-life object detection. One such example is shown in 3.1.1.

The literature review provided in [14] shows that recent research using deep learning algorithms, powered by a convolutional neural network (CNN) can learn more features of an object, and divide deep learning into three stages regions proposal stage, feature extraction stage and finally, classification and localisation stage.

The deep learning framework can be categorised into region proposal based and object regression or classification based. Region proposal based will propose region of interest then try to classify objects in it, such as region-based CNN (RCNN), Fast RCNN, Faster RCNN and Mask RCNN, while regression or classification-based methods are aimed to adopt a unified framework to detect object directly such as single shot detector (SSD), and you only look once (YOLO) [17].

Previous works [16], [17] provide a full review of current deep learning algorithms, showing that you only look once (YOLO) provide one of the fastest solutions for object detection with lower false positive rate of background compared to other algorithms, also it is more versatile as it can be applied to detect objects on artistic work [16], [27].

Deep learning algorithm performance and speed are evaluated using the Microsoft common objects in context (COCO) database and the Pascal visual object classes (VOC) databases. Models trained on VOC database as in Figure 7 confirm that YOLO V3 offers acceptable mAP with consideration to the processed FPS. Since the author is aiming for real-time object detection, the YOLOv3 framework was chosen.

Models trained on the COCO database as in Figure 8 shows that mask region-based convolutional neural network using next generation of residual network (MASK RCNN + ResNext101) has the best mean average precision (mAP) as it was built aiming for best performance, not for fast detection, on the other hand, you only look once version 3 (YOLOv3) offers high mAP with higher FPS.



The deep learning algorithm you only look once (YOLO) version three was proposed in [19], dividing the image into smaller grids, and then applies a modified version of Darknet-53 neural network for feature extraction from each grid. YOLO has predefined three anchor boxes, that will be used by each grid to propose three bounding boxes that may have an object with the condition that this object centre resides in this grid centre as presented in Figure 9. This process is then repeated two more times, dividing the image using bigger scales, so in total, the image is divided into three grid-scales. In the end, many bounding boxes using non-maximal suppression will select the winning bounding box that contains the object with the condition that its confidence score is more than 25%.

The YOLO structure consists of 107 layers, as configured in YOLO configuration file, in details these layers are 75 convolutional layers for feature extraction supporting the three dividing grid scales, each layer consists of activation layer with activation function of leaky rectified linear unit (RELU) with predefined stride, padding, kernel size and filters and batch normalisation layer [19].

Twenty-three shortcut layers (known by skip layers), together with the convolutional layers, fulfil the concept of learning from residuals. Shortcut layers are configured with a linear activation function. Then four Route layer which can operate like a shortcut layer or to concatenate different layer, and two upsampling layers configured with stride equal to two, aimed to provide the grid-scale increasing, and finally, three YOLO layers used to extract the three predicted bounding boxes from each grid-scale, using the nine predefined anchor bounding boxes that are designed based on the trained dataset using k-means clustering as presented in Figure 9 [19].

An Example for YOLO structure that will be used in detection in this thesis is presented in Figure 9, also showing a chub fish detection on one grid at the centre of the object, this grid operation through the three grid scales, that will propose nine bounding boxes, then the highest confidence score will be selected, and all other bounding boxes with overlapping IOU will be discarded.

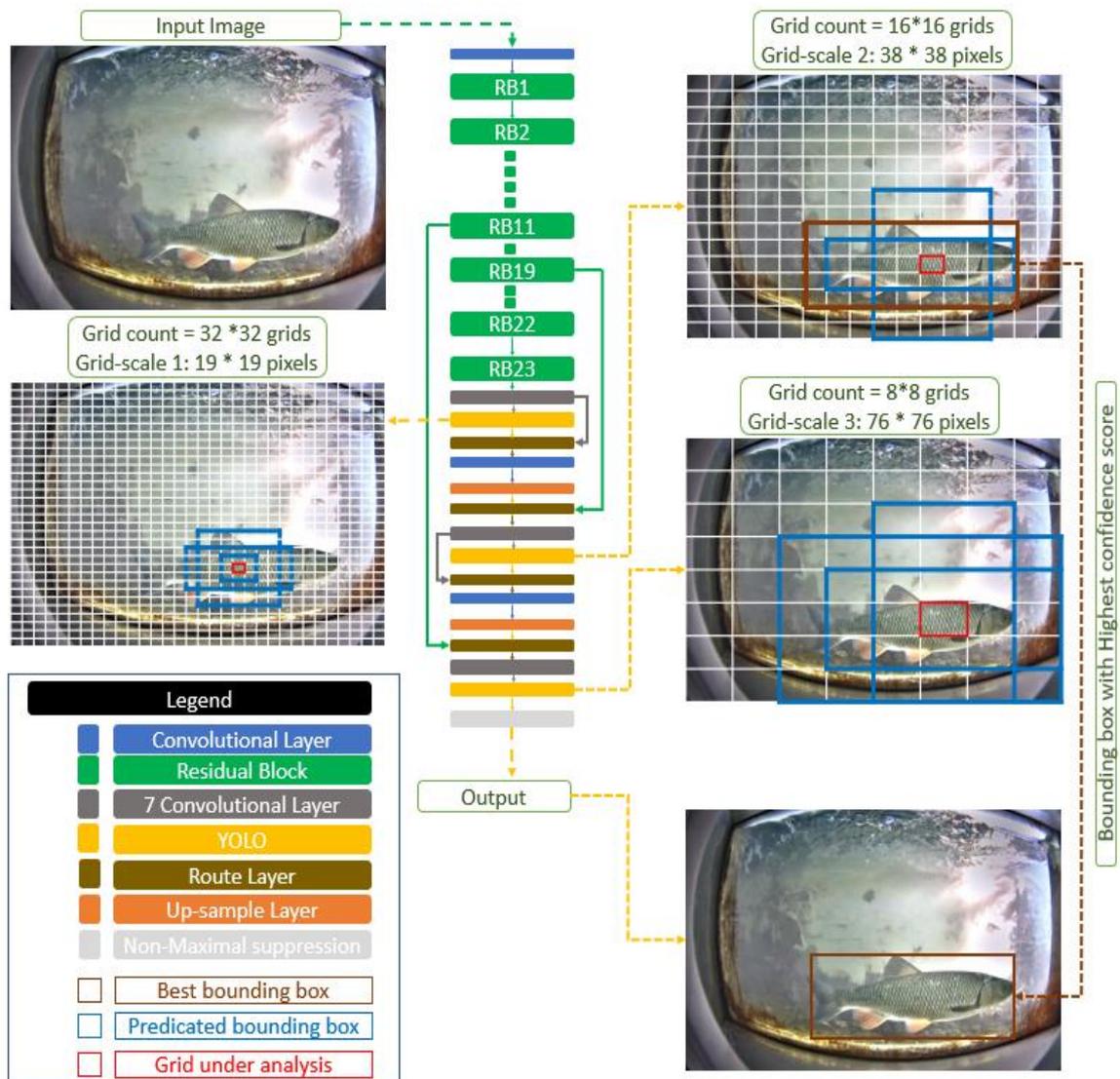


Figure 9. YOLO version three structure with an example on a chub fish developed based on the configuration file and after [19], [21].

## **3 State of the Art**

This chapter introduces the state of the art in underwater object detection, image enhancement and restoration. They are first treated separately before being included in a new image enhancement workflow to improve the dataset and then apply the output to train deep learning for underwater object detection.

### **3.1 Underwater Object Detection**

This subsection introduces state of the art for underwater object detection showing traditional ways for detecting objects based on object characteristics, and recent research of detecting objects using deep learnings.

#### **3.1.1 Underwater Object Detection Based on Object Characteristics**

Previous research using conventional methods to detect object underwater applied algorithms for contour recognition, shape, colour or a combination of them to detect if there is an object in the image and classify it.

The proposed approach taken in [2] for object detection first determined for every image if there is an object or not using colour segmentation, as the setup of fish imagery was in a manmade box with a known background colour. The background colour was blue, and the scene is nearly uniformly illuminated

Since the background colour is known apriori, an object can be detected in the foreground by subtracting the red channel from the blue channel as fish object pixels will have lower blue and higher red channels value thus a mask can be created resulting object pixels with value one and background pixels with value zero.

Afterwards, if an object is present, the method will extract its contour simply by using extracted mask information and identify it first as fish or not by performing data reduction to perform contour with fewer points, for example 40 points, and then by analysing the normalized length and turn angle between points fish object can be detected, if fish start tracking it and identify its species.

Species can be identified by checking the landmark points of the fish and turn angle distribution analysis (TADA), which can be assessed based on accuracy which is the

percentage ratio of correct prediction to the total predictions. TADA performance reached an accuracy of 73.3% for a dataset of 300 fish images for six species.

In general, this approach is well-developed to its time as it has its limitations due to the special setup of the environment with very good lighting to the fact that, it can detect object only when the whole fish is in the image without any fish bending, shadow and existence of other objects, so special setup will lead to good results, but in real-life, this method is not robust when the fish is partially occluded, which often leads to false object detection.

### **3.1.2 Underwater Object Detection Based on Deep Learning**

Deep learning greatly improved object detection. The proposed approach in [2] required a specific setup and not suitable for real-life use. Deep learning allowed computers to identify objects in more life examples through learning by itself the object from labelled dataset identifying the object in different positions, thus leading to real-life usage with better results.

The proposed approach in [4] used a deep learning algorithm for fish detection and classification. However, the approach was aimed to work on underwater imagery by first applying foreground extraction to improve the object detection, then enhanced images are then fed to a convolutional neural network (CNN) consisting of 2 convolutional layers where first layer kernel size is  $5*5*3$  and second layer kernel size is  $13*13$ .

The output of these two convolution layers is applied to feature pooling layer then to spatial pyramid pooling layer to help in recognising the object in different poses, then the classifier layer using support vector matrix (SVM) to identify the final classification of the object. The dataset used is from the fish for knowledge (F4K) dataset, which has 22,370 images of 23 fish species and all resolution resized to  $47*47$  pixels.

This approach uses a not very deep network compared to different algorithms reaching an accuracy of 98.57%, as accuracy represents the percentage ratio of correct prediction to the total predications.

But disadvantages of this research are that the foreground extraction has its limitation in real-life usage as different non-still non-fish objects and dataset has different image resolution from  $20*20$  to  $200*200$  pixels which they recovered by unifying the resolution

to 47\*47 which if increased will need more deeper network thus increasing the overall processing time. Adding to this the use of unequal image distribution between species with highest species has 12,112 images, and the lowest species has 25 images, and that the dataset is consisting of images with good quality and illumination and does not have image enhancement designed for underwater, thus may not be robust for real-life object detection for underwater images with low quality.

The proposed approach in [3] used images and a deep learning object detection algorithm, Fast R-CNN to classify the underwater images. The dataset created from fish for knowledge (F4K) dataset video repository have 12 fish species with a better balancing of images quantity compared to [4] and trained it using stochastic gradient descent (SGD).

Experimented dataset generated with different algorithms from R-CNN, Fast R-CNN and Fast R-CNN using singular value decomposition (SVD), showing that processing time for one image 24.945, 0.311 and 0.273 seconds respectively with a mean average precision (mAP) of 81.2%, 81.4% and 78.9% respectively.

The disadvantage of this approach is that the dataset was formed using selected images with good lighting and pose of fish and no digital image processing is used which is expected to have lower performance in case of real-life underwater imagery.

## **3.2 Underwater Image Restoration and Enhancement**

A great deal of researching has been developed for underwater digital image processing using both image restoration and image enhancement, in this subchapter will present a sample of these research in three categories of image restoration using traditional algorithms, image restoration using deep learning trained on synthetic images and image enhancement.

### **3.2.1 Image Restoration**

The proposed approach in [5] showed that the underwater environment has dominant, intermediate and inferior colours, using an interesting approach for histogram stretching each channel with different thresholds, then apply histogram specification with respect to Rayleigh distribution, then apply contrast correction in HSV colour space then compose channels to get the output image as shown in Figure 10.

Histogram stretching is colour based, for example, if 256 histogram bins are used from 0 to 255, then dominant colour which is the blue channel will be stretched from 0 to the minimum between 242 and the blue channel maximum used histogram bin. The inferior colour, which is the red channel, will be stretched from the maximum between 13 and red channel minimum used histogram bin to 255. the intermediate colour, which is the green channel, will be stretched from 0 to 255. Colours can change based on water type.

Histogram specification with respect to Rayleigh distribution used to overcome the degradation effect by light propagation in water. Contrast correction and colour composition will be handled in HSV colour space by setting the stretching output value of S and V to be 1% of the lower and upper limits to avoid over and under-saturation.

Although this paper introduced a good approach and results which were verified by developing its approach using Python and used it on many images in the used dataset, still need more parameters to be adjusted, for example, some underwater images dominant colour is green, so by using the original algorithm settings in [5] the image restoration will fail. For successful restoration need to modify the algorithm to have the green as the dominant colour and modify the stretching thresholds of the red colour as presented in Table 1.

Also, the depth of capturing the image was a factor, so a wise selection of parameters values for stretching threshold, Rayleigh coefficient and colour channels definition for every image. In the end, this algorithm is serving specific water type, and up to specific depth, that is why in this thesis will not use image restoration algorithm.

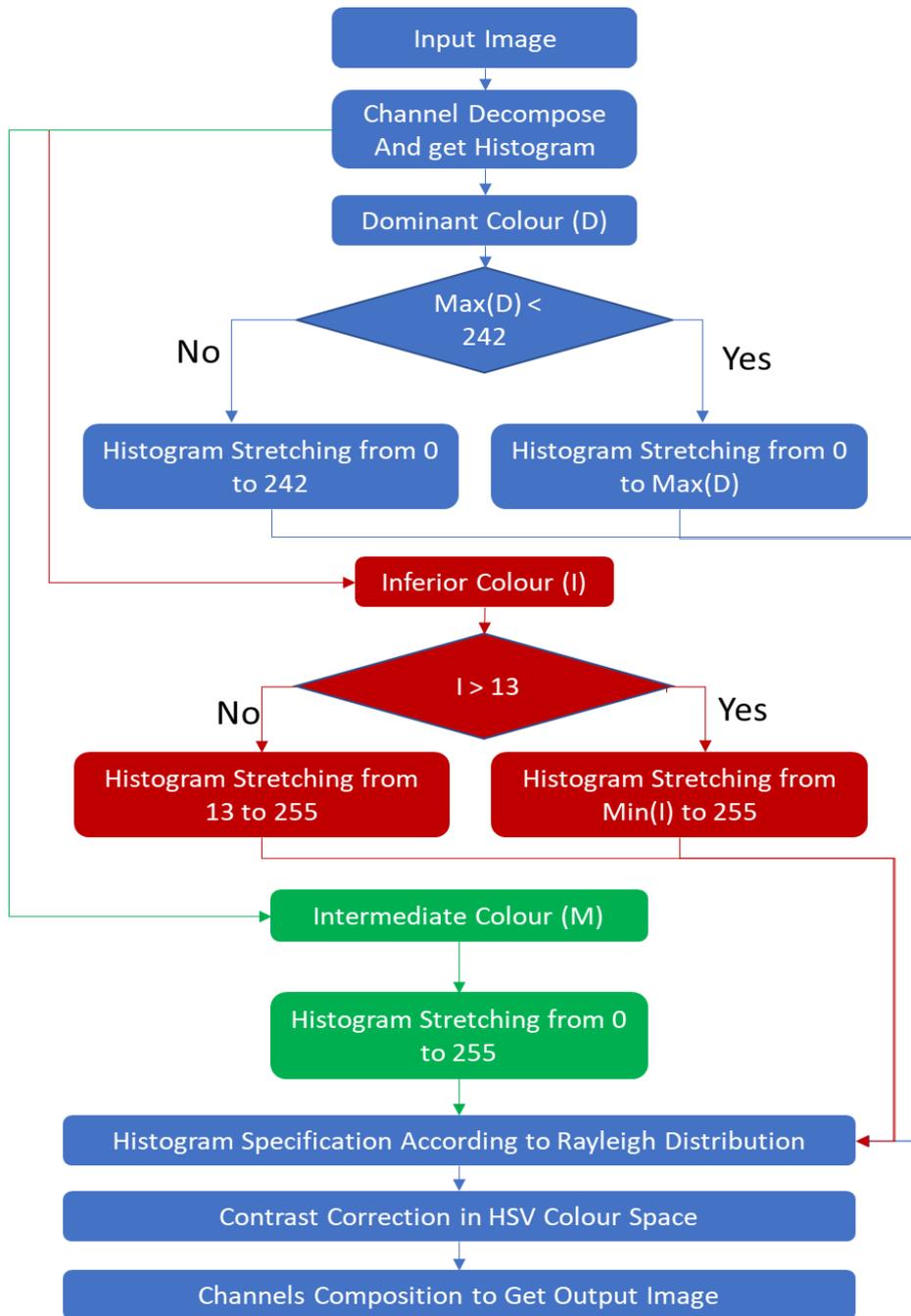
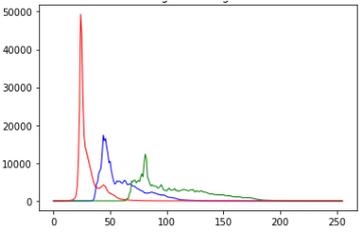
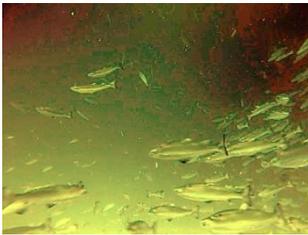
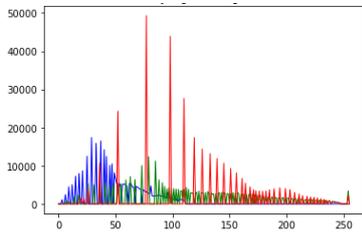
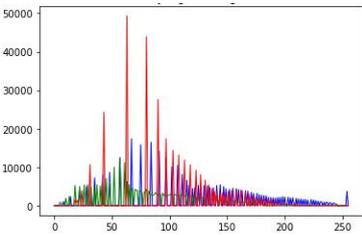


Figure 10. Block diagram in [5] for the proposed image restoration algorithm.

Table 1. An example of processing an input image using the algorithm in [5], where the input image dominant colour is green. Using original settings in this algorithm will provide a failed restored image. For successful restoration need to modify the algorithm to have the green as the dominant colour and modify the stretching thresholds of the red colour.

	Image	Histogram.
Input image		
The output image using the algorithm in [5], with the dominant colour is blue (original settings in the paper)		
The image processed with the algorithm in [5], with green as the dominant colour and modifications on the stretching thresholds of the red colour.		

### 3.2.2 Image Restoration by Synthesising and Deep Learning

Another example of image restoration is synthesising underwater effects on a non-underwater image dataset and feed it to the CNN algorithm to learn the inverse model so later can be used to perform image restoration. One advantage of such approach is that the CNN algorithm can be trained on various models hence can be trained for different water types, or even use a classifier to judge water type and automatically apply correct inverse modelling to restore an image.

The proposed approach in [6] named UWCNN, synthesis underwater images as different water types will introduce different attenuation model versus light wavelength. Thus, the underwater formation can be achieved as for specific wavelength. Underwater image synthesising is done by assuming distance  $d(x)$  range from 0.5m to 15m and global homogeneous background light  $0.8 < B_\lambda < 1$  and applying wavelength medium attenuation coefficient  $\beta_\lambda$  based on water type to be trained. Final dataset of images were 5000 images for training and 2495 images for validation at resolution 310 \* 230 pixels.

UWCNN underwater convolution neural network represented in Figure 11 was following Dense net structure without batch normalization for better memory efficiency, having 3 Blocks each block consist of 3 dense connected layers of 2D convolutional layers with filters 3\*3\*16 and activation layer using activation function RELU and ending with concatenation layer.

All blocks were densely connected where the convolutional neural network (CNN) was trained on the residuals using Adaptive momentum estimation (ADAM ) with originally proposed values as proposed by authors in [22] with learning rate 0.0002. A loss layer calculated the loss using MSE, mean square error and SSIM, Structural Similarity Index.

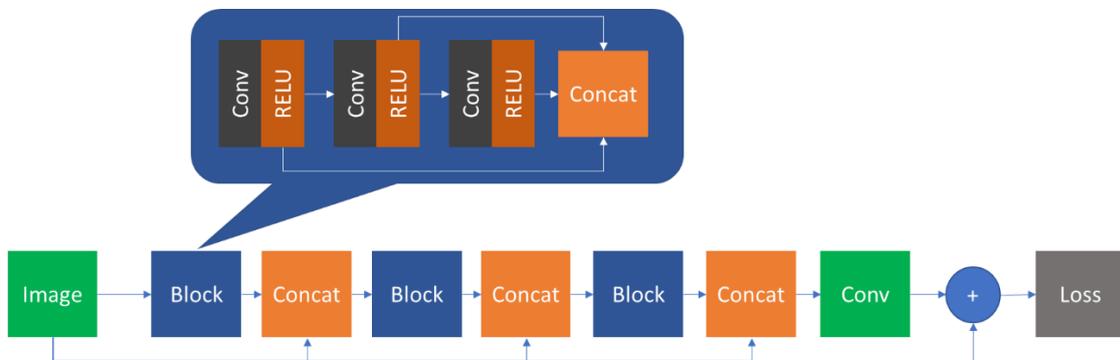


Figure 11.UWCNN model deep learning block diagram, after [6].

UWCNN was easier to train it on different water type and use it to restore underwater images but depending more on synthesising images does not reflect real-world underwater images hence output quality is less than normal image restoration images.

### 3.2.3 Image Enhancement

Unlike image restoration techniques, image enhancement aims to improve image quality regardless of the model of water type and depth.

The proposed approach in [8] applies an emphasis homomorphic filtering to correct non-uniform illumination and then to Histogram matching aimed to increase the influence of inferior colour channel output from this stage will be low contrast image that will be enhanced by enhanced dual image fusion.

Dual image fusion, input image will be used to produce two images by dividing the image histogram at midpoint and produced regions will stretch independently then both fed into two-dimension discrete wavelet transform (2D-DWT) for fusion and output of inverse 2D-DWT will have better global contrast and by using contrast limited adaptive histogram specification (CLAHS) will introduce better local contrast.

A good addition here is that the homomorphic filtering used a Butterworth high-pass filter to eliminate low-frequency component, as illumination typically variate slowly across the image while reflection can introduce abrupt change in illumination.

The input image will be transferred to the log domain for less computational complexity and, then transferred to the frequency domain to apply the Butterworth high-pass filtering, then a reverse FFT and log domain to obtain the output image.

Emphasis homomorphic filter is used to amplify the high-frequency components more than the low-frequency component, rather than completely delete the low-frequency component, which is a very promising idea. However, using a high-pass filter will lead to a reduction of the details and contrast of the image, as most likely, the very low frequencies have the highest magnitude.

The proposed approach in [7] presented in Figure 12, introduces input image to white balancing technique aimed to discard colour casting by modifying intensity, then two parallel paths will be used to improve the image, first is colour correction path and the second is for sharpening the image. Colour correction will use the white balanced image as its input. Sharpening path will use the white balanced image after removing noise using the bilateral filter.

Both paths will go through the same process, and each input will be passed to weighting process which includes Laplacian constant aimed for detecting global contrast, and local contrast is handled by comparing each pixel and its neighbourhoods average, and saliency is used to emphasize the objects that lost prominence, last weight is exposedness aimed to preserve the local contrast appearance.

The same time input of each path will be decomposed to a pyramid by Laplacian operator of different scales will provide patterns of details between these different scales, and at the same time we will apply a Gaussian pyramid to the summation of the weights using same levels as in Laplacian pyramid to the input image, then multiply level-wise the Laplacian of the input image to the gaussian of the weight, using this way halos creation can be avoided.

Python Code was developed according to this paper, with the support of MATLAB code developed in [23], and tested on our dataset with images with a variety of underwater imagery difficulties and showed a good performance but still need better handling for non-uniform illumination problems as discussed in 4. Adding to this that white balancing algorithm was not efficient for all cases as discussed in 4.2.

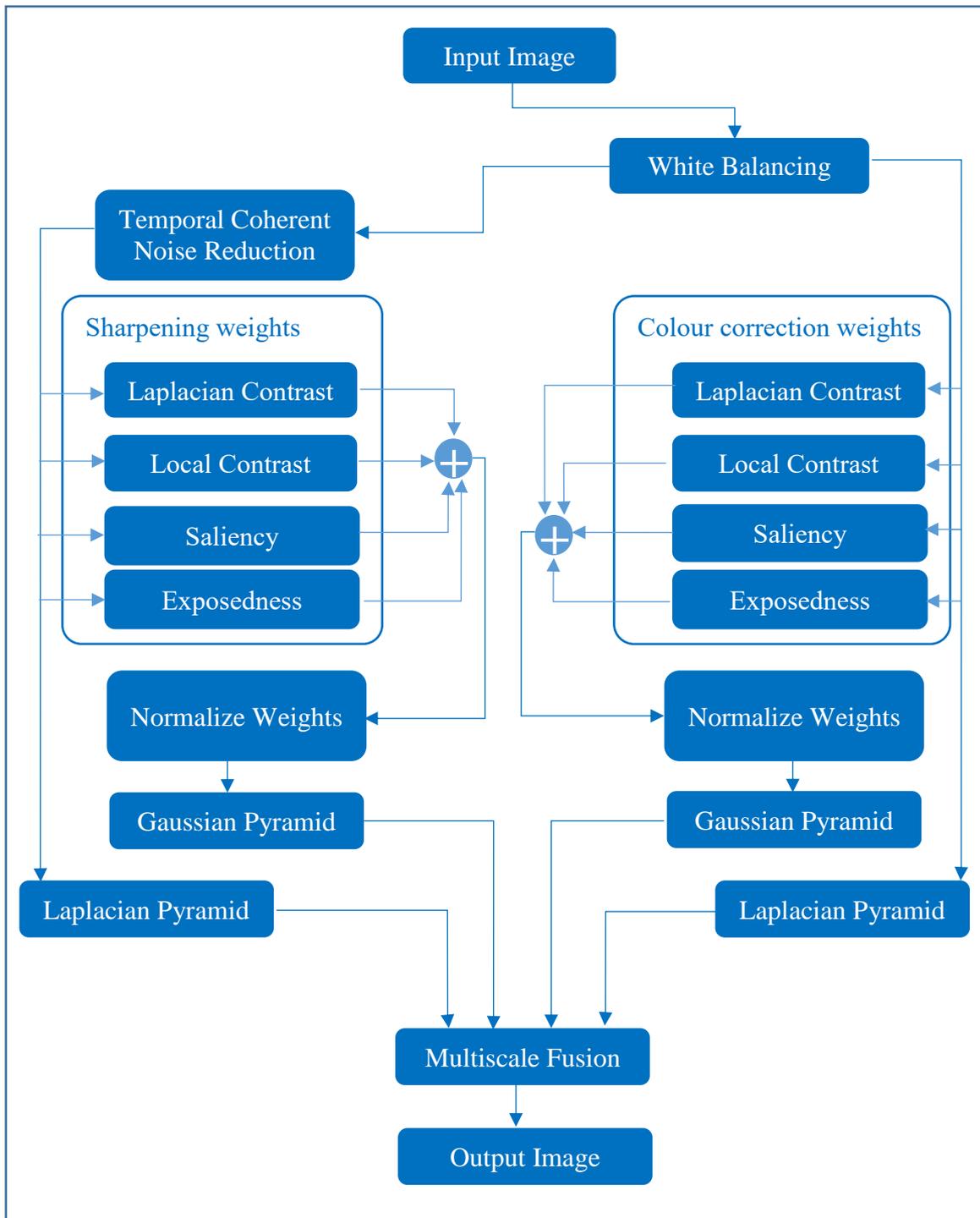


Figure 12. The proposed approach in [7] for image enhancement, as input image will be subjected to processing in two paths for colour correction and sharpening, then output image will be acquired by fusion.

### **3.2.4 Image Restoration Compared to Image Enhancement**

In general, image restoration algorithms aim to model the effect of water on the captured image which make model specific to a certain type of water and depth for example algorithm work on open ocean will not offer same quality improvement in coastal water rather than river water, adding to this at different depth.

On the other hand, deep learning image restoration algorithms is good solution for this as the algorithm can be trained for different water types and some papers offered solution for the algorithm to support multi-model image restoration at the same time, but it is still based on synthesized images of underwater effect which does not offer good quality as in image enhancement algorithms that is why image enhancement algorithm will be the aim in this thesis.

## 4 Digital Image Processing

Underwater imagery suffers from major problems resulting in that underwater images has low sharpness and details with colour castings and diminishing, thus in this chapter will propose an algorithm to improve underwater images as its block diagram presented in Figure 13.

Underwater, major problems are the non-uniform illumination, colour casting and diminishing resulting in that objects, in [7] image enhanced by addressing the colour casting, colour diminishing, which resulted in good results for images with decent lighting conditions at low depth in water (less than 9m) and, without non-uniform illumination that can be resulted from both sea structure and backscattering.

In this thesis, the author introduces the homomorphic filtering idea that authors used in [8] but, using bandpass filtering, reducing the contrast and image detail loss, compared to the high-pass filter. And implementation will be only on lightness channel of LAB colour space to decrease computational difficulties, then the output will be fed to the algorithm proposed by authors in [7], see Figure 13, thus improving non-uniform illumination of the input image.

To avoid overexposure or underexposure during white balancing, automatic clipping thresholds at two times of the channel standard deviation from the average value, thus improving exposure and colour dynamic range resulted from white balancing using simplest colour balance proposed by authors in [24].

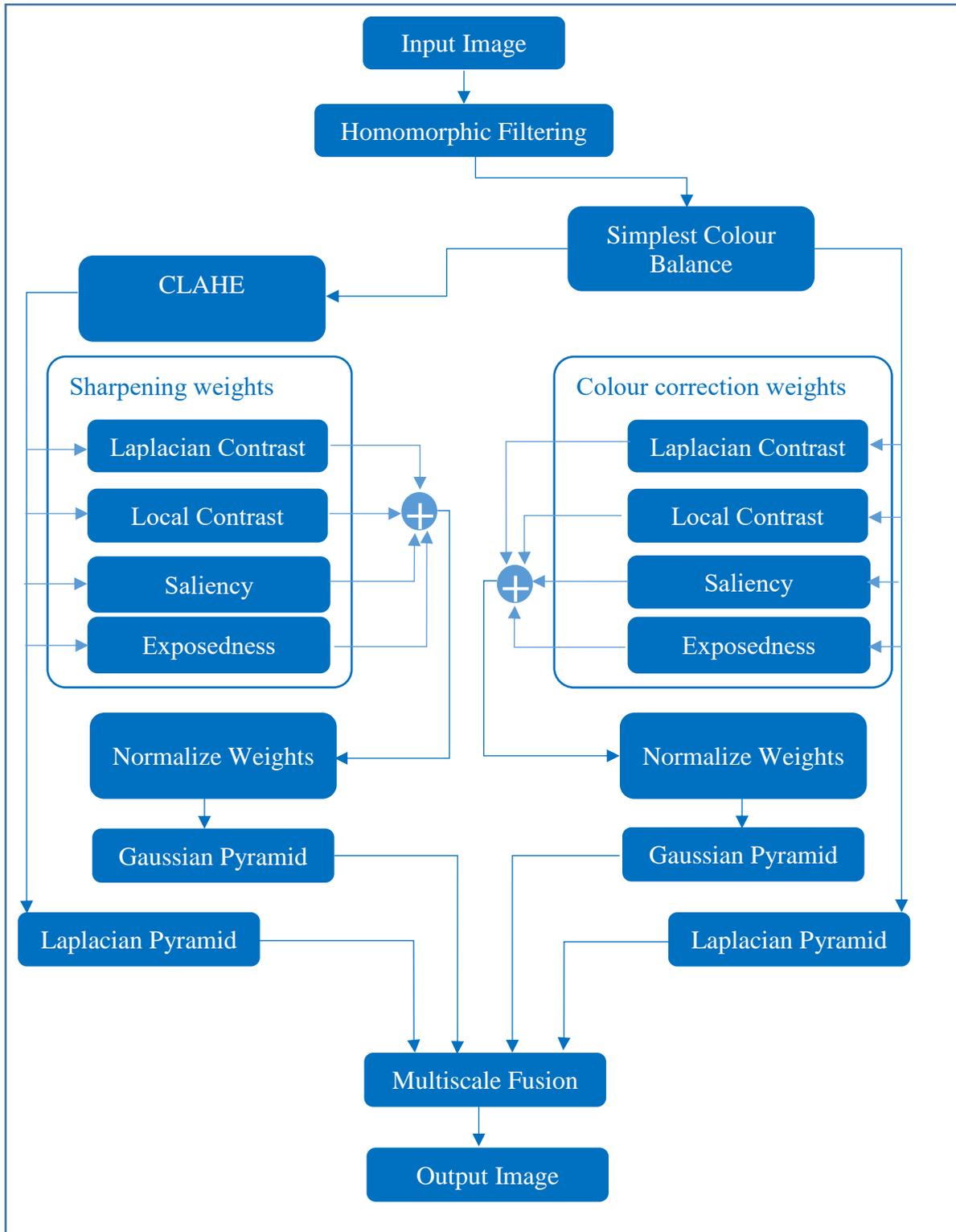


Figure 13. The proposed algorithm for underwater image enhancement, after [7] and [8], as input image will be subjected to processing in two paths for colour correction and sharpening, then output image will be acquired by fusion.

## 4.1 Emphasis Homomorphic Filtering

The purpose of this method is to eliminate non-uniform illumination resulted from underwater photography. Emphasis homomorphic filtering was first introduced in [8], with the objective to eliminate non-uniform illumination that can be caused by both sea structure and backscattering during photography.

Here it is assumed that illumination variance along image remains restricted within low frequencies, but using high-pass filter used by authors in [8] will also reduce contrast and image details within pixels that have very low frequency, even zero frequency which has no variance at all and most likely to have the highest magnitude. Therefore, using a high-pass filter in emphasis homomorphic filtering is unnecessary, reducing the magnitude of some frequencies that have no relation with illumination.

This thesis aims to use the bandpass filter to optimize the filtering by not reducing the components of very low frequency. Filtering is applied to the Light channel in LAB colour space instead of applying it to the three channels in RGB Colour space, as the purpose here is to address non-uniform illumination thus saving computational effort as well.

### 4.1.1 Image Illumination

Illumination model of an image can be presented as shown in Equation (1) that the image intensity  $I$  of pixel coordinate  $x, y$  can be represented as the multiplication of illumination  $L$  and reflection  $R$  of the same pixel coordinates.

$$I(x, y) = L(x, y).R(x, y) \quad (1)$$

Image illumination normally varies gradually across an image. Thus, in the frequency domain, it can be understood as a low-frequency component, while reflection from object surface can abruptly change at object location within image pixels thus can be understood to be a high-frequency component. By eliminating the low-frequency component, this will tend to eliminate illumination from the global scene and maintain an object's reflection, thus eliminating non-uniform illumination if it existed.

### 4.1.2 Frequency Domain of an Image

Images represented in the frequency domain describe the rate of change between pixels in the spatial domain, and such a representation can provide essential information about the image in terms of eliminating non-uniform illumination.

An image in the frequency domain has low frequencies at the centre of the image, and the frequency will increase as far as you are from the centre of the image. According to this high-pass filter will aim to eliminate the frequencies in the centre of the image and vice versa.

Figure 14 shows the frequency domain of light channel on image in LAB colour space, as in spatial domain image is with coordinates of  $x$  and  $y$  representing the pixels coordination according to the image width and height respectively, while in the frequency domain image it is represented by  $u$  and  $v$ , representing the frequencies of the image in the two dimensions.

Fourier transformation of an image in the spatial domain will acquire the image in the frequency domain, while inverse Fourier transformation will be used to return from the frequency domain to the spatial domain.

The important point here is that low frequencies usually have the highest magnitude (especially frequencies from zero to one), and as the frequency increase the magnitude is decreasing, in other words, most of the adjacent pixels have a small rate of changes between each other, and small number of the neighbouring pixels that have a higher rate of change between each other's

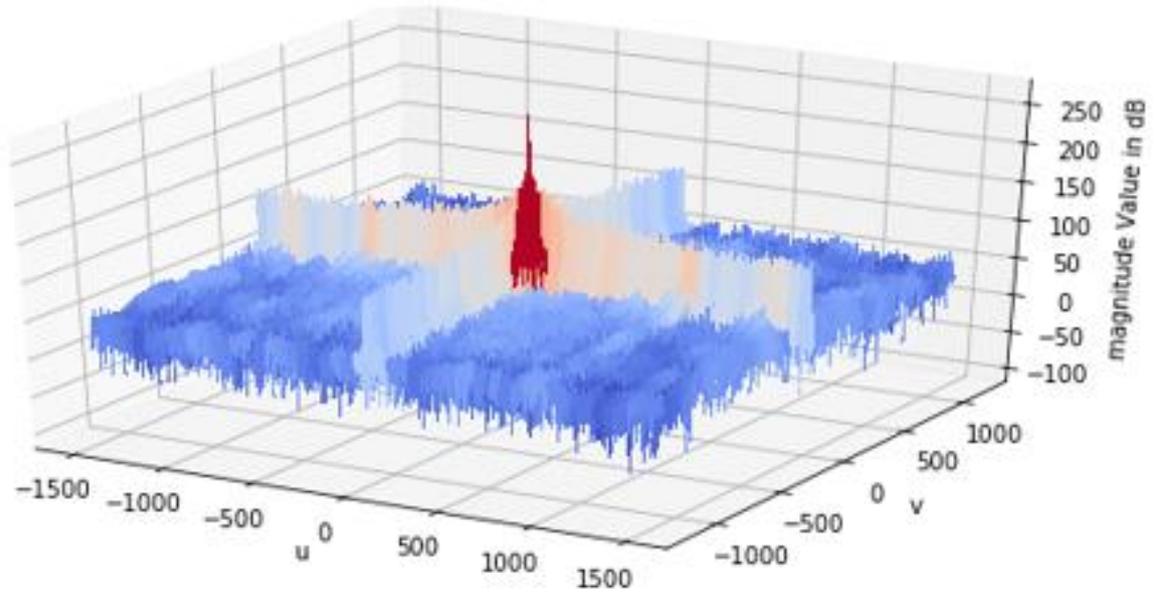


Figure 14. The image in the frequency domain where the low frequencies are in the centre of the image. Frequencies tend to increase as the distance from the centre increases. For every frequency, the magnitude in dB for the rate of change in the spatial domain.

### 4.1.3 Filtering Image in the Frequency Domain

Filtering of an image is used to adapt the image to a certain application. In general, a low-pass filter smooths the image, and a high-pass filter sharpens the edges and a bandpass filter selectively smooths and sharpens other pixels within the image.

#### 4.1.3.1 Filter Types and Emphasising a Filter

Since illumination variances reside in low frequencies, high-pass filters and bandpass filters are commonly used to eliminate non-uniform illumination. As an example, the filter's 3D representation in the frequency domain using a Butterworth bandpass filter is in Figure 15, Gaussian high-pass filters in Figure 16 and Butterworth high-pass filter is presented in Figure 17.

For constructing these filters, distance  $d(u, v)$  of certain frequency from the centre is calculated is shown in Equation (2).

$$d(u, v) = [((u - center_u)^2 + (v - center_v)^2)^{\frac{1}{2}}] \quad (2)$$

As  $u, v$  are the spatial frequencies coordinate in the frequency domain,  $center_u$  and  $center_v$  and the coordinate of the pixel at the centre in the frequency domain which is the pixels of zero frequency.

The transform function of a Gaussian lowpass filter is provided in Equation (3), which will be used to provide Gaussian high-pass filter as in Equation (4).

A Butterworth low-pass filter is presented in Equation (5) and will be used to provide Butterworth high-pass filter as in Equation (6), and finally, the summation of Butterworth filter and the high-pass filter at different cut-off frequencies will provide Butterworth bandpass filter as in Equation (7), with the condition that low-pass filter cut-off frequency  $\sigma_L$  is lower than the high-pass filter cut-off frequency  $\sigma_H$ .

$$H_{\text{Gaussian low pass filter}} = e^{-\frac{d(u,v)^2}{2*\sigma^2}} \quad (3)$$

$$H_{\text{Gaussian high pass filter}} = 1 - H_{\text{Gaussian low pass filter}} \quad (4)$$

$$H_{\text{Butterworth low pass filter}} = \frac{1}{1+\left(\frac{d(u,v)}{\sigma}\right)^{2n}} \quad (5)$$

$$H_{\text{Butterworth high pass filter}} = 1 - H_{\text{Butterworth low pass filter}} \quad (6)$$

$$H_{\text{Butterworth band pass filter}} = \left(1 - \frac{1}{1+\left(\frac{d(u,v)}{\sigma_H}\right)^{2n}}\right) + \frac{1}{1+\left(\frac{d(u,v)}{\sigma_L}\right)^{2n}} \quad (7)$$

Where  $\sigma$  is the cut-off frequency of the filter to be applied and in case of Butterworth the  $n$  is the filter order to identify the slope of the filter for the Butterworth bandpass filter  $\sigma_H$  is the higher cut-off the frequency and  $\sigma_L$  represent the lower cut-off frequency.

Emphasising a filter is a process where the desired frequencies are amplified, and unwanted frequencies will be kept the same or decrease its amplitude using Equation (8).

$$\text{Emphasis filter} = \alpha + \beta * \text{HPF} \quad (8)$$

A high-pass filter *HPF* output will be multiplied by scaling factor  $\beta$  that should be greater than one, and then added to biasing factor  $\alpha$  that should be less than one as in Figure 15 for Butterworth bandpass filter and Figure 16 for Gaussian high-pass filter and Figure 17 for Butterworth high-pass filter, but instead showing all the image frequencies will present the frequency range with different variations .

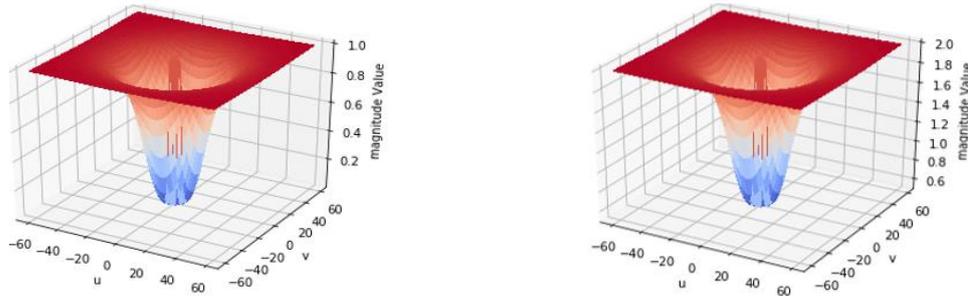


Figure 15. Butterworth bandpass filter where a higher cut-off the frequency  $\sigma_H = 20$ , and the lower cut-off frequency  $\sigma_L = 4$  is on the left, and on the right with the same settings is the emphasis Butterworth bandpass Filter where  $\alpha = 0.5$  and  $\beta = 1.5$ , showing frequency range with different amplification variation.

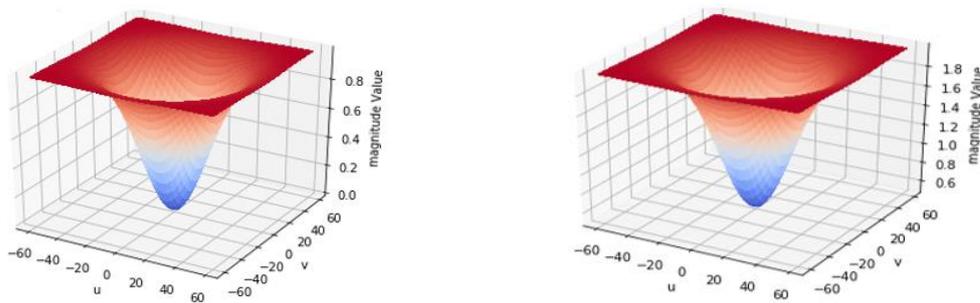


Figure 16. Gaussian high-pass filter where cut-off frequency = 20 is on the left, and on the right with the same settings is the emphasis Gaussian high-pass filter where  $\alpha = 0.5$  and  $\beta = 1.5$ , showing frequency range with different amplification variation.

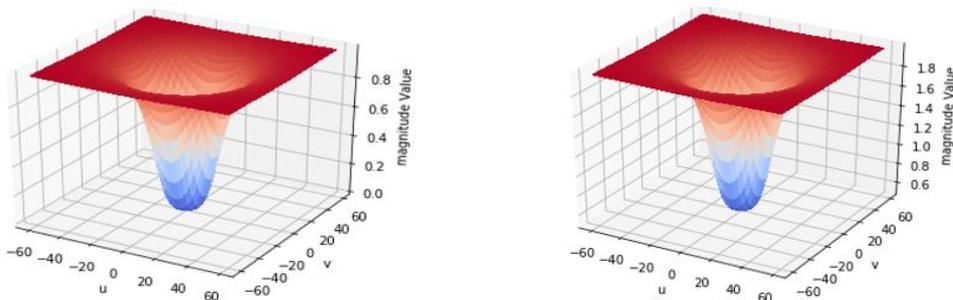


Figure 17. Butterworth high-pass filter where cut-off frequency = 20 and filter order 2 is on the left, and on the right with the same settings is the emphasis Butterworth high-pass filter where  $\alpha = 0.5$  and  $\beta = 1.5$ , showing frequency range with different amplification variation.

#### 4.1.3.2 High-pass Filter Compared to Bandpass Filter

Using a high-pass filter will eliminate the low frequencies which include the non-uniform illumination within image, but also will eliminate low frequencies that may not be related to non-uniform illumination like zero frequency which represent that there was no change between pixels and each other and worth mentioning that usually, this is the frequency

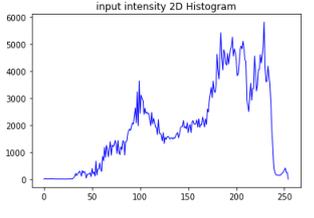
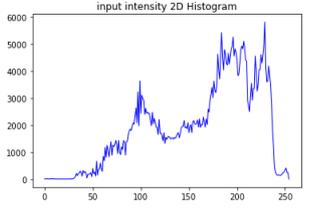
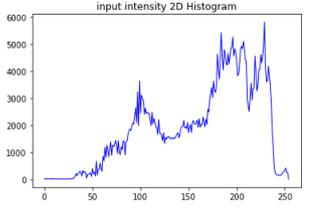
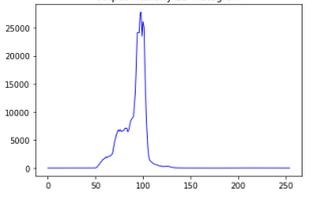
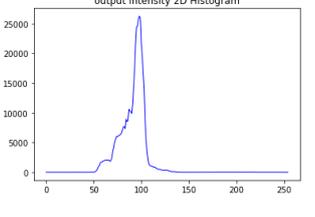
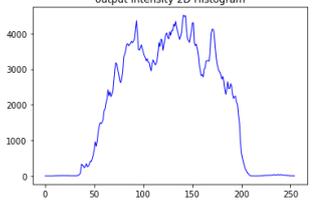
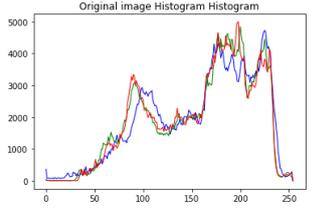
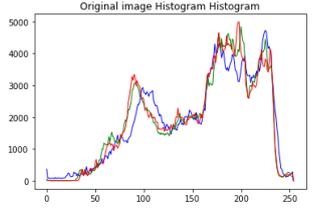
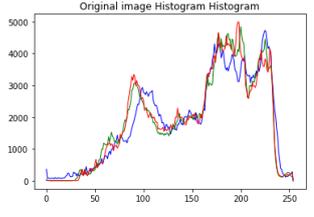
with the highest magnitude, resulting in compressing the dynamic change of the image so in comparison between input image and output image histogram, the output histogram will be in much lower dynamic range thus decreasing the image details and contrast.

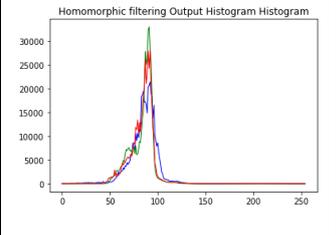
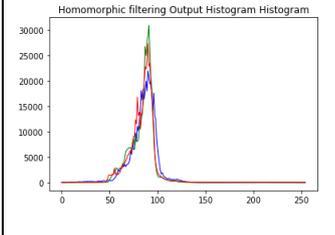
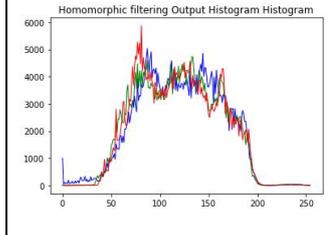
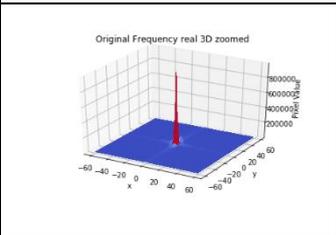
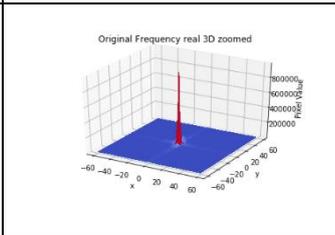
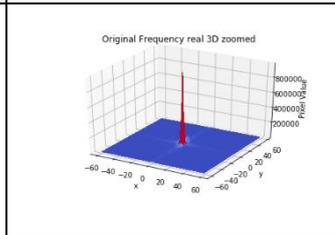
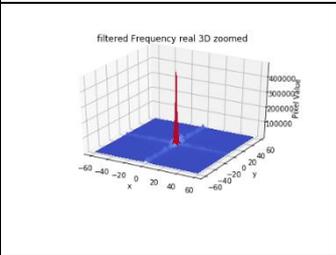
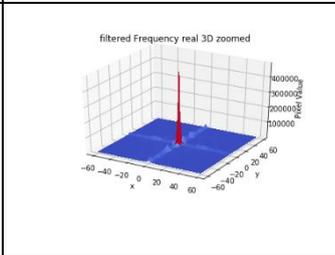
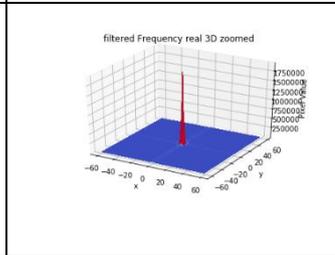
On the other hand, the bandpass filter can achieve the same result with less loss of image details and contrast, as it will selectively decrease frequencies that contribute to the non-uniform illumination, saving the very low frequencies and improving the output dynamic range and the image details.

An example of an input image being processed by different filters is presented in Table 2, and that these filters decreased the illumination factor and left only the light reflected from objects surfaces. The high-pass filters of Gaussian and Butterworth filters produce images with lower dynamic range, compared to Butterworth bandpass filter, proving that bandpass filter is a better approach for eliminating non-uniform illumination with fewer image details and contrast loss.

The bandpass filter output image shows that the general illumination in the image is unified, and even the shadow of the fish is not recognized as in the input image. Histogram of Light channel in LAB colour space shows that bandpass filter introduces better dynamic range compared to high-pass filter. Thus, the bandpass filter results in lower image detail loss.

Table 2. Applying an emphasis homomorphic filtering with different filters of Gaussian high-pass filters and Butterworth high-pass and bandpass filters, each column represents one filter type settings, input and output image with their histogram, Light channel in LAB colour space, frequency domain.

	Gaussian high-pass filter	Butterworth high-pass filter	Butterworth bandpass filter
Filtering settings	$\alpha = 0.5, \beta = 1.5$ and $\sigma = 20$	$\alpha = 0.5, \beta = 1.5$ and $\sigma = 20$	$n = 2, \alpha = 0.5, \beta = 1.5$ $\sigma_H = 20$ and $\sigma_L = 4$
Input image			
Output image			
Input image light channel histogram			
Output image light channel histogram			
Input image histogram			

	Gaussian high-pass filter	Butterworth high-pass filter	Butterworth bandpass filter
Output image histogram			
Input image the frequency domain			
Output image in the frequency domain			

#### 4.1.4 Proposed Filtering Method

As described in Equation (1), the image intensity is equal to the multiplication of illumination and reflection, so on order to decrease computational difficulty will convert image to colour space LAB and all operations will be performed on the lightness (L) channel where it will be converted to log domain as it will simple addition operation as in Equation (9) .

$$\ln(I(x, y)) = \ln(L(x, y)) + \ln(R(x, y)) \quad (9)$$

Adding padding to an image in the log domain is needed to reduce the effect of zero leakage from using the inverse fast Fourier transform. Afterwards, it is necessary to convert the log domain image back to the frequency domain using a 2D fast Fourier transformation (FFT). Next, the chosen filter is applied by multiplying the filter transfer function by the image in the frequency domain, producing the filtered image in the frequency domain, as presented in Equation (10).

$$filtered_{image} = FFT_{image} * H_{filter} \quad (10)$$

Finally, the filtered image is then transformed into the log domain and then to the spatial domain to get the final output as presented in the block diagram in Figure 18.

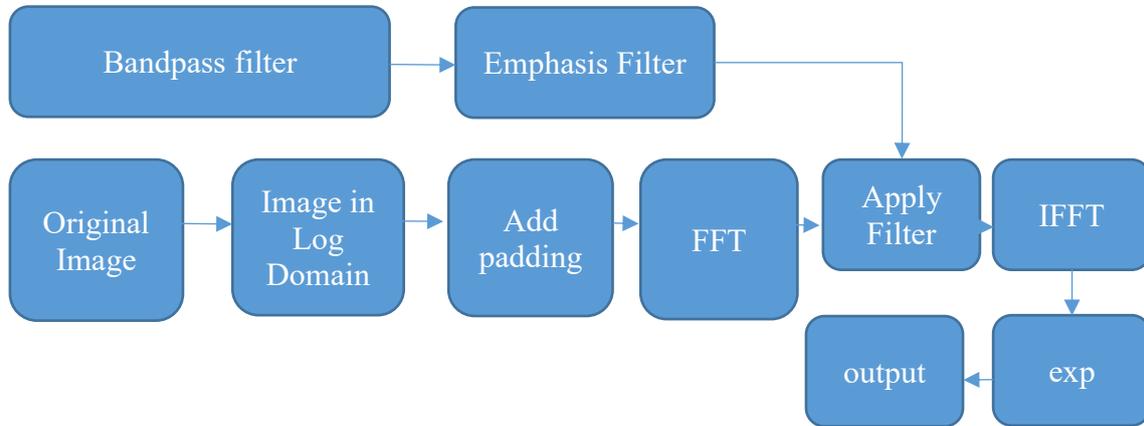


Figure 18. Emphasis homomorphic filtering proposed algorithm block diagram.

#### 4.1.5 Emphasis Homomorphic Filtering Results

Experimenting emphasis homomorphic bandpass filtering on images found that best settings are biasing factor  $\alpha = 0.5$  and scaling factor  $\beta = 1.5$  and lower cut-off frequency  $\sigma_L = 4$  and higher cut-off frequency  $\sigma_H = 20$  and filter order at low frequency to  $n = 6$  providing higher steep of filtering slope and at high-frequency  $n = 2$ , as shown in Table 3, the non-uniform illumination and shadowing are decreased.

Table 3. Results of emphasis homomorphic filter using the bandpass Butterworth filter to eliminate non-uniform illumination.

Input image	Output image
 <p>Wuppertalband Auer Mühle Drumm 2016-05-07 13:33:12</p>	 <p>Wuppertalband Auer Mühle Drumm 2016-05-07 13:33:12</p>
 <p>Wuppertalband Auer Mühle Drumm 2017-11-12 18:24:52</p>	 <p>Wuppertalband Auer Mühle Drumm 2017-11-12 18:24:52</p>
 <p>Wuppertalband Auer Mühle Drumm 2016-05-09 17:15:09</p>	 <p>Wuppertalband Auer Mühle Drumm 2016-05-09 17:15:09</p>
 <p>2017-06-10 14:44:29</p>	 <p>2017-06-10 14:44:29</p>

## 4.2 White Balancing

Using the proposed simplest colour balance algorithm based on sorting by the authors in [24], and which was used by authors in MATLAB Code [23], that was used to develop the proposed algorithm by the authors in [7] instead of their original white balancing technique that increased the average illumination value estimated with a percentage  $\lambda$ .

The proposed algorithm in [24] aimed to perform white balancing and contrast enhancement, by stretching the channels in RGB colour space to occupy the maximum possible range, and provided a solution for aberrant pixels by clipping a small range on the lowest and highest pixels values, thus properly stretching the dynamic range.

But proposed algorithm in [24] has no specific value of that small range to be clipped, and MATLAB code by authors in [23] provided to make it a hardcoded percentage that will be divided equally between high and low values. The hardcoded percentage used was 5%.

The author experimented the 5% hardcoded threshold on many images showed that not all the time, the full dynamic range is used for each channel. Also using a hardcoded value for all channels leads sometimes to over or under clipping for one of the channels resulting in overstretching of this colour and overexposure or underexposure.

### 4.2.1 White Balancing Proposed Method

The challenge here is to provide clipping thresholds that are optimized for every channel to avoid over or under clipping, as for example, some images have channels that have a high steep increase on one of the edges as the input image showed in Table 1, so the usage of hardcoded percentage will lead to over clipping this edge which affects the exposure of this channel.

In this thesis, the author proposes to automate the selection of the clipping thresholds, for every channel will acquire the channel average using Equation (11). Afterwards, the channel standard deviation using Equation (12) is applied using the thresholds to be the average plus two times the standard deviation using Equation (14) and the average minus two times the standard deviation using Equation (13). Using this process ensures that 95.4% of the channel's dynamic range will be used for histogram normalization later.

Averaging the channel, where X is the image width and Y is the image height, and V is the pixels value at location x, y on image width and height respectively as in Equation (11).

$$\mu = \frac{\sum_{x=1}^X \sum_{y=1}^Y V(x,y)}{X.Y} \quad (11)$$

Standard deviation can be calculated using Equation (12).

$$\sigma = \sqrt{\frac{\sum_{x=1}^X \sum_{y=1}^Y (V(x,y) - \mu)^2}{X.Y - 1}} \quad (12)$$

Then the lower clipping threshold is calculated with Equation (13), and the higher clipping threshold is calculated with the Equation (14).

$$C_l = 2\sigma - \mu \quad (13)$$

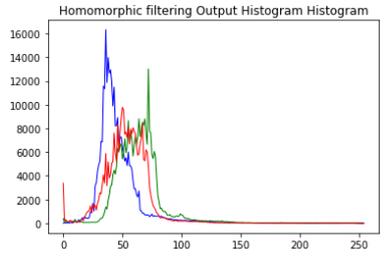
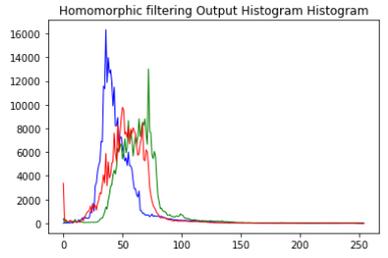
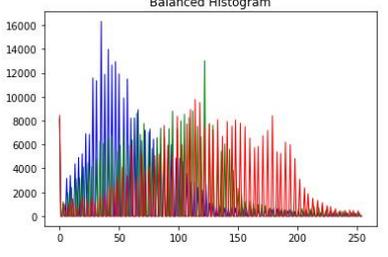
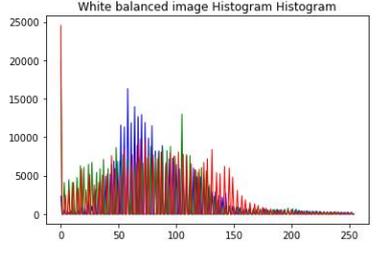
$$C_H = 2\sigma + \mu \quad (14)$$

In this work, twice the standard deviation was applied, as it will cover 95.4% of the channel image values' distribution.

#### 4.2.2 White Balancing Results

The comparison between simplest colour balance and the new white balance is presented in Table 4, where the simplest colour balance depending on hardcoded percentage lead to over clipping of the red channel and thus overstretching. This causes the image to appear redder than the new white balance method which dynamically assigned the clipping thresholds to result in an image with better colour representation.

Table 4. White balance using traditional simplest colour balance and proposed new white balanced, showing that the simplest colour balance output appears redder than the new white balance. As the new white balance algorithm is more optimized per channel and for every image.

	Simplest colour balance	New white balance
Input image		
Output image		
Input image histogram		
Output image histogram		

### 4.3 Temporal Coherent Noise Reduction

Aiming to reduce noise and furthermore improving the illumination conditions will apply on light channel in LAB colour space only the contrast limited adaptive histogram equalization (CLAHE) as proposed in [23] unlike Bilateral filter that is proposed in [7], where CLAHE is aiming to divide the image into smaller areas (8\*8 pixels), and then histogram equalization is performed on them, and to avoid [7], where CLAHE is aiming to divide the image into smaller areas ( 8\*8 pixels) and to avoid noise amplification a contrast limit is applied so if any pins are over 40 pixels will be clipped and distributed uniformly to other bins and then histogram equalization is performed on them as showing the results in Table 5.

Table 5. Temporal coherent reduction output using contrast limited adaptive histogram equalization (CLAHE).



## 4.4 Weights

All the following weights are based on the algorithm proposed in [7] to improve image quality, by obtaining the white-balanced image and getting the weights for improving colour correction and on parallel path get the temporal coherent noise-reduced image and calculating the same weights to enhance image sharpness.

### 4.4.1 Laplacian Contrast Weight

Aiming to enhance global contrast by acquiring the second derivative of the light channel in the LAB Colour space, the second derivative result of the zero-crossing is taken, and then this is an edge pixel as shown in Table 6.

### 4.4.2 Local Contrast Weight

With the objective to enhance local contrast by checking the light channel in LAB colour space, the relation of every pixel with its neighbouring pixels, where the resulted weight is dependent on the pixel luminance level and the local average of the neighbours of this pixel, as shown in Table 6.

### 4.4.3 Saliency Weight

In order to emphasize objects, this weight is taken in LAB colour space where the resulting weight  $S$  is the summation of the squared subtraction between the pixel value ( $l$ ,  $a$ ,  $b$  for every channel respectively) and the mean ( $l_m$ ,  $a_m$ ,  $b_m$  for every channel respectively) of this channel as in Equation (15). The results are summarized in Table 6.

$$S = (l - l_m)^2 + (a - a_m)^2 + (b - b_m)^2 \quad (15)$$

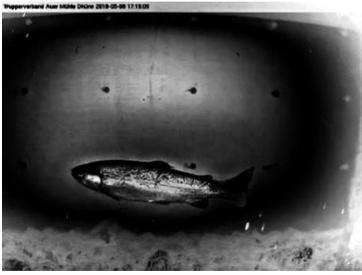
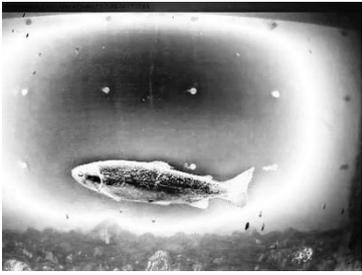
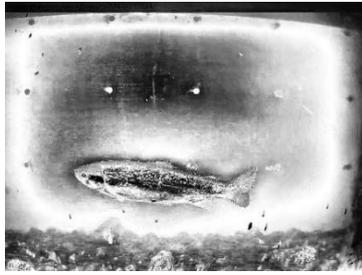
### 4.4.4 Exposedness Weight

Affects the extent to which a pixel should be exposed by operating on the light channel in the LAB colour space as in Table 6.

### 4.4.5 Weights Results

The result of the previously mentioned weight operations is presented in Table 6. Showing the results of each weight according to the two paths, aiming for colour correction and image sharpening.

Table 6. Weights results for every path, for colour correction path and sharpening path, will calculate the weights for Laplacian contrast, local contrast, saliency and exposedness weights.

	Colour correction path	Sharpening path
Input image		
Laplacian contrast weight		
Local contrast weight		
Saliency weight		
Exposedness weight		

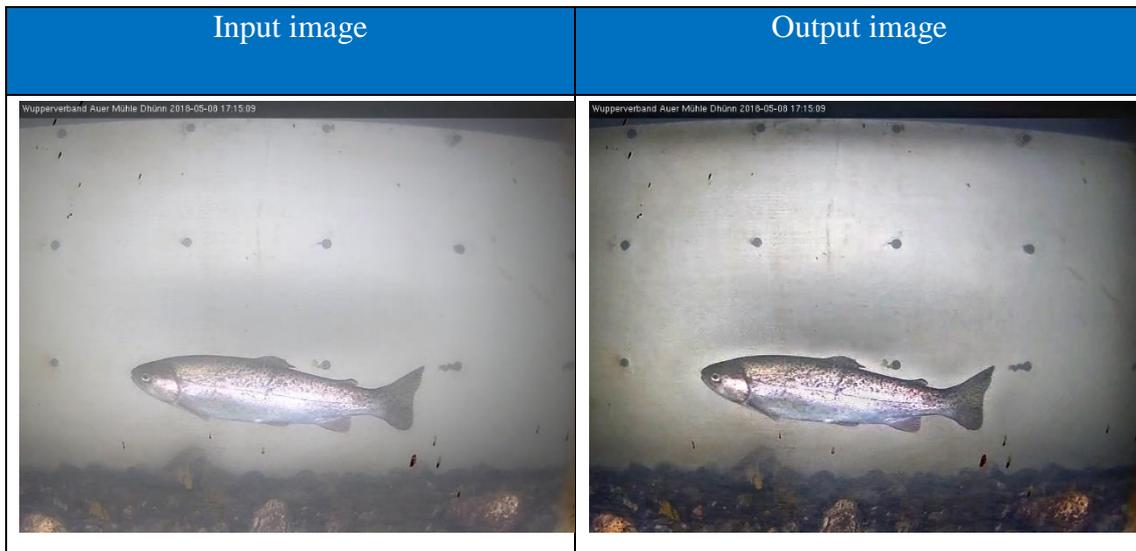
## 4.5 Multicast Fusion

Using the proposed method in [7] implemented the multicast fusion by using multicast Laplacian pyramid decomposition, where output weights result will be subjected to Gaussian pyramid to apply low-pass filtering on them at different scales.

Then input image from every path will be represented by set of images at different scales then subtract each two adjacent scales from each other thus resulting representation is the Laplacian pyramid of the input images, and for each scale will multiply for every scale the Gaussian pyramid low-pass filtered weight by the Laplacian pyramid input image then apply pyramid reconstruction by summing all the scales contribution to each other.

Implementing fusion at different scales will decrease the potential artefacts of sharp transitions, such as edges. The multicast fusion output thus represents the final output of the image, as shown in Table 7. The algorithm was developed in python.

Table 7. Digital image enhancement proposed results showing input image and the final output image.



## 4.6 Image Enhancements Results

The proposed underwater image enhancement algorithm is tested using a dataset of underwater images, and an example is shown in Table 8, the code for the algorithm is in [25]. The output is excellent and solved underwater difficulties earlier presented, which is promising to improve the deep learning training and object detection.

Table 8. Results of image enhancement using the proposed algorithm and algorithms in [5], [7], showing input and output image for each algorithm.

Input image	The proposed algorithm output image
	
Based on the algorithm in [5]	Based on the algorithm in [7]
	

The author has developed a python code for algorithms in [5], [7] that uses image restoration and image enhancement using fusion techniques, respectively [25]. That python code has been processed through the image dataset, and as shown in Table 8, that the proposed underwater image enhancement outperforms the algorithms proposed in [5], [7], in terms of better illumination distribution and colour correction. More examples are presented in Appendix 2 – Underwater Image Enhancement Results.

## 5 Underwater Object Detection

Two datasets are generated, the first is the raw collected fish image dataset, and the second generated by applying the underwater image enhancement algorithm proposed in the previous chapter on the first dataset. In this chapter will train the chosen deep learning network, “you only look once (YOLO)” to learn on both datasets and present the results, aiming to prove that fish object detection metrics will improve by using image enhancement.

### 5.1 Deep Learning Network Training Settings

As in the background chapter 2.2, it was showed that YOLO is one of the good approaches for object detection, as it provides a good mean average precision with high FPS allowing real-time object detection, thus YOLO version three windows edition developed in [26] will be used as the deep learning network.

YOLO training settings are presented in Table 9. Settings include hardware used, training resolution, parameters to avoid overfitting the model, learning rate, image augmentations, saving training parameters and maximum iterations.

Training implemented on a GPU card NVIDIA RTX 2080 TI for 5,000 iterations using resolution 608\*608 pixels, and network parameters will be saved every 100 iterations. Dataset will be subjected to simple image augmentation by adjusting the hue, saturation and exposure of the images.

Learning rate defines how the optimizer during training will adapt to solve the problem, so large learning rate will result in rapid changes in the values of the weights thus lead to fewer iterations but may not reach the global minimum of the problem, and the lower learning rate is vice versa. Thus, the dynamic learning rate is used.

If a static value of learning rate is used at the beginning most likely, the model will diverge as the gradient is not stable. Authors of YOLO proposed to start by burn-in stage to overcome this problem [21]. Based on this, the dynamic learning rate consists of four stages. The four stages are burn-in, maximum learning rate, 1<sup>st</sup> learning rate adjustment and 2<sup>nd</sup> learning rate adjustment.

The burn-in stage is for the first 300 iterations where the learning rate is increased from 0 to 0.001, and then the maximum learning rate stage starts till the 1500<sup>th</sup> iteration, where learning rate value is 0.001. The first learning rate adjustment is applied at the 1,500<sup>th</sup> iteration by where learning rate become 0.0001 which will provide fine-tuning of the training parameters and at the 2,500<sup>th</sup> iteration the second learning rate adjustment where learning rate is 0.00001 providing finer tuning of the training parameters.

Overfitting a network means that the network is performing very good on the training dataset, but in test dataset and other datasets, the performance degrades. YOLO uses two parameters to overcome the overfitting, by using the momentum that will penalize any large changes of weights between iterations, and decay parameters to penalize weights with large values.

YOLO Feature extraction is based from a deep convolutional neural network (CNN) of 53 CNN layers called DarkNet-53, that has been pre-trained on the ImageNet database [19]. Thus, initiating YOLO training is an advantage of YOLO as it does not start learning from scratch or using random values for the parameters. Instead, it uses a pre-trained model parameters values of its backbone convolutional layers (Darknet-53), which is a form transfer learning.

Table 9. Fish object detection training settings.

Category	Configuration item	Configuration value
Network	Deep learning network	YOLO
Hardware	GPU card used	NVIDIA RTX 2080 TI
Training resolution	Image resolution during training	608*608 pixels
Network settings to avoid overfitting	Momentum	0.9
	Decay	0.0005
Learning rate adjustment	Learning rate	0.001
	Burn in	300 Iterations
	Adjust the learning rate at which iteration	At 1,500 <sup>th</sup> iteration And at 2,500 <sup>th</sup> iteration
	Scales of adjustments	0.1 (at 1,500 <sup>th</sup> iteration) 0.1 (at 2,500 <sup>th</sup> iteration)
Image augmentations	Saturation	1.5
	Exposure	1.5
	Hue	0.1
Data saving	Save data every	100 Iterations
Maximum training iterations	Maximum iteration	5,000 Iteration

## 5.2 Dataset

Using a collected dataset of 7,162 fish images, with the fish types trout, brown trout, chub, barbel, salmon and common nase. Adding 7,162 underwater images without fish objects to obtain a balanced dataset. With image augmentation, the total number of images reached 64,458 images.

This dataset will be used in training YOLO based on raw images, and then apply the image enhancement algorithm mentioned in chapter 4 to produce the dataset with enhanced images and train another model on it.

### 5.2.1 Original Datasets Images

The base dataset is an important asset in training deep learning network not only in achieving good training results but, also in developing a robust model with a variety of possibility to identify the object whenever it exists on the image. So, if a dataset only contained good images with one fish on white background the model will provide good training results and testing dataset if the same will provide good mAP and IOU, but later on real-life usage, the result will be disappointing due to that the fish may be similar to background, fish are overlapping, image with noise and many other difficulties.

In this thesis, the author aimed to develop model trained on a realistic dataset with a different variety of fish types. Images are taken from field studies in both freshwater river environments as well as salmon aquaculture sea cages and captured by different cameras types such as coloured and infrared cameras. The author extracted the image dataset to contain as many possible good and challenging situations. The dataset contains fish images with high turbidity, multi overlapping fish, complex background textures, parts of fish, noisy images, different positions and orientations of the fish and with clear details of the fish.

Although using such a dataset may degrade the metrics of model evaluation because it contains poor quality imagery, it will benefit the research community in the end by having a robust model fit enough to identify fish in real-life imagery. A short example of these are in Table 10, and more examples exist in Appendix 1 – Fish Image Dataset Examples.

The reason behind using a balanced dataset is to develop a robustly trained model, with the ability to identify the object and as well to learn that there is a possibility of no objects exist in the image, which will benefit the mAP by decreasing false positive on background image, especially that YOLO has less error of false positive on background image, as proven by the study in [27] comparing between fast YOLO and fast RCNN.

Table 10. Images types of fish in the dataset without image enhancement and with image enhancement, samples used in training like images that have good fish details, overlapping fish and fish like the background.

Fish type/Dataset	The dataset with image enhancement	The dataset without image enhancement
Good fish details		
Overlapping fish		
Fish in a complex background		

### 5.2.2 Dataset Image Augmentation

Dataset image augmentation increases image dataset diversity, by performing different image processing operations introducing different position or image quality of the object to be detected, thus increasing the image dataset helping to generalize the training of YOLO, for better object detection capabilities.

Image processing operations used are image rotation, scaling flipping along horizontal axes, flipping across vertical axes, intensity modification, cropping and adjust hue and saturation values. All these operations will be adjusted by random values within a configured range, more details in Table 11.

Table 11. Image processing operations configuration to introduce image augmentation.

Image processing operation	Configuration bounds
Cropping	Random crop up to 20% of the image.
Rotation	Rotate by random values from $\pm 20$ degrees
Intensity modification	Multiply by random value from 0.2 to 1.5 making images much darker or whiter
Horizontal flip	Possibility of 30% of images to horizontal flip an image.
Vertical flip	Possibility of 30% of images to vertical flip an image.
Hue and saturation adjust	Add random values from -60 to 60 to hue and saturation of the image.
Gaussian noise	Add noise sampled from Gaussian distribution with random standard deviation values from 10 to 60.
Scale	Image scaling from 0.5 to 0.7 of image size.

Every image in both datasets will be used to produce four augmented images, each image has different set of operations as shown in Table 12, and these operations are depending on random values within the specified range, so some of these operations may not be implemented like the horizontal image flip will be applied to 30% of the images.

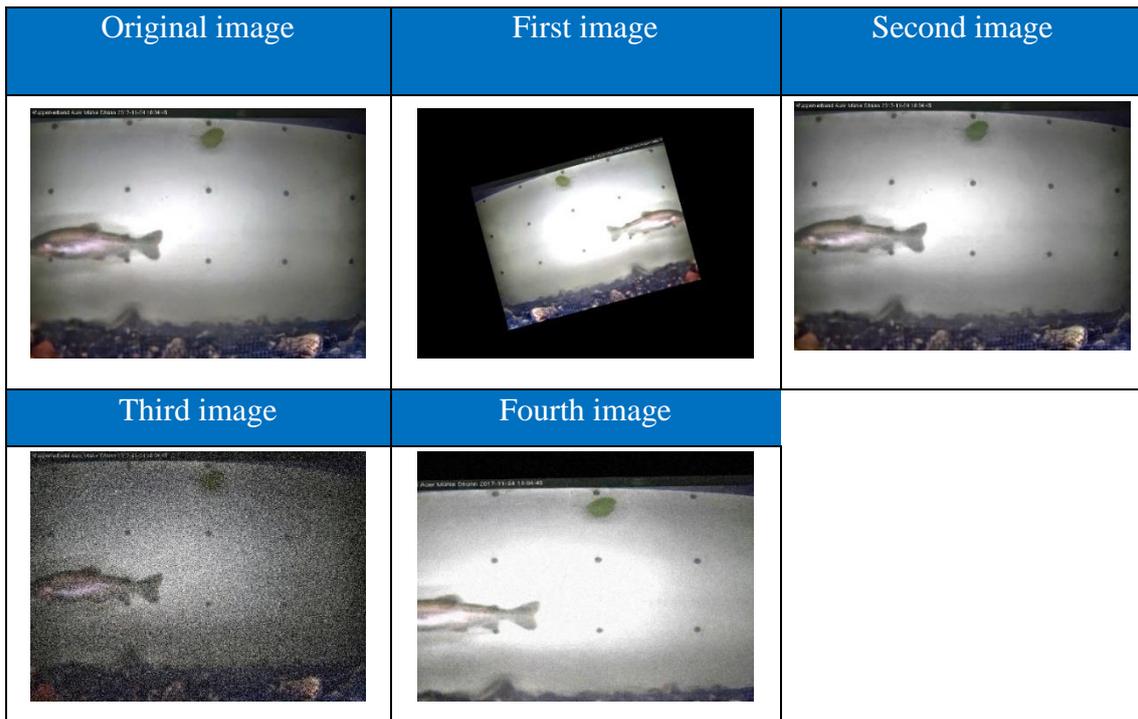
The first image is a combination of scaling, rotation, horizontal flip, vertical flip and intensity modification of the original image. The second image is a combination of a horizontal flip, vertical flip and hue and saturation adjustment of the original image. The third image is a combination of adding Gaussian noise, horizontal flip, vertical flip and intensity modification and the fourth image is a combination of all mentioned operations.

An example of image augmentation results of an original image is presented in Table 13. The code for these operations is in [25].

Table 12. Image augmentation will be used for every image in the datasets to generate four images. Every image will be the result of shown different image processing operations.

Augmented images	Cropping	Rotation	Intensity modification	Horizontal flip	Vertical flip	Scale	Hue and saturation adjust	Gaussian noise
First image		✓	✓	✓	✓	✓		
Second image				✓	✓		✓	
Third image			✓	✓	✓			✓
Fourth image	✓	✓	✓	✓	✓	✓	✓	✓

Table 13. Image augmentation results of example original image.



### 5.2.3 Final Dataset Information

Final set information is shown in Table 14, showing the total count of images used for training and testing, with showing how many images have fish and how many without a fish object.

Table 14. Fish images datasets used for training and testing for YOLO training with and without image enhancement.

Dataset / Image count		The dataset without image enhancement	The dataset with image enhancement
Training images count	Total	64,458 images	64,458 images
	With fish	32,229 images	32,229 images
	No fish	32,229 images	32,229 images
Test images count	Total	7,162 images	7,162 images
	With fish	3,581 images	3,581 images
	No fish	3,581 images	3,581 images

### 5.3 Fish Detection Using YOLO Training and Results

YOLO was configured with settings and environment described in 5.1, is trained on datasets previously mentioned in 5.2, will provide two models which here will present the training and evaluation metrics, to compare the training on underwater images with and without image enhancement.

Training YOLO on image dataset with or without image enhancement using settings in Table 9, took almost 8 hours for 5,000. The Dynamic rate adjustment stages explained in 5.1 is presented in Figure 19.

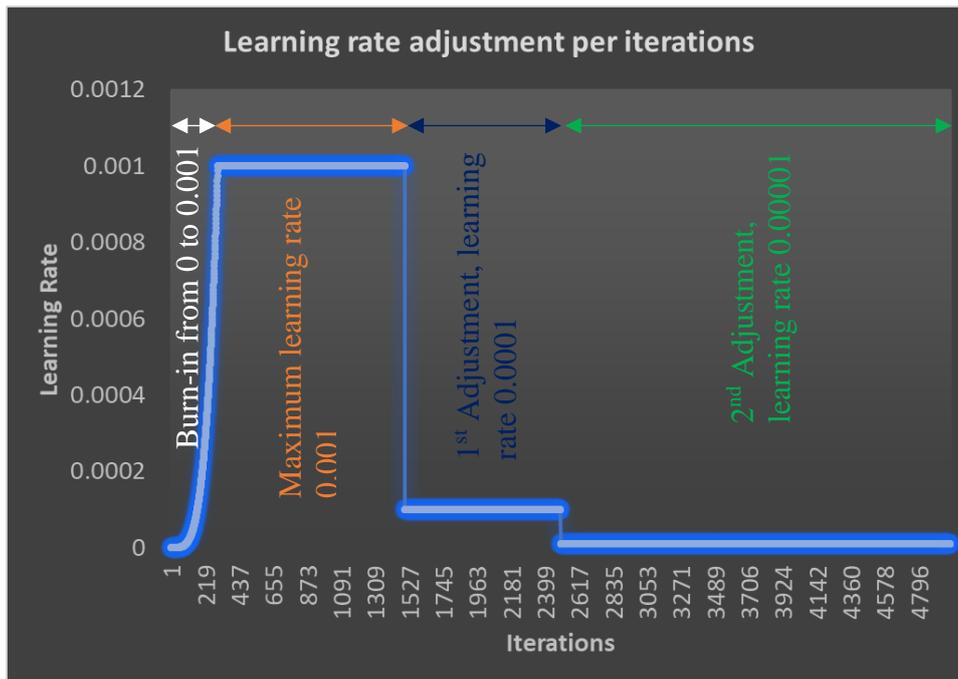


Figure 19. Learning rate adjustment per iterations showing the different values of learning rate starting by burn-in stage from iteration zero to 300<sup>th</sup> values are increased from 0 to 0.001 and at the 1500<sup>th</sup> iteration changed to 0.0001 till iteration 2,500<sup>th</sup> changed to 0.00001 for better tuning of weights.

Initiating the training YOLO will use a pre-trained model's parameters values to be a starting point, then assess the results using the loss function which usually starts with high loss, then by optimizing this training model's parameters value, the model will start adapting to the problem, and the loss will decrease.

As achieved during training without image enhancement, the average loss started from 4,170 to 0.17 while training with image enhancement the average loss started from 4,835 to 0.16 as shown in Figure 20, which has the average loss limited from 0 to 0.5 to show more details better. These results on both models show that both models are adapted to the problem with the almost same performance, with slight improvement with image enhancement.

But the ability of the optimizer to adapt to the problem and solve it does not mean that the trained model can properly detect this object. This can be due to multiple reasons, from that training dataset was not enough to include all possible scenarios describing that object, to the fact that the quality of the image dataset is not sufficient to extract the object feature efficiently.

To efficiently assess the trained models, it is necessary to assess each model on a test dataset that may have similar characteristics of the training dataset. And a better approach is to test object detection on a completely different dataset containing this object, as this reflects how well the training was able to extract object feature during training, and later during testing to make successful detection. Both approaches are used in this thesis to assess the trained models.

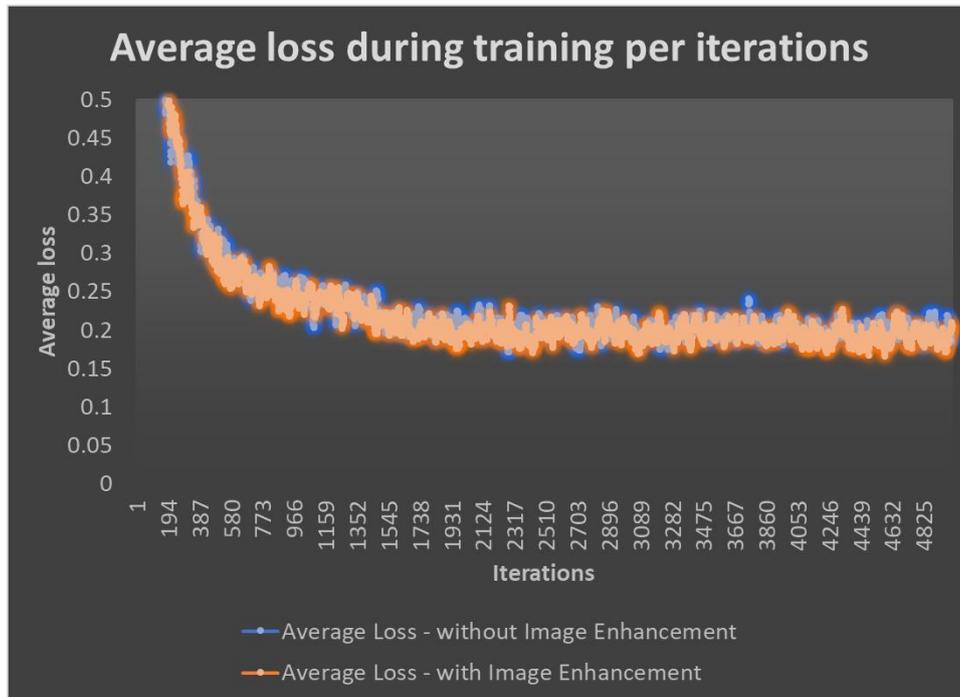


Figure 20. Average loss during training per iterations for training with and without image enhancement, showing the average loss from 0 to 0.5 for better representations.

The metrics used to assess both models' classification and localization performance are the mean average precision (mAP) and intersection over union (IOU), respectively. Both will be calculated where the threshold is that confidence score is more than 50%, on the test dataset every 402 iterations, which can be an indirect way to ensure not overfit the model.

Summarized results are shown in Figure 21 and Figure 22, shows the mAP and IOU respectively, starting from the 1206<sup>th</sup> iteration for better representations as below this iteration training is not efficient. The model trained with image enhancement shows better performance than the other model trained on a raw dataset without image enhancement. As mAP increased by 5% and IOU increased by 3%.

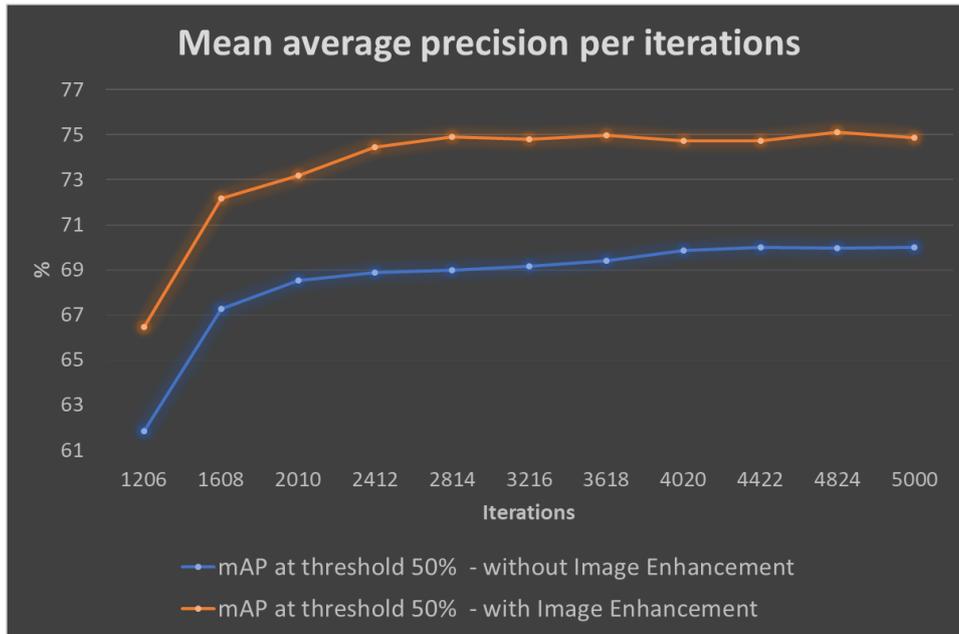


Figure 21. Mean average precision (mAP) per iterations from the 1206<sup>th</sup> iteration to 5000<sup>th</sup> iteration, comparing mAP for training with image enhancement increased by 5% more than training without image enhancement.

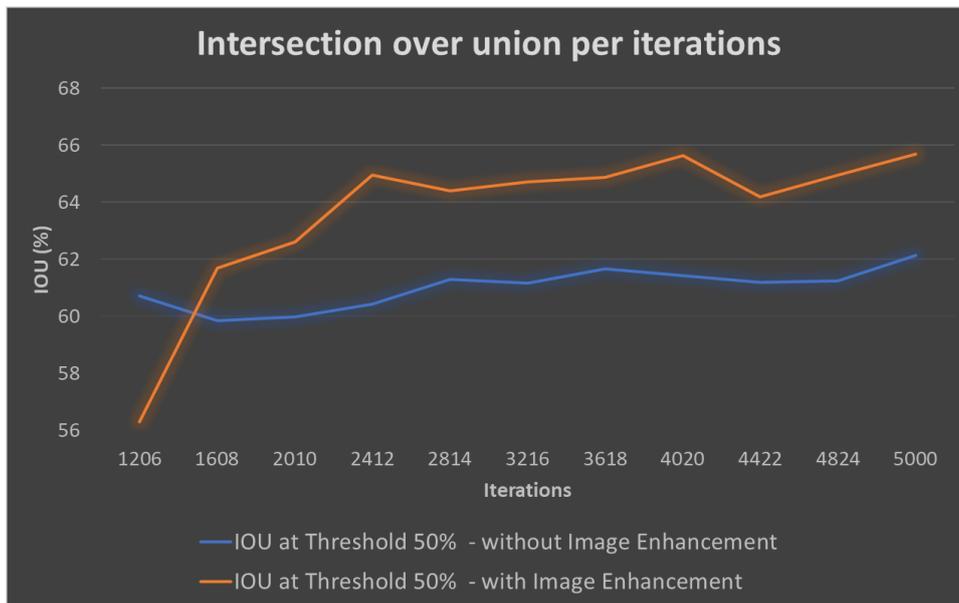
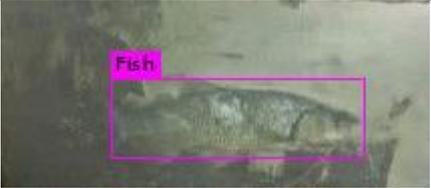
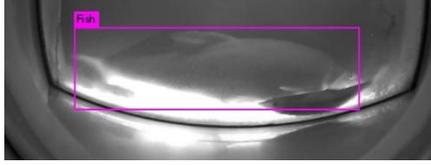
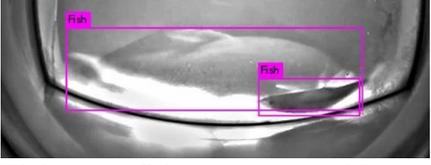
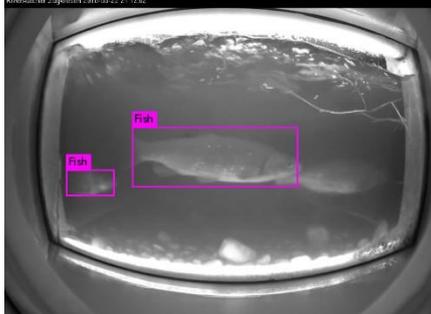
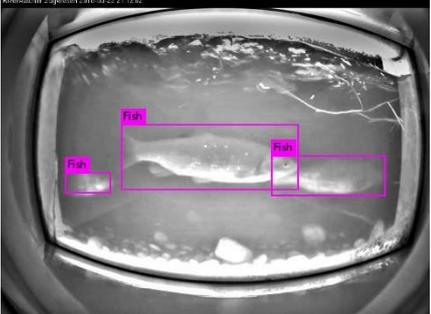


Figure 22. Intersection over Union (IOU) per iterations from the 408<sup>th</sup> iteration to 5000<sup>th</sup> iteration, showing IOU for training with image enhancement has increased IOU by 3%.

The confidence score reflects how much is classification confident about having an object of this type. A closer look at confidence score by checking the result on a few images, from the testing dataset, is presented in Table 15, shows that the training and detection on images with image enhancement have higher confidence score. This explains the better

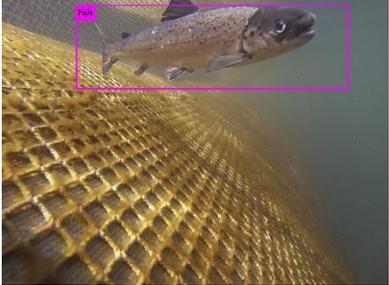
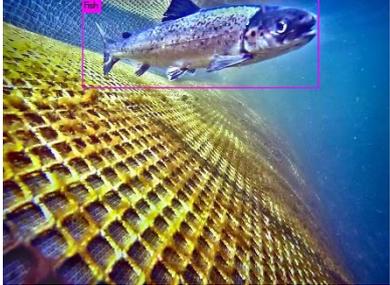
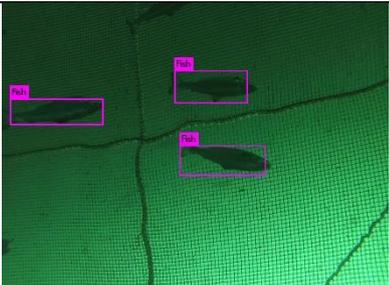
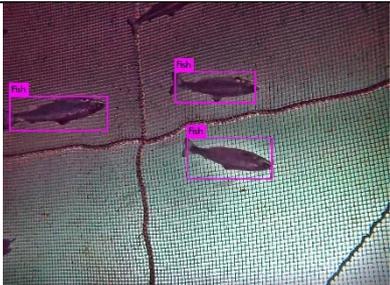
detection of the dataset with image enhancement, as more objects will be correctly identified with image enhancement due to better feature extraction. Also detected object has a better IOU compared to the ground truth.

Table 15. Example of fish detection from the test images in dataset shows that fish detection using image enhancement have higher confidence score and better detection of the fish (better IOU compared to ground truth).

	Fish detection without image enhancement	Fish detection with image enhancement
Example 1:	 <p>Confidence score = 73%</p>	 <p>Confidence score = 76%</p>
Example 2:	 <p>Fish 1 confidence score = 87% Fish 2 not detected</p>	 <p>Fish 1 confidence score = 87% Fish 2 confidence score = 31%</p>
Example 3:	 <p>No detection</p>	 <p>Confidence score = 45%</p>
Example 4:	 <p>Fish 1 confidence score = 34% Fish 2 confidence score = 66% Fish 3 not detected</p>	 <p>Fish 1 confidence score = 36% Fish 2 confidence score = 70% Fish 3 confidence score = 60%</p>

But for real testing and better checking of overfitting of the model, Table 16 shows results based on images from marine aquaculture sea cage datasets, which are very different than the imagery used for training and testing. This validation test clearly shows that fish detection after applying image enhancement continued to have higher a confidence score and better localisation of the fish, as it had a higher IOU.

Table 16. Example of fish detection on images not from the training dataset or test datasets, it is from a different video set, fish detection using image enhancement have higher confidence score and better detection of the fish (better IOU compared to ground truth).

	Fish detection without image enhancement	Fish detection with image enhancement
Image 1 result	 <p>Confidence score = 41%</p>	 <p>Confidence score = 61%</p>
Image 2 result	 <p>Confidence score = 74%, 68%, 42%</p>	 <p>Confidence score = 79%, 77%, 51%</p>

The result showing in Table 17 has the best metrics for each model evaluated on the test datasets, showing that the training with image enhancement introduced better results with 5% mAP increase and 3% IOU increase. Proving that during training and testing with image enhancement, the object feature extraction process was more efficient than without image enhancement.

Table 17. Best training results for both training with and without image enhancement.

	Training without image enhancement	Training with image enhancement
Mean average precision	70.01%	75.11 %
Intersection over union	62.13%	64.95 %

Image enhancement improved the object detection metrics, and the usage of the dataset collected from the field studies improved the robustness of the detection in real-life.

Although the metrics of training are considered excellent compared to YOLO performance on different datasets, the author thinks that these results can be further improved in two streams. The first stream is improving the detection metrics by enhancing YOLO version three setbacks from better detection of large objects as mention in [19], to improving IOU by better object localization [27]. The second stream increased the dataset by introducing more fish types, more fish scales in the images and more fish type and instances.

## 6 Conclusion

A robust trained model for underwater fish detections included multiple stages, and every stage needs to be carefully developed to achieve a model which can identify and localize objects in real-life images. The trained model achieved an overall performance of 75.11 % for mean average precision (mAP) and 64.95% for intersection over union by IOU for object classification and localisation, respectively.

The dataset collected for this work was unique, and included some 7,162 fish images with a diversity of site locations, cameras and water types, with good and difficult object representation, and adding to this diversity image augmentation and underwater images without fish to achieve a balanced dataset of 64,458 images, which improved the trained model robustness in real-life object detection as this dataset is significantly better than those found in the scientific literature.

Underwater imagery faces many difficulties degrading the scene representation on the image, which will decrease the ability of the trained model to extract object features efficiently. Based on this, and in contrast to previous research, we introduced image enhancement on the dataset to improve the object feature extraction, and by extension, the object detection metrics. Underwater image enhancement, enhanced the object classification and localization by increasing mean average precision (mAP) by 5% and intersection over union by IOU by 3%, respectively.

Existing underwater image enhancement algorithms are often not suitable for real-world field imagery, and the author enhanced the dataset developed in this work using the algorithms in [5], [7], which showed a lack of correct non-uniform illumination and colour correction, thus proposed a new algorithm eliminating non-uniform illumination and correctly white balancing the image with sharpness and contrast enhancement.

Non-uniform illumination was eliminated by using an emphasis homomorphic bandpass filter. This decreases the frequency where the illumination variation exists, using a bandpass filter to optimise the frequency filtering. This is an advancement over previous research in [8] that uses high-pass filtering resulted in decreasing the non-uniform illumination with better image details and contrast.

The author used the simplest colour balancing algorithm for performing white balance operation on the image and introduced a new optimization method for selecting the clipping thresholds per channel, improving the colour correction of the image.

In this thesis, the proposed algorithm enhanced the quality of the dataset, and thus improved the ability to extract features using deep neural networks, yet there is room for improvements, and the author proposes the following three suggestions for future work:

The first suggested improvement is to increase the dataset size and diversity of fish types, physical scales, orientations, water types and lighting. The second is to improve the YOLO code for object localization, improve the detection of large objects, and finally segmentation. The third is to develop and train a deep learning network for underwater image enhancement based on the proposed algorithm, which the author is trying to develop but not yet finished. With these three improvements, a trained model able to enhance images and perform object detection and segmentation and can be trained to identify different types of fish, ideally with a performance which vastly exceeds a human in both speed and accuracy.

## 7 Bibliography

- [1] N. Dale, *National research council nutrient requirements of poultry — ninth revised edition (1994)*, vol. 3, no. 1. 1994.
- [2] T. Smolinski, M. Milanova, A.-E. Hassanien, D. Magness, F. Huettmann, and J. Morton, “Applications of Computational Intelligence in Biology,” *Appl. Comput. Intell. Biol.*, vol. 122, no. May, pp. 183–207, 2008.
- [3] X. Li, M. Shang, H. Qin, and L. Chen, “Fast accurate fish detection and recognition of underwater images with Fast R-CNN,” *Ocean. 2015 - MTS/IEEE Washingt.*, pp. 1–5, 2016.
- [4] H. Qin, X. Li, J. Liang, Y. Peng, and C. Zhang, “DeepFish: Accurate underwater live fish recognition with a deep architecture,” *Neurocomputing*, vol. 187, pp. 49–58, 2016.
- [5] A. S. Abdul Ghani and N. A. Mat Isa, “Underwater image quality enhancement through integrated color model with Rayleigh distribution,” *Appl. Soft Comput. J.*, vol. 27, pp. 219–230, 2015.
- [6] S. Anwar, C. Li, and F. Porikli, “Deep Underwater Image Enhancement,” pp. 1–12, 2018.
- [7] C. Ancuti, C. O. Ancuti, T. Haber, and P. Bekaert, “Enhancing underwater images and videos by fusion,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 81–88, 2012.
- [8] A. S. Abdul Ghani and N. A. Mat Isa, “Homomorphic filtering with image fusion for enhancement of details and homogeneous contrast of underwater image,” *Indian J. Geo-Marine Sci.*, vol. 44, no. 12, pp. 1904–1919, 2015.
- [9] R. C. Gonzalez and R. E. Woods, “Introduction,” in *Digital Image Processing*, 2nd ed., Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1992, pp. 1–31.
- [10] R. Maini and H. Aggarwal, “A Comprehensive Review of Image Enhancement Techniques,” *CoRR*, vol. abs/1003.4, pp. 8–13, 2010.
- [11] P. Singh and R. Pandey, “A Comparative Study to Noise Models and Image Restoration Techniques,” *Int. J. Comput. Appl.*, vol. 149, pp. 18–27, 2016.
- [12] R. C. Gonzalez and R. E. Woods, “Image Enhancement in the Spatial Domain,” in *Digital Image Processing*, 2nd ed., Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1992, pp. 76–142.
- [13] N. Bt Shamsuddin, W. F. Bt Wan Ahmad, B. B. Baharudin, M. K. B. M. Rajuddin, and F. Bt Mohd, “Significance level of image enhancement techniques for underwater images,” *2012 Int. Conf. Comput. Inf. Sci. ICCIS 2012 - A Conf. World Eng. Sci. Technol. Congr. ESTCON 2012 - Conf. Proc.*, vol. 1, pp. 490–494, 2012.
- [14] L. J. Johnson, “The Underwater Optical Channel,” 2015. [Online]. Available: [https://www.researchgate.net/publication/280050464\\_The\\_Underwater\\_Optical\\_Channel](https://www.researchgate.net/publication/280050464_The_Underwater_Optical_Channel). [Accessed: 15-Dec-2019].
- [15] Jerlov, *N. G. Marine Optics*. Amsterdam: Elsevier, 1976.
- [16] H. Wu, Q. Liu, and X. Liu, “A Review on Deep Learning Approaches to Image Classification and Object Segmentation,” vol. 1, no. 1, pp. 1–5, 2018.
- [17] Z. Q. Zhao, P. Zheng, S. T. Xu, and X. Wu, “Object Detection with Deep Learning: A Review,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 30, no. 11, pp. 3212–3232, 2019.
- [18] Z. Huang and J. Wang, “DC-SPP-YOLO: Dense Connection and Spatial Pyramid Pooling Based YOLO for Object Detection,” *CoRR*, vol. abs/1903.0, 2019.
- [19] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement,” *CoRR*, vol. abs/1804.0, 2018.
- [20] J. Redmon and A. Farhadi, “YOLO: Real-Time Object Detection.” [Online]. Available: <https://pjreddie.com/darknet/yolo/>. [Accessed: 11-Dec-2019].
- [21] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi, “You Only Look Once:

- Unified, Real-Time Object Detection,” *CoRR*, vol. abs/1506.0, 2015.
- [22] D. P. Kingma and J. Ba, “Adam: A Method for Stochastic Optimization,” pp. 1–15, 2014.
- [23] Z. Hao, “MATLAB Code of Underwater Image Enhance via Fusion,” *Github*, 2017. [Online]. Available: <https://github.com/IsaacChanghau/OptimizedImageEnhance/tree/master/matlab/FusionEnhance>. [Accessed: 11-Nov-2019].
- [24] N. Limare, J.-L. Lisani, J.-M. Morel, A. B. Petro, and C. Sbert, “Simplest Color Balance,” *Image Process. Line*, vol. 1, pp. 297–315, 2011.
- [25] A. Moutaz, “Underwater image enhancement algorithm with deep learning trials,” *Gitlab*, 2019. [Online]. Available: [https://gitlab.cs.ttu.ee/moabos/uw\\_deeplearning\\_objectdetection.git](https://gitlab.cs.ttu.ee/moabos/uw_deeplearning_objectdetection.git). [Accessed: 21-Dec-2019].
- [26] Alexey and J. Redmon, “Yolo-v3 and Yolo-v2 for Windows and Linux,” *Github*, 2019. [Online]. Available: <https://github.com/AlexeyAB/darknet>. [Accessed: 11-Dec-2019].
- [27] K. G. Shreyas Dixit, M. G. Chadaga, S. S. Savalgimath, G. Ragavendra Rakshith, and M. R. Naveen Kumar, “Evaluation and evolution of object detection techniques YOLO and R-CNN,” *Int. J. Recent Technol. Eng.*, vol. 8, no. 2 Special issue 3, pp. 824–829, 2019.

## Appendix 1 – Fish Image Dataset Examples

Table 18. Images types of fish in the dataset without image enhancement and with image enhancement, samples used in training like images that have a good fish details, foggy fish shape, overlapping fish, fish like the background, parts of fish.

Fish type/Dataset	The dataset with image enhancement	The dataset without image enhancement
Good fish details		
Overlapping fish		
Overlapping fish (2 fish)		
Fish like background		

Fish type/Dataset	The dataset with image enhancement	The dataset without image enhancement
Foggy fish shape		
Part of the fish		
Infrared camera image		
Infrared camera image (noisy image)		

## Appendix 2 – Underwater Image Enhancement Results

Table 19. The proposed image enhancement algorithm results on different scenes showing input and output image.

Input image	Based on the algorithm in [5]	Based on the algorithm in [7]	The proposed algorithm output image
		