Universidad de Málaga

Escuela Técnica Superior de Ingeniería de Telecomunicación

Programa de Doctorado en Ingeniería de Telecomunicación

TESIS DOCTORAL

Nephrops norvegicus burrows detection and classification from underwater videos using Deep Learning techniques

Autor:

Atif Naseer

Directores:

Enrique Nava Baro

Yolanda Vila Gordillo

Málaga 2024

Escuela de Doctorado
doctorado@uma.es

DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD DE LA TESIS PRESENTADA PARA OBTENER EL TÍTULO DE DOCTOR

Atif NASEER

Estudiante del programa de doctorado INGENIERÍA DE TELECOMUNICACIÓN de la Universidad de Málaga, autor de la tesis, presentada para la obtención del título de doctor por la Universidad de Málaga, titulada: "*Nephrops norvegicus* burrows detection and classification from underwater videos using deep learning techniques", realizada bajo la dirección del Dr. Enrique Nava Baro y de la Dra. Yolanda Vila Gordillo,

DECLARO QUE:

La tesis presentada es una obra original que no infringe los derechos de propiedad intelectual ni los derechos de propiedad industrial u otros, conforme al ordenamiento jurídico vigente (Real Decreto Legislativo 1/1996, de 12 de abril, por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia), modificado por la Ley 2/2019, de 1 de marzo.

Igualmente asumo, ante a la Universidad de Málaga y ante cualquier otra instancia, la responsabilidad que pudiera derivarse en caso de plagio de contenidos en la tesis presentada, conforme al ordenamiento jurídico vigente.

En Málaga, a 3 de noviembre de 2023

Fdo.: Atif NASEER

Fdo.: Pablo Otero Roth (tutor)

Fdo.: Enrique Nava Baro
(director)

Fdo.: Yolanda Vila Gordillo
(directora)

This page intentionally left blank.

UNIVERSIDAD DE MÁLAGA

AUTORIZACIÓN DE LOS DIRECTORES DE TESIS DOCTORAL

El alumno del Programa de Doctorado en Ingeniería de Telecomunicación, Atif Naseer, con pasaporte nº PAK DB9821862, es primer autor de las siguientes publicaciones en revistas indexadas en los Journal Citation Reports (JCR) del Web of Science (WoS):

- **Naseer, A.**; Nava Baro, E.; Daud Khan, S.; Vila, Y.; Doyle, J. (2022) "Automatic detection of *Nephrops norvegicus* burrows from underwater imagery using deep learning," *Computers, Materials & Continua*, vol. 70, no.3, pp. 5321–5344, 2022. https://doi:10.32604/cmc.2022.020886.
- **Naseer, A.**; Nava Baro, E.; Daud Khan, S.; Vila, Y. (2022) "A novel detection refinement technique for accurate identification of *Nephrops norvegicus* burrows in underwater imagery". *Sensors* 2022, 22, 4441. https://doi.org/10.3390/s22124441.

Estas publicaciones avalan su tesis doctoral y ninguna otra tesis.

Por todo ello, sus directores de tesis Enrique Nava Baro y Yolanda Vila Gordillo autorizan al Sr. Atif Naseer a depositar su tesis doctoral ante las Autoridades académicas de la Universidad de Málaga.

En Málaga, a 6 de noviembre de 2023.

Fdo: Enrique Nava Baro

Fdo: Yolanda Vila Gordillo

UNIVERSIDAD DE MÁLAGA

AUTORIZACIÓN DEL TUTOR

El alumno del Programa de Doctorado en Ingeniería de Telecomunicación, Atif Naseer, con pasaporte nº , ha realizado su tesis doctoral en la Universidad de Málaga y yo, Pablo Otero Roth, profesor titular de la Universidad de Málaga y profesor del Programa de Doctorado en Ingeniería de Telecomunicación de la Escuela de Doctorado de la Universidad de Málaga, he sido su tutor.

La tesis doctoral de Atif Naseer cumple los requisitos establecidos por la Escuela de Doctorado de la Universidad de Málaga y por el Programa de Doctorado en Ingeniería de Telecomunicación.

Por el presente escrito, autorizo a su autor, Atif Naseer, a depositar su tesis doctoral.

En Málaga, a 7 de noviembre de 2023.

Fdo: Pablo Otero Roth

I dedicate this thesis to the cherished memory of my late mother, whose boundless love, unwavering guidance, and profound wisdom continue to shape and inspire me. Her unwavering support and belief in my dreams will forever be the driving force behind my academic achievements and personal growth.

This page intentionally left blank.

.

Acknowledgements

I want to express my deepest gratitude and appreciation to all those who have supported and contributed to the completion of this dissertation.

First and foremost, I am immensely grateful to my supervisors, Enrique Nava Baro and Yolanda Vila Gordillo, and my mentor, Pablo Otero, for their invaluable guidance, expertise, and unwavering commitment to my research. Their insightful feedback, encouragement, and patience have been instrumental in shaping the direction and quality of this work.

To my parents, thank you for instilling in me a love for learning, nurturing my curiosity, and always believing in my abilities.

My loving wife, thank you for your endless patience, understanding, and unwavering support. Your belief in me, your words of encouragement, and your willingness to listen and provide valuable insights have been my constant motivation.

To my children, thank you for understanding and adapting to this dissertation's demands on our family. Your love, hugs, and laughter have provided much-needed joy and balance during some of the most challenging times.

I am indebted to the Spanish Oceanographic Institute, Cadiz, Spain, and Marine Institute, Galway, Ireland, for providing the necessary dataset for research.

I am thankful to Jennifer Doyle from Marine Institute, Galway, Ireland, for helping me with the annotation validations.

Lastly, I want to express my deep appreciation to all the participants and individuals of WGNEPS who generously contributed their time, expertise, and insights to this study. Their valuable contributions have been fundamental to the success of this research.

I extend my sincerest thanks to everyone mentioned above and those who may have inadvertently been left unmentioned. Your support, encouragement, and contributions have been invaluable, and I am truly grateful for your presence in my academic and personal life.

This page intentionally left blank.

Abstract

Problems faced by marine scientists during the assessment of *Nephrops norvegicus* species during underwater TV surveys have been addressed in this thesis. One of the main contributions of the work has been the study of the behavior of deep learning algorithms on the complex underwater dataset.

Currently, the *Nephrops* data are collected through the UWTV surveys and are reviewed manually by trained experts. Burrows systems are quantified following the protocol established by ICES.

Our first contribution is to develop the dataset for the deep learning models. No such dataset exists that someone can use to validate the results. After many revisions, the current work selected a few videos for annotation (the videos are selected with Marine experts based on the *Nephrops* burrows densities). The Marine expert validates each annotation before adding it to the dataset. After validating each annotation, a curated dataset is used for training and testing the model.

Different types of deep learning-based models have been finetuned and applied to the created dataset. The work proposed five different neural networks: MobileNet, Inception, ResNet50, ResNet101, and YOLOv3. All the models are trained and tested with the different combinations of datasets. A complete methodology is proposed for automatically detecting *Nephrops* burrows. The automatic detection algorithms could replace the human review of data, with the promise of better accuracy, coverage of more significant areas and higher consistency in the assessment.

Deep learning algorithms performed very well in identifying the burrows. Generic CNN-based object detectors still face challenges in underwater object detection. These challenges include image blurring, texture distortion, color shift, and scale variation, which result in low precision and recall rates. This thesis contributes by developing a Novel Detection Refinement Technique for Accurate Identification of *Nephrops norvegicus* burrows. The proposed technique is based on each detection's spatial–temporal value. When integrated with any detector, the proposed method consistently increased the performance. The performance was calculated using mAP.

Another contribution lies in the tracking and counting burrows. Multiple OpenCV tracking algorithms are applied to that task, but due to three significant challenges, these tracking algorithms fail to track the *Nephrops* burrow. The first challenge is the camera's movement. The second challenge is the characteristics and size of burrows that are not fixed. The third challenge is the angle/opening of the burrow. The traditional object-tracking mechanism is not very effective. We proposed the tracking and counting of burrows using the spatial-temporal

values of each burrow. The unique burrows are counted using the intersection values of detected burrows in consecutive frames.

From an experimental point of view, our contribution lies in comparing burrows detection with different models, the deep analytics and application of detection refinement algorithm by calculating the precision, recall and F1 score. The proposed tracking algorithm is also compared with the OpenCV tracking algorithms. All these experiments were performed for the different combinations of datasets and different levels of parameters. Results show that our approach has better results regarding burrows detections, refinements, tracking and counting of burrows.

Table of Contents

This page intentionally left blank.

Summary

The earth's ecosystem mainly comprises oceans, producing 50% oxygen and 97% water. Also, it is a significant source of our daily food as it provides 15% of proteins in the form of marine animals. There are more studies on terrestrial ecosystems than on marine ecosystems because it is more challenging to study the marine ecosystem, especially in the deeper sea areas. Also, studying the marine ecosystem is costly and requires special equipment and human expertise to research a particular area.

Research in underwater image analysis has gained popularity in many applications of marine sciences. There are various research directions in underwater image analysis, for instance, aquatic species classification and detections, seafloor image recognition, coral reef classification, and flora and fauna recognition. Underwater image analysis requires a set of images processing tasks, including underwater object detection, classification, visual content recognition, and image annotation of large-scale marine species. Certain challenges, such as turbidity, color variations, and illumination changes, make underwater environments difficult for the models to detect and classify the objects automatically.

Monitoring the habitats of marine species is challenging for biologists and marine experts. The environmental features, such as depth-based color variations and the movement of species, make it a challenge. Marine scientists used satellites, shipborne, and camera sensors several years ago to collect underwater species images. In recent years, with the advancement of technology, scientists have used underwater Remotely Operated Vehicles (ROVs), Autonomous Underwater Vehicles (AUVs), and sledge and drop frame structures equipped with high-definition cameras to record videos and images of marine species. These vehicles can capture high-definition photos and videos. Besides all this quality equipment, the underwater environment is still challenging for scientists and marine biologists. The two main factors which make it difficult are the free natural environment and variations of the visual content, which may arise from variable illumination, scales, views, and non-rigid deformations.

There are thousands of species in the ocean all over the world. One of Europe's most important commercial species is the Norway lobster, *Nephrops norvegicus*. The Norway lobster, *Nephrops norvegicus*, is one of the leading commercial crustacean fisheries in Europe, where in 2018, the total allowable catch (TAC) was set at 32,705 tons for International Council for the Exploration of the Sea (ICES) areas.

Nephrops norvegicus species (hereafter referred to as *Nephrops*) are distributed from 10 m to 800 m in depth in the Atlantic NE waters and the Mediterranean Sea, where sediment is suitable for constructing their burrows. *Nephrops* spend most of their time inside the burrows, and their emergence behavior is influenced by time of year, light intensity, and tidal strength. These burrows can be detected through optimal lighting set-up during video recordings of the seabed.

The burrows themselves can be easily identified from surface features once specialist training has been taken. This species excavates into and inhabits burrow systems mainly in muddy seabed sediments, with more than 40% silt and clay.

A *Nephrops* burrow system typically can have single or multiple openings to different tunnels. A unique individual is assumed to occupy a burrow system. Burrows show signature features that are specific to *Nephrops*. The burrow features are summarized as follows:

1. At least one burrow opening is particularly half-moon shape.
2. There is often proof of expelled sediment, typically in a wide delta-like 'fan' at the tunnel opening, and scratches and tracks are frequently evident.
3. The centre of all the burrow openings has a raised structure.
4. *Nephrops* may be present (either in or out of the burrow)

The *Nephrops* burrow system comprises one or more burrows of the abovementioned characteristics. The presence of more than one burrow nearby doesn't mean the presence of more than one *Nephrops*.

Nephrops spend most of their time inside the burrows, and their emergence behaviour is influenced by several factors: time of year, light intensity, or tidal strength. For this reason, abundance indices obtained from the commercial catch or the traditional bottom trawl surveys are considered poorly representative of the *Nephrops* population and are not considered appropriate.

The abundance of *Nephrops* populations is currently monitored by underwater television (UWTV) surveys on many European grounds. The methodology used in UWTV surveys was developed in Scotland in the 1990s and is based on identifying and quantifying the burrow systems over the known area of *Nephrops* distribution. *Nephrops* abundance from UWTV surveys is the basis of assessment and advice for managing these stocks.

Videos are recorded using a camera system mounted on the sledge with an angle to the bottom ranging between 37–60° depending on the country. The recorded videos are saved in the DVDs, which are later reviewed manually by the trained marine experts and quantified by following the protocol established by the ICES.

The ICES is an international organization of 20 member countries. ICES is working on the marine sciences with more than 6,000 scientists from 7000 different marine institutes of the member countries. Multiple marine groups are working under the ICES umbrella. The Working Group on *Nephrops* Surveys (WGNEPS), formerly the Study Group on *Nephrops* Surveys (SGNEPS), is the coordinating expert group for *Nephrops* UWTV and trawl surveys. The expert group specializes in *Nephrops norvegicus* underwater television and trawl surveys within ICES. The group aims to provide international coordination for *Nephrops* UWTV and

trawl surveys in the North Atlantic. WGNEPS has focused on *Nephrops* planning, protocols, quality control, design, and survey development issues.

UWTV and Trawl surveys provide population estimates for *Nephrops* based on Functional Units (FU) in ICES areas and in a preliminary and exploratory way in some Geographical sub-areas (GSA) in the Mediterranean.

There are two main UWTV survey design approaches currently in use: grid (fixed or randomized) or stratified random design, where in some surveys, there is a buffering between stations to ensure better spatial coverage. Both approaches will allow the application of geostatistical models to estimate abundance and precision levels. Usually, the grid is extended adaptively until boundaries are established. The stratified random approach uses a priori data on sediment and or integrated VMS data to define strata with more similar densities. The definition of the survey boundaries and their stratification is essential to meet the required level of precision.

UWTV surveys were pioneered in Scotland in the early 90s to monitor the abundance of *Nephrops* populations. Estimating Norway lobster populations using this method involves identifying and quantifying burrow density over the known area of *Nephrops* distribution that can be used as an abundance index of the stock. *Nephrops* abundance from UWTV surveys is the basis of assessment and advice for managing these stocks. The UWTV surveys should be carried out annually when water clarity conditions are optimal, and weather conditions are likely to be calm. Surveys are not restricted to a particular time of day, and 24-hour operations can occur.

A sledge is used for the survey that is designed in Scotland. The sledge should be robust enough to secure all instruments, but the system must be flexible enough to adjust the balance. A proper light system on the sledge should be evenly distributed over the entire field of view. Two laser lights are fixed vertically on the sledge, showing the field of view.

Functional Units (FU) assess and manage *Nephrops* populations, where there is a specific survey for each FU. In 2019, 19 surveys covered the 25 FU's in ICES and one geographical subarea (GSA) in the Adriatic Sea. These surveys were conducted using standardized equipment and agreed protocols under the remit of WGNEPS. This study considers data from the Gulf of Cadiz (FU 30) and the Smalls (FU 22) *Nephrops* grounds to detect the *Nephrops* burrows using the image data collected from different stations in each FU using our proposed methodology.

The underwater environment is hard to analyze as it presents formidable challenges for Computer Vision and machine learning technologies. The image classification and detection in underwater images differ significantly from other visual data. Also, data collection in an aquatic environment is the biggest challenge. One reason for this is light, as light and water are not considered good friends, because when light passes through the water, it cannot absorb and reach the sea surface, which makes the images or videos a blurring effect. Also, scattering and

non-uniform lighting make the environment more challenging for data collection. Poor visibility is also a common problem in the underwater environment. The poor visibility is due to the ebb and flow of tides, which causes fine mud particles to suspend in the water column. The ocean current is another factor that causes frequent luminosity change. Visual features like lightning conditions, color changes, and low pixel resolution make it challenging. Some environmental elements, such as depth-based color variations and the turbidity or movement of species, make data collection very difficult in an underwater environment. Thus, two main factors which make it difficult are the free natural environment and variations of the visual content, which may arise from variable illumination, scales, views, and non-rigid deformations.

Currently, the *Nephrops* data are collected through the UWTV surveys and are reviewed manually by the trained experts. Many of the data were difficult to process due to complex environmental conditions. Burrows systems are quantified following the protocol established by ICES. The image data (which refers to video or still data) for each station is reviewed independently by at least two experts, and the counts are recorded for each minute onto the log sheet records. Each row of the log sheet records the minute, the number of burrows system count, and the time stamp. Count data are screened to check for any unusual discrepancies using Lin's Concordance Correlation Coefficient (CCC) with a threshold of 0.5. Lin's CCC measures the ability of counters to precisely reproduce each other's counts on a scale of 0.5 to 1, where 1 is perfect concordance. Only stations with a threshold lower than 0.5 were reviewed again by the experts.

With the massive amount of data collected for videos and images, manually annotating and analysing is laborious and requires a lot of data review and processing time. All stations manually analyse the UWTV surveys to classify and count the *Nephrops*. Due to limited human capabilities, the manual evaluation of image data requires a lot of time by trained experts to process the data to be quality-controlled and ready for use in the stock assessment. Due to these factors, only a limited amount of collected data is used for analysis that usually does not provide deep insights into a problem. Also, in some stations, it is tough for the human eye to classify and detect the burrows from a running video.

With the recent advancement in artificial intelligence and computer vision technology, many researchers employ AI-based tools to analyze marine species. Some people use feature extraction mechanisms to count and identify the species, while others use advanced techniques such as neural networks. Convolutional neural networks (CNN) bring a revolution in object detection. Deep convolutional neural networks gain tremendous success in object detection, classification, and segmentation tasks. These networks are data-driven and require a lot of labelled data for training.

The literature cannot provide any concrete solution to automatically detect and classify the *Nephrops* burrow system for habitat monitoring. This thesis is an effort to automate the existing method of *Nephrops* burrows counting. The work proposed a complete framework for

automatically detecting *Nephrops* burrows systems. The thesis work is divided into three major parts. The first part of the thesis shows the framework for automatic detection of *Nephrops* burrows using deep neural networks. Deep learning networks are used to automatically detect and classify the *Nephrops* burrows that take underwater video data as input and learn the hierarchical features from the input data to detect the burrows in each input video frame. The FU 30 and FU 22 datasets were collected using different image acquisition systems (Ultra HD 4 K video camera and HD stills camera) from *Nephrops* populations in 2018-19. At FU 22, 42 UWTV stations were surveyed in 2018. Out of 42, we used seven stations for data preparation. The sledge recorded 10–12 min videos at different frame rates ranging from 15 fps, 12 fps, and 10 fps at Ultra HD. Also, the high-definition images were captured with the camera. The images were recorded with a resolution of 2048 x 1152 pixels. At FU 30, the videos are recorded at 25 frames per second in good lighting conditions. Every station at FU 30 has 10–12 min recorded video footage. A total of 70 UWTV stations were surveyed in 2018. Out of 70 surveyed stations, 10 were rejected due to poor visibility and lighting conditions. Seven stations were selected for our experimentation with good lighting conditions, low noise and few artefacts, higher contrast, and a high density of *Nephrops* burrows. Data collected from FU 22 and FU 30 is converted into frames. The collected data set has a lot of frames with low and non-homogeneous lightning and poor contrast. The frames without burrows or poor visibility are discarded during the annotation phase, and consecutive frames with similar information are discarded. The next step is to annotate the collected data. Image annotation is a technique Computer Vision uses to create training and test ground truth data, as supervised deep learning algorithms require this information. Usually, any object is annotated by drawing a bounding box around it. Currently, the marine experts who work with *Nephrops* burrows are not using any annotation tool to annotate *Nephrops* burrows, as this is a time-consuming job. We used the Microsoft VOTT image annotation tool to annotate the burrows manually. The annotations are saved in the Pascal VOC. The saved XML annotation file contains the image name, class name (*Nephrops*), and bounding box details of each object of interest in the image. The annotated images are validated by marine sciences experts from the Gulf of Cadiz, Spain and Ireland. The validation of annotation is essential to obtain high-quality ground-truth information. This process took a long time as confirming every annotation is time-consuming and sensitive. After validating each annotation, a curated dataset is used for training and testing the deep neural network models. The annotated images are recorded into XML files and converted to TensorFlow (TF) Record files, a sequence of binary strings that TensorFlow requires to train the model. The dataset is divided into two subsets: train and test. The aim is to apply deep learning models to detect, classify, and count the *Nephrops* burrows automatically. Instead of training the network from scratch, this work utilized transfer learning to fine-tune the Faster R-CNN Inceptionv2, MobileNetv2, ResNet50, ResNet101, and YOLO v3 models in TensorFlow. Inceptionv2 is one of the architectures that have a high degree of accuracy. The

basic design of Inceptionv2 helps to reduce the complexity of CNN. We used a pre-trained version of the network model trained on the COCO dataset.

Inceptionv2 is configured to detect and classify only one class (c = 1), "*Nephrops*". The MobileNetv2 CNN architecture was proposed by Sandler et al. in 2018. One of the main reasons for choosing the MobileNetv2 architecture was the relatively small training dataset from FU 30. This architecture optimizes memory consumption and execution speed with minor errors. MobileNetv2 architecture has depth-wise separable convolution instead of conventional convolution. This architecture initially has a convolution layer with 32 filters, followed by 17 residual bottleneck layers. ResNet50 is a variant of the model ResNet. The ResNet50 has 48 convolutional layers, one max pool, and one average pool layer, so it is a 50-layer-deep convolutional network. The ResNet101 is a dense convolutional neural network that has 101 layers. YOLOv3 uses darknet to train the model. The darknet originally had 53 layers. In YOLOv3, another 53 layers are added to the darknet for detection, making 106 layers of fully convolutional architecture.

The model training, validation, and testing are conducted on a Linux Machine powered by an NVIDIA TitanXP GPU. Multiple combinations are created for model training, i.e., trained separate models for FU 22 and FU 30 datasets, training a model by combining both datasets (called hybrid model), and training and testing with different datasets. Models were trained using a random approximately 70–75% sample of the annotated dataset. The remaining is used for testing. The turning checkpoints during training are recorded after every 10k iterations, and the mAP50 is on the validation dataset. The model is evaluated using mAP, precision and recall curve, and visual inspection of the images with automatic detections. The model is tested to assess the performance. The models are tested against unseen images from the FU 30 and FU 22 datasets and evaluate the model's performance. The experiments show the evaluation of these networks quantitatively and qualitatively. Thirty different combinations of sets of experiments are performed with varying models of training. Each set is iterated seven times. So, 210 experiments were carried out. The models used 200 images from the FU 30 dataset to train the model, while 48 images were used to test the models.

Similarly, these models used 618 images from the FU 22 dataset for training and 359 for testing. The models trained using the FU 30 data set and tested using the FU 22 dataset used 200 images for training the model and 150 images for testing. The models that used the FU 22 data set for training and the FU 30 data set for testing used 618 images to train the model, while 200 images were for testing. Finally, the models that used the hybrid data set for training and testing used 818 images for training the model while 407 images for testing the model. Quantitatively, the work evaluates the performance of mAP, a prevalent metric in measuring object detector algorithms' accuracy, like Faster R-CNN, SSD, etc. Average precision calculates the average precision value for recall values over 0 to 1. Precision measures prediction accuracy, while Recall measures the positive predictions. The mAP is computed with the dataset of *Nephrops*

from FU 22 and FU 30 stations over 100k iterations. The work achieves a mAP higher than 75%, a positive indication to change the current paradigm of manual counting of *Nephrops* burrows. The performance of models is also evaluated using precision-recall curves. For model evaluation, true positive (TP), false positive (FP), and false negative (FN) annotations are calculated. The results prove that deep learning algorithms are a valuable and effective strategy to help marine science experts assess the abundance of *Nephrops norvegicus* species when underwater video/image surveys are carried out yearly, following ICES recommendations. The automatic detection algorithms could replace the tedious and manual review of data, which is nowadays the standard procedure, with the promise of better accuracy, coverage of more significant sampling areas, and higher assessment consistency. This work makes a considerable advancement for WGNEPS on the *Nephrops norvegicus* counting for stock assessment, where it is shown to detect and accurately count the *Nephrops* burrows automatically.

The second part of this thesis is talking about detection refinements. In the first part of the thesis, the models achieved good results in detecting the burrows from the image test data. However, when these trained models were tested on a video from the Gulf of Cadiz (FU 30), the accuracy of the detectors degraded. The problem is figured out with many FP and missed TP detections that adversely affect the accuracy of these models. This work proposes a detection refinement mechanism based on spatial-temporal information to enhance the detection of missed true positives and suppress false positive detections. The work presented in the literature used temporal information to track the faces and suppress false positive detections. Their approach used low-level tracking to detect the faces in real images. Furthermore, their approach does not recover the missed detections. In our problem, the low-level tracking cannot be applied as the *Nephrops* burrows are on the ground, where the characteristics are very different from the natural image. The previous work integrates temporal information to track the faces and suppress the false positives. In the proposed approach, spatial and temporal information is used to suppress the false positives and recover the missed detections. The work is divided into two parts. First, the model is trained using state-of-the-art Faster RCNN models Inceptionv2, ResNet50, and ResNet101 to detect *Nephrops* burrows. The work's second part applies the proposed spatial–temporal-based detection refinement algorithm. Each detected burrow's spatial and temporal information is obtained in a video sequence. This information is used across multiple frames to refine the *Nephrops* burrow detections. The spatial-temporal mechanism helped in suppressing the FP burrows. It allowed us to find the missed TP detection, achieving better accuracy and tracking and counting burrows in a video sequence. To address the detector's challenges, the work proposed "A Novel Detection Refinement Technique for Accurate Identification of *Nephrops norvegicus* Burrows in Underwater Imagery" based on spatial-temporal analysis that enhances the mAP of a generic detector. The proposed detection refinement mechanism identified the missed detections,

recovered them, and suppressed the false positives. Generally, our approach has the following contributions:

- The spatial-temporal filtering (STF) model extracts the spatial and temporal information of all the detections of the consecutive frames of an input video by suppressing the false positives and recovering the missed detections. The proposed method will improve the performance of the generic detectors (such as Inception and ResNet, in our case).
- Performance evaluation of the proposed framework on our proposed novel dataset. From the experiment results, the effectiveness of the proposed approach is presented.

This algorithm is divided into two sections, i.e., suppression of false positives and identification of missed detections. The results are evaluated by different experiments performed using the proposed detection refinement algorithm. The work uses three models (Inception, ResNet50, and ResNet101) for training with the FU 30 dataset. Each model is trained up to 100k iterations, and a log is maintained for each 10k iteration for evaluation. The quantitative analysis uses an annotated video with a frame rate of 25 fps to test the Inception, ResNet50, and ResNet101 models. The video is divided into five temporal segments, each of one minute. Each temporal segment has 1500 frames.

All three models record the number of detections of each temporal segment. The detection is then processed through the proposed refinement algorithm to identify the TP, FP, and missed detections. The algorithm is run with window size (W) = 8, 12, and 16. 'W' is the temporal window that reads the consecutive frames to identify the missing and false positive detections. In each temporal window, the algorithm is tested with a threshold (λ) value of 0.3 and 0.4 to determine the number of TP, FP, and missed detection. The F1-score (geometric mean of precision and recall metrics) in each minute of the video is also calculated in each temporal window. The performance of the proposed detection refinement algorithm is analyzed using qualitative analysis by applying it to the results obtained from the Inception, ResNet50, and ResNet101 models. The proposed method consistently increased the model's performance when integrated with any detector. This mechanism helps marine science experts in the assessment of the abundance of this species. The proposed mechanism is also helpful in counting the unique burrows, as discussed in the next section.

The third part of the thesis is about the tracking and counting of *Nephrops* burrows. In this work, the data from FU 30 is used and trained by YOLOv3 (You Only Look Once), a real-time object detection algorithm to identify the *Nephrops* burrows. YOLOv3 is a single-stage and extremely fast and accurate model. The annotated data set is converted to YOLO Darknet TXT annotation format to train the models. The work is implemented in TensorFlow and OpenCV deep learning libraries. The model is trained with an FU 30 station. The trained model is tested with the videos from the FU 30 station to detect the burrows. The major challenge is to track and count the unique burrows in consecutive frames.

The literature used many tracking algorithms that can be used underwater. Some of them used OpenCV KCF tracker, while others used Optical flow or Kalman filters to track the objects in the underwater environment. The *Nephrops* burrows have different characteristics than other marine objects with fixed dimensions, and their features can be easily extracted. Also, the size of each burrow and the angle of the same burrow vary in the consecutive frames due to the variation in camera angle. The other factor that makes the tracking challenging is the camera movement. The burrows are fixed objects. They are not moving while the camera is moving in the forward direction, making the tracking of burrows difficult. Tracking and counting *Nephrops* burrows are proposed using the spatial-temporal values of each burrow. The proposed spatial-temporal technique tracks each burrow based on its spatial and temporal values and counts the unique burrows. The unique burrows are counted using the intersection values of detected burrows in consecutive frames. The experiments were performed on different temporal sets of a sample video from the FU 30 station. The work also implemented the state-of-the-art OpenCV object tracker algorithms to track the *Nephrops* burrows and discover false positive results. The work compared the proposed algorithm results with these algorithms and found that the proposed solution provides much more accurate results than already available tracking algorithms. The results are presented quantitatively and qualitatively. The results show the counting accuracy up to 90%. The developed system is a complete framework that annotates, detects, corrects, tracks and counts the *Nephrops* burrows. The system is currently trained with FU 30, but the work is tested on a different dataset from Iceland, Scotland, Italy, and Aberdeen, UK, and got some promising initial results.

This page intentionally left blank.

Resumen de la tesis doctoral en español

Título de la tesis doctoral en español

Detección y clasificación de madrigueras de *Nephrops norvegicus* a partir de vídeos submarinos mediante técnicas de "Deep Learning"

RESUMEN

Presentación

El ecosistema terrestre está formado principalmente por océanos, que producen el 50% del oxígeno atmosférico y que acumula el 97% del agua. Además, es una fuente importante de nuestra alimentación diaria ya que aporta un 15% de proteínas en forma de animales marinos. Sin embargo, hay más estudios sobre los ecosistemas terrestres que sobre los ecosistemas marinos. Porque es más difícil estudiar el ecosistema marino, especialmente sus regiones más profundas. El estudio del ecosistema marino es muy costoso y requiere equipamientos y experiencia humana especiales para investigar en ese ámbito tan particular.

Por otro lado, la investigación en el análisis de imágenes submarinas ha ganado popularidad en muchas aplicaciones de las ciencias marinas. Los Oceanógrafos suelen decir a los ingenieros, cuando discuten sobre tecnologías: "lo que queremos saber es qué pasa en el fondo del mar". Y es muy complicado saber lo que pasa en el fondo del mar. A lo largo de los años de la Oceanografía se han desarrollado decenas de sensores que ayudan a contestar a la pregunta. En esta ocasión, sin embargo, la expresión popular de "una imagen vale más que cien palabras" parece que no está del todo desencaminada. La adquisición de imágenes está convirtiéndose en una herramienta de gran utilidad a los Oceanógrafos. Pero miles de imágenes complicadas y no siempre de óptima calidad demandan de técnicas que ayuden a su interpretación. Existen varias líneas de investigación en el análisis de imágenes submarinas, por ejemplo, la clasificación y detección de especies acuáticas, el reconocimiento de imágenes del fondo marino, la clasificación de arrecifes de coral y el reconocimiento de flora y fauna. El análisis de imágenes submarinas requiere un conjunto de tareas de procesamiento de imágenes, incluida la detección de objetos submarinos, la clasificación, el reconocimiento visual de contenido y la anotación de imágenes de especies marinas a gran escala. Ciertos desafíos, como la turbidez, las variaciones de color y los cambios de iluminación, dificultan que los entornos submarinos detecten y clasifiquen los objetos automáticamente.

La monitorización de los hábitats de las especies marinas es un desafío para los biólogos y expertos marinos. Las características ambientales, como las variaciones de color basadas en la profundidad y el movimiento de las especies, lo convierten en un desafío. Los científicos marinos utilizaron satélites, sensores a bordo de barcos y cámaras hace varios años para

recopilar imágenes de especies submarinas. En los últimos años, con el avance de la tecnología, los científicos han utilizado vehículos submarinos operados a distancia (ROV, *remotely operated vehicle*), vehículos submarinos autónomos (AUV, *autonomous underwater vehicle*) y estructuras de trineo y bastidor abatible equipadas con cámaras de alta definición para grabar videos e imágenes de especies marinas. Estos vehículos pueden capturar fotos y videos de alta definición. Además de todos estos equipos de calidad, el entorno submarino sigue siendo un reto para los científicos y biólogos marinos. Los dos factores principales que lo dificultan son la libertad del entorno natural y las variaciones del contenido visual, que pueden surgir de la iluminación variable, las escalas, las vistas y las deformaciones no rígidas.

Nephrops norvegicus - la cigala – y sus madrigueras

Hay miles de especies en el océano de todo el mundo. Una de las especies comerciales más importantes de Europa es la cigala, *Nephrops norvegicus* (en lo sucesivo, *Nephrops*). La *Nephrops* es una de las principales pesquerías comerciales de crustáceos en Europa, donde en 2.018 el total admisible de capturas (TAC) se fijó en 32.705 toneladas para las zonas del Consejo Internacional para la Exploración del Mar (CIEM).

La especie *Nephrops* se distribuye desde los 10 m hasta los 800 m de profundidad en las aguas del Atlántico NE y del mar Mediterráneo, donde los sedimentos son adecuados para construir sus madrigueras. Las *Nephrops* pasan la mayor parte de su tiempo dentro de sus madrigueras, y su comportamiento está influenciado por la época del año, la intensidad de la luz y la fuerza de las mareas. Estas madrigueras pueden detectarse mediante técnicas de análisis de imagen si se ha usado una configuración óptima de la iluminación durante las grabaciones de vídeo del fondo marino. Las madrigueras en sí mismas pueden identificarse fácilmente a partir de las características de la superficie del fondo una vez que se ha adquirido la capacitación especializada. Esta especie excava y habita en sistemas de madrigueras principalmente en sedimentos fangosos del fondo marino, con más de un 40% de limo y arcilla.

Un sistema de madriguera *Nephrops* generalmente puede tener una o varias aberturas y diferentes túneles. Se asume que un único individuo ocupa un sistema de madriguera. Las madrigueras *Nephrops* muestran características distintivas que es lo que permite identificarlas y distinguirlas de otras madrigueras o simplemente accidentes del fondo. Las características de la madriguera se resumen de la siguiente manera:

1. Al menos una abertura de madriguera tiene forma de media luna.
2. A menudo hay pruebas de sedimentos expulsados, normalmente en un amplio "abanico" en forma de delta en la abertura del túnel, y los arañazos y las huellas son evidentes con frecuencia.
3. El centro de todas las aberturas de la madriguera tiene una estructura elevada.
4. Las *Nefrops* pueden estar presentes (ya sea dentro o fuera de la madriguera)

El sistema de madrigueras de *Nephrops* comprende una o más madrigueras de las características antes mencionadas. La presencia de más de una madriguera cercana no significa la presencia de más de una *Nephrops*.

Las *Nefrops* pasan la mayor parte de su tiempo dentro de las madrigueras, y su comportamiento está condicionado por varios factores: la época del año, la intensidad de la luz o la fuerza de las mareas. Por esta razón, los índices de abundancia obtenidos de las capturas comerciales o de las prospecciones tradicionales de arrastre de fondo se consideran poco representativos de la población real de cigalas y no se considera una técnica de estimación apropiada.

Estimación de la población

La abundancia de las poblaciones de *Nephrops* es actualmente monitorizada por estudios de televisión submarina (UWTV, *underwater television*) en una mayoría de países europeos. La metodología utilizada en los estudios UWTV se desarrolló en Escocia en la década de 1990 y se basa en la identificación y cuantificación de los sistemas de madrigueras en las regiones donde se conoce que hay una distribución de *Nephrops*. La abundancia de *Nephrops* de los estudios de UWTV es la base de la evaluación del stock de sus poblaciones y del asesoramiento a las autoridades competentes en la gestión de su explotación.

Los vídeos se graban utilizando un sistema de cámara montado en el trineo con un ángulo hacia la parte inferior que oscila entre 37° y 60° (distintos países usan distintos ángulos, según su criterio y experiencia previa, así como de los medios ópticos de que dispongan). Los vídeos grabados se guardan en soporte DVD. Los vídeos son posteriormente visualizados por los expertos marinos con cualificación al respecto, siguiendo el protocolo establecido por el CIEM.

El Consejo Internacional para la Exploración del Mar, CIEM (en inglés ICES, *International Council for the Exploration of the Sea*), es una organización internacional de 20 países miembros. La acción y razón de ser del CIEM son las ciencias marinas. Cuenta con más de 6.000 científicos de más de 700 institutos marinos diferentes de los países miembros. Cada año, más de 2.500 científicos colaboran en actividades del CIEM. El CIEM tiene un Grupo de Trabajo sobre Censos de Nefrops (WGNEPS), antes llamado Grupo de Estudio sobre Censos de *Nefrops* (SGNEPS), que es el grupo de expertos que coordinan los estudios sobre poblaciones de *Nephrops* mediante UWTV y redes de arrastre y que están especializados en la realización de encuestas sobre la población de cigalas usando esas dos técnicas. Uno de los objetivos del grupo es la coordinación internacional de las encuestas de poblaciones de *Nephrops* en el Atlántico Norte. WGNEPS se ha centrado en la planificación, los protocolos, el control de calidad, el diseño y los problemas de desarrollo de las encuestas, con objeto de mejorar las estimaciones y, al homogenizar las técnicas de realización de encuestas, los resultados obtenidos por científicos de distintos institutos puedan ser comparados.

Los estudios de UWTV y de arrastre proporcionan estimaciones de la población de cigalas basadas en unidades funcionales (FU) en las zonas del CIEM y de forma preliminar y exploratoria en algunas subzonas geográficas (GSA) del Mediterráneo.

En la actualidad se utilizan dos enfoques principales de diseño de encuestas UWTV: diseño de cuadrícula (fija o aleatoria) o diseño aleatorio estratificado, en el que en algunas encuestas hay solapamiento entre estaciones para garantizar una mejor cobertura espacial. Ambos enfoques permiten la aplicación de modelos geoestadísticos para estimar los niveles de abundancia y precisión. Por lo general, en el primer enfoque, la cuadrícula se extiende de forma adaptativa hasta que se establecen límites por parte de los expertos, basados en experiencia previa. El enfoque aleatorio estratificado utiliza datos a priori sobre sedimentos y datos de estratificación vertical marina (VMS, *vertical marine stratification*) que se integran en la encuesta para definir estratos con densidades más similares. La definición de los límites de la encuesta y su estratificación es esencial para cumplir con el nivel de precisión requerido.

Los estudios UWTV fueron pioneros en Escocia a principios de los años 90 para monitorear la abundancia de las poblaciones de *Nephrops*. La estimación de las poblaciones de cigala utilizando este método implica identificar y cuantificar la densidad de madrigueras en el área conocida de distribución de *Nephrops* que se puede utilizar como índice de abundancia de la población. La abundancia de *Nephrops* de los estudios de UWTV es la base de la evaluación y el asesoramiento para la gestión de estas poblaciones. Los estudios UWTV deben llevarse a cabo anualmente cuando las condiciones de claridad del agua sean óptimas y sea probable que las condiciones climáticas sean tranquilas. Las encuestas no están restringidas a una hora particular del día y pueden llevarse a cabo operaciones durante las 24 horas del día.

En la técnica estandarizada por WGNEPS se utiliza un trineo que se diseñó en Escocia para la adquisición de datos. El trineo debe ser lo suficientemente robusto como para asegurar todos los instrumentos de medida y adquisición de datos, pero también debe ser lo suficientemente flexible como para asegurar su equilibrio. El sistema de iluminación adecuado en el trineo se distribuye de forma que se garantice una iluminación uniforme en todo el campo de visión. Dos luces láser están fijadas verticalmente en el trineo, mostrando el campo de visión.

Las Unidades Funcionales (FU) evalúan y gestionan las poblaciones de *Nephrops*. Hay una valoración (o encuesta) específica para cada FU. En 2019, 19 encuestas abarcaron las 25 FU del CIEM y una GSA, la del mar Adriático. Estas encuestas se llevaron a cabo utilizando equipos estandarizados y protocolos acordados bajo el mandato de WGNEPS. Este estudio considera datos de los territorios de *Nephrops* del Golfo de Cádiz (FU 30) y los Smalls (FU 22) para detectar las madrigueras de *Nephrops* utilizando los datos de imagen recogidos de diferentes estaciones en cada FU utilizando nuestra metodología propuesta. Los Smalls es una zona de capturas en el mar Céltico, al sur de Irlanda y al suroeste del canal de san Jorge, que es de enorme importancia económica para la flota pesquera de esa región de Irlanda.

El entorno submarino es difícil de analizar, ya que presenta desafíos formidables para las tecnologías de visión artificial y aprendizaje automático. La clasificación y detección de imágenes en imágenes submarinas difiere significativamente de la de otros tipos de imágenes. Además, la recopilación de datos en un entorno acuático conlleva enormes dificultades técnicas. Una de las razones de esto es la luz o, mejor dicho, la ausencia de luz, porque cuando la luz se propaga a través del agua, la absorción, refracción, dispersión y aberración deforman las imágenes. Además, la dispersión y la iluminación no uniforme hacen que el entorno sea más difícil para la calidad de las imágenes recopiladas. La mala visibilidad también es un problema común en el entorno submarino. La escasa visibilidad se debe al flujo y reflujo de las mareas, lo que hace que las partículas finas de lodo se suspendan en la columna de agua. La corriente oceánica es otro factor que provoca frecuentes cambios de luminosidad. Las características visuales, como las condiciones de iluminación, los cambios de color y la baja resolución de píxeles, lo hacen un desafío. Algunos elementos ambientales, como las variaciones de color basadas en la profundidad y la turbidez o el movimiento de las especies, dificultan mucho la recopilación de datos en un entorno submarino. Por lo tanto, dos factores principales que lo dificultan son el entorno natural libre y las variaciones del contenido visual, que pueden surgir de la iluminación variable, las escalas, las vistas y las deformaciones no rígidas.

Actualmente, los datos de *Nephrops* se recopilan a través de las encuestas UWTV y son revisados manualmente por los expertos cualificados. Muchos de los datos eran difíciles de procesar debido a las complejas condiciones ambientales. Los sistemas de madrigueras se cuantifican siguiendo el protocolo establecido por el CIEM. El proceso es muy tedioso y lento, lo que es causa de error y, en cierta forma, una pérdida del valioso tiempo de los expertos. A continuación, se describe someramente ese proceso. Los datos de imagen (que se refieren a datos de video o fijos) para cada estación son revisados de forma independiente por al menos dos expertos y los recuentos se registran para cada minuto en los registros de la hoja de registro. Cada fila de la hoja de registro registra el minuto, el número de madrigueras que cuenta el sistema y la marca de tiempo. Los datos de recuento se examinan para comprobar si hay discrepancias inusuales utilizando el coeficiente de correlación de concordancia (CCC) de Lin con un umbral de 0,5. El CCC de Lin mide la capacidad de los contadores para reproducir con precisión los recuentos de los demás en una escala de 0,5 a 1, donde 1 es la concordancia perfecta. Solo las estaciones con un umbral inferior a 0,5 fueron revisadas de nuevo por los expertos.

Con la enorme cantidad de datos recopilados en vídeos e imágenes, anotar y analizar manualmente es laborioso y requiere mucho tiempo de revisión y procesamiento de datos. Todas las estaciones analizan manualmente los sondeos de UWTV para clasificar y contar las Nefrops. Debido a las limitadas capacidades humanas, la evaluación manual de los datos de imagen requiere mucho tiempo por parte de expertos capacitados para procesar los datos con

el fin de que se controle la calidad y estén listos para su uso en la evaluación de las existencias. Debido a estos factores, solo se utiliza una cantidad limitada de datos recopilados para el análisis que, por lo general, no proporciona una visión profunda de un problema. Además, en algunas estaciones, es difícil para el ojo humano clasificar y detectar las madrigueras a partir de una grabación de vídeo.

Con el reciente avance de la inteligencia artificial y la tecnología de visión por ordenador, muchos investigadores emplean herramientas basadas en IA para analizar especies marinas. Algunas personas utilizan mecanismos de extracción de características para contar e identificar las especies, mientras que otras utilizan técnicas avanzadas como las redes neuronales. Las redes neuronales convolucionales (CNN) suponen una revolución en la detección de objetos. Las redes neuronales convolucionales profundas obtienen un gran éxito en las tareas de detección, clasificación y segmentación de objetos. Estas redes están basadas en datos y requieren una gran cantidad de datos etiquetados para el entrenamiento.

Materiales y métodos

Para detectar y clasificar automáticamente el sistema de madrigueras de *Nephrops* para el monitoreo del hábitat, la literatura no puede proporcionar ninguna solución concreta. Esta tesis es un esfuerzo por automatizar el método existente de conteo de madrigueras de *Nephrops*. El trabajo propuso un marco completo para la detección automática de sistemas de madrigueras de *Nephrops*. El trabajo de tesis se divide en tres grandes partes. La primera parte de la tesis muestra el marco para la detección automática de madrigueras de *Nephrops* utilizando redes neuronales profundas. Las redes de aprendizaje profundo se utilizan para detectar y clasificar automáticamente las madrigueras de *Nephrops* que toman datos de vídeo submarino como entrada y aprenden las características jerárquicas de los datos de entrada para detectar las madrigueras en cada fotograma de vídeo de entrada. Los conjuntos de datos FU 30 y FU 22 se recopilaron utilizando diferentes sistemas de adquisición de imágenes (cámara de video Ultra HD 4K y cámara de imágenes fijas HD) de poblaciones de *Nephrops* en 2018-19. En FU 22, se encuestaron 42 estaciones de UWTV en 2018. De un total de 42, se utilizaron siete estaciones para la preparación de los datos. El trineo grabó videos de 10 a 12 minutos a diferentes velocidades de fotogramas que iban desde 15 fps, 12 fps y 10 fps en Ultra HD. Además, las imágenes de alta definición fueron capturadas con la cámara. Las imágenes fueron grabadas con una resolución de 2048 x 1152 píxeles. En FU 30, los vídeos se graban a 25 fotogramas por segundo en buenas condiciones de iluminación. Cada estación de FU 30 tiene secuencias de video grabadas de 10 a 12 minutos. En 2018 se encuestó a un total de 70 emisoras de UWTV. De las 70 estaciones encuestadas, 10 fueron rechazadas debido a las malas condiciones de visibilidad e iluminación. Se seleccionaron siete estaciones para nuestra experimentación con buenas condiciones de iluminación, bajo nivel de ruido y pocos artefactos, mayor contraste y una alta densidad de madrigueras de *Nephrops*.

Propuesta técnica

Los datos recopilados de FU 22 y FU 30 se convierten en tramas. El conjunto de datos recopilados tiene una gran cantidad de fotogramas con iluminación baja y no homogénea y poco contraste. Los fotogramas sin madrigueras o poca visibilidad se descartan durante la fase de anotación, y los fotogramas consecutivos con información similar se descartan. El siguiente paso es anotar los datos recopilados. La anotación de imágenes es una técnica que *Computer Vision* utiliza para crear datos reales de entrenamiento y prueba, ya que los algoritmos de aprendizaje profundo supervisados requieren esta información. Por lo general, cualquier objeto se anota dibujando un cuadro delimitador a su alrededor. Actualmente, los expertos marinos que trabajan con madrigueras de *Nephrops* no están utilizando ninguna herramienta de anotación para anotar madrigueras de *Nephrops*, ya que este es un trabajo que requiere mucho tiempo. Utilizamos la herramienta de anotación de imágenes VOTT de Microsoft para anotar las madrigueras manualmente. Las anotaciones se guardan en Pascal VOC. El archivo de anotación XML guardado contiene el nombre de la imagen, el nombre de la clase (*Nephrops*) y los detalles del cuadro delimitador de cada objeto de interés de la imagen. Las imágenes anotadas son validadas por expertos en ciencias marinas del Golfo de Cádiz, España e Irlanda. La validación de la anotación es esencial para obtener información veraz de alta calidad. Este proceso llevó mucho tiempo, ya que la confirmación de cada anotación requiere mucho tiempo y es sensible. Después de validar cada anotación, se utiliza un conjunto de datos seleccionado para entrenar y probar los modelos de redes neuronales profundas. Las imágenes anotadas se registran en archivos XML y se convierten en archivos de registro de *TensorFlow* (TF), una secuencia de cadenas binarias que *TensorFlow* requiere para entrenar el modelo. El conjunto de datos se divide en dos subconjuntos: entrenar y probar. El objetivo es aplicar modelos de aprendizaje profundo para detectar, clasificar y contar las madrigueras de *Nephrops* de forma automática. En lugar de entrenar la red desde cero, este trabajo utilizó el aprendizaje por transferencia para ajustar los modelos *Faster R-CNN Inceptionv2*, *MobileNetv2*, *ResNet50*, *ResNet101* y *YOLO v3* en *TensorFlow*. *Inceptionv2* es una de las arquitecturas que tienen un alto grado de precisión. El diseño básico de Inceptionv2 ayuda a reducir la complejidad de CNN. Utilizamos una versión previamente entrenada del modelo de red entrenado en el conjunto de datos COCO.

Inceptionv2 está configurado para detectar y clasificar solo una clase (c = 1), " *Nephrops* ". La arquitectura *CNN MobileNetv2* fue propuesta por Sandler et al. en 2018. Una de las principales razones para elegir la arquitectura *MobileNetv2* fue el conjunto de datos de entrenamiento relativamente pequeño de FU 30. Esta arquitectura optimiza el consumo de memoria y la velocidad de ejecución con errores menores. La arquitectura *MobileNetv2* tiene convolución separable en profundidad en lugar de convolución convencional. Esta arquitectura tiene inicialmente una capa de convolución con 32 filtros, seguida de 17 capas de cuello de botella residuales. *ResNet50* es una variante del modelo *ResNet*. *ResNet50* tiene 48 capas

convolucionales, un grupo máximo y una capa de grupo promedio, por lo que es una red convolucional de 50 capas de profundidad. *ResNet101* es una red neuronal convolucional densa que tiene 101 capas. *YOLOv3* utiliza *darknet* para entrenar el modelo. Originalmente, la *darknet* tenía 53 capas. En *YOLOv3*, se agregan otras 53 capas a la *darknet* para su detección, lo que hace 106 capas de arquitectura totalmente convolucional.

El entrenamiento, la validación y las pruebas del modelo se llevan a cabo en una máquina Linux con una GPU (*graphics processing unit*) *NVIDIA TitanXP*. Se crean varias combinaciones para el entrenamiento de modelos, es decir, se entrenan modelos separados para conjuntos de datos FU 22 y FU 30, se entrena un modelo combinando ambos conjuntos de datos (lo que se denomina modelo híbrido) y se entrenan y prueban con diferentes conjuntos de datos. Los modelos se entrenaron utilizando una muestra aleatoria de aproximadamente el 70-75% del conjunto de datos anotado. El resto se utiliza para las pruebas. Los puntos de control de giro durante el entrenamiento se registran después de cada 10k iteraciones, y el mAP50 está en el conjunto de datos de validación. El modelo se evalúa mediante mAP, curva de precisión y recuperación, e inspección visual de las imágenes con detecciones automáticas. El modelo se prueba para evaluar el rendimiento. Los modelos se prueban con imágenes no vistas de los conjuntos de datos FU 30 y FU 22 y evalúan el rendimiento del modelo. Los experimentos muestran la evaluación cuantitativa y cualitativa de estas redes. Se realizan treinta combinaciones diferentes de conjuntos de experimentos con diferentes modelos de entrenamiento. Cada conjunto se repite siete veces. Así, se llevaron a cabo 210 experimentos. Los modelos utilizaron 200 imágenes del conjunto de datos FU 30 para entrenar el modelo, mientras que se utilizaron 48 imágenes para probar los modelos.

Del mismo modo, estos modelos utilizaron 618 imágenes del conjunto de datos FU 22 para el entrenamiento y 359 para las pruebas. Los modelos entrenados con el conjunto de datos FU 30 y probados con el conjunto de datos FU 22 utilizaron 200 imágenes para entrenar el modelo y 150 imágenes para las pruebas. Los modelos que utilizaron el conjunto de datos FU 22 para el entrenamiento y el conjunto de datos FU 30 para las pruebas utilizaron 618 imágenes para entrenar el modelo, mientras que 200 imágenes fueron para las pruebas. Por último, los modelos que utilizaron el conjunto de datos híbridos para el entrenamiento y las pruebas utilizaron 818 imágenes para entrenar el modelo, mientras que 407 imágenes para probar el modelo. Cuantitativamente, el trabajo evalúa el rendimiento de mAP, una métrica prevalente en la medición de la precisión de los algoritmos de detección de objetos, como *Faster R-CNN*, SSD, etc. Precisión media calcula, como su nombre indica, el valor de precisión media para los valores de recuperación superiores a 0 a 1. La precisión mide la exactitud de la predicción, mientras que la recuperación mide las predicciones positivas. La mAP se calcula con el conjunto de datos de *Nephrops* de las estaciones FU 22 y FU 30 en 100k iteraciones. El trabajo logra una mAP superior al 75%, una indicación positiva para cambiar el paradigma actual de conteo manual de madrigueras de *Nephrops*. El rendimiento de los modelos también se evalúa

mediante curvas de precisión-recuperación. Para la evaluación del modelo, se calculan las anotaciones de verdadero positivo (TP), falso positivo (FP) y falso negativo (FN). Los resultados demuestran que los algoritmos de aprendizaje profundo son una estrategia valiosa y eficaz para ayudar a los expertos en ciencias marinas a evaluar la abundancia de la especie *Nephrops norvegicus* cuando se llevan a cabo estudios de imagen/vídeo submarinos anualmente, siguiendo las recomendaciones del CIEM. Los algoritmos de detección automática podrían reemplazar la tediosa y manual revisión de datos, que hoy en día es el procedimiento estándar, con la promesa de una mayor precisión, cobertura de áreas de muestreo más significativas y una mayor consistencia de evaluación. Este trabajo supone un avance considerable para WGNEPS en el conteo de *Nephrops norvegicus* para la evaluación de poblaciones, donde se demuestra que detecta y cuenta con precisión las madrigueras de cigalas de forma automática.

La segunda parte de esta tesis trata sobre los refinamientos de detección. En la primera parte de la tesis, los modelos lograron buenos resultados en la detección de las madrigueras a partir de los datos de prueba de imagen. Sin embargo, cuando estos modelos entrenados se probaron en un video del Golfo de Cádiz (FU 30), la precisión de los detectores se degradó. El problema se resuelve con muchas detecciones de FP y TP perdidas que afectan negativamente a la precisión de estos modelos. Este trabajo propone un mecanismo de refinamiento de detección basado en información espaciotemporal para mejorar la detección de verdaderos positivos perdidos y suprimir las detecciones de falsos positivos. El trabajo presentado en la literatura utilizó información temporal para rastrear rostros y suprimir las detecciones de falsos positivos. Su enfoque utilizó el seguimiento de bajo nivel para detectar los rostros en imágenes reales. Además, su enfoque no recupera las detecciones perdidas. En nuestro problema, el seguimiento de bajo nivel no se puede aplicar ya que las madrigueras de *Nephrops* están en el suelo, donde las características son muy diferentes de la imagen natural. El trabajo mencionado integra información temporal para rastrear rostros y suprimir los falsos positivos. En el enfoque propuesto ahora, se utiliza información espacial y temporal para suprimir los falsos positivos y recuperar las detecciones perdidas. La obra se divide en dos partes. En primer lugar, el modelo se entrena utilizando los modelos RCNN más rápidos de última generación Inceptionv2, ResNet50 y ResNet101 para detectar madrigueras de *Nephrops*. En la segunda parte del trabajo se aplica el algoritmo de refinamiento de detección basado en el espacio-tiempo propuesto. La información espacial y temporal de cada madriguera detectada se obtiene en una secuencia de vídeo. Esta información se utiliza en varios fotogramas para refinar las detecciones de madrigueras de *Nephrops*. El mecanismo espaciotemporal ayudó a suprimir las madrigueras de FP. Nos permitió encontrar la detección de TP perdida, logrando una mayor precisión y rastreando y contando madrigueras en una secuencia de video. Para abordar los desafíos del detector, el trabajo propuso "Una nueva técnica de refinamiento de detección para la identificación precisa de madrigueras de *Nephrops* en imágenes submarinas" basada en un

análisis espaciotemporal que mejora la mAP de un detector genérico. El mecanismo de refinamiento de detección propuesto identificó las detecciones perdidas, las recuperó y suprimió los falsos positivos. En general, nuestro enfoque tiene las siguientes contribuciones:

- El modelo de filtrado espaciotemporal (STF, *space-time filtering*) extrae la información espacial y temporal de todas las detecciones de los fotogramas consecutivos de un vídeo de entrada suprimiendo los falsos positivos y recuperando las detecciones perdidas. El método propuesto mejorará el rendimiento de los detectores genéricos (como *Inception* y *ResNet*, en nuestro caso).
- Evaluación del rendimiento del marco propuesto en nuestro nuevo conjunto de datos propuesto. A partir de los resultados del experimento, se presenta la efectividad del enfoque propuesto.

Este algoritmo se divide en dos secciones, es decir, supresión de falsos positivos e identificación de detecciones perdidas. Los resultados se evalúan mediante diferentes experimentos realizados utilizando el algoritmo de refinamiento de detección propuesto. El trabajo utiliza tres modelos (*Inception*, *ResNet50* y *ResNet101*) para el entrenamiento con el conjunto de datos FU 30. Cada modelo se entrena hasta 100 mil iteraciones y se mantiene un registro para cada 10 mil iteraciones para su evaluación. El análisis cuantitativo utiliza un vídeo anotado con una velocidad de fotogramas de 25 fps para probar los modelos *Inception*, *ResNet50* y *ResNet101*. El vídeo está dividido en cinco segmentos temporales, cada uno de un minuto. Cada segmento temporal tiene 1500 fotogramas.

Los tres modelos registran el número de detecciones de cada segmento temporal. A continuación, la detección se procesa a través del algoritmo de refinamiento propuesto para identificar las detecciones TP, FP y perdidas. El algoritmo se ejecuta con un tamaño de ventana (W) = 8, 12 y 16. 'W' es la ventana temporal que lee los fotogramas consecutivos para identificar las detecciones faltantes y de falsos positivos. En cada ventana temporal, el algoritmo se prueba con un valor de umbral (λ) de 0,3 y 0,4 para determinar el número de TP, FP y detección perdida. La puntuación F1 (media geométrica de las métricas de precisión y recuerdo) en cada minuto del vídeo también se calcula en cada ventana temporal. El rendimiento del algoritmo de refinamiento de detección propuesto se analiza mediante análisis cualitativo aplicándolo a los resultados obtenidos de los modelos *Inception*, *ResNet50* y *ResNet101*. El método propuesto aumentó constantemente el rendimiento del modelo cuando se integró con cualquier detector. Este mecanismo ayuda a los expertos en ciencias marinas en la evaluación de la abundancia de esta especie. El mecanismo propuesto también es útil para contar las madrigueras únicas, como se discute en la siguiente sección.

La tercera parte de la tesis trata sobre el seguimiento y conteo de madrigueras de *Nephrops*. En este trabajo, los datos de FU 30 son utilizados y entrenados por *YOLOv3* (*You Only Look Once*), un algoritmo de detección de objetos en tiempo real para identificar las madrigueras de *Nephrops*. *YOLOv3* es un modelo de una sola etapa y extremadamente rápido y preciso. El

conjunto de datos anotados se convierte al formato de anotación TXT de *YOLO Darknet* para entrenar los modelos. El trabajo se implementa en las bibliotecas de aprendizaje profundo *TensorFlow* y *OpenCV*. El modelo se entrena con una estación FU 30. El modelo entrenado se prueba con los videos de la estación FU 30 para detectar las madrigueras. El mayor reto es rastrear y contar las madrigueras únicas en fotogramas consecutivos.

La literatura utiliza muchos algoritmos de seguimiento que se pueden utilizar bajo el agua. Algunos de ellos utilizaron el rastreador *OpenCV KCF*, mientras que otros utilizaron filtros de flujo óptico o Kalman para rastrear los objetos en el entorno submarino. Las madrigueras de *Nephrops* tienen características diferentes a las de otros objetos marinos con dimensiones fijas, y sus características se pueden extraer fácilmente. Además, el tamaño de cada madriguera y el ángulo de la misma madriguera varían en los fotogramas consecutivos debido a la variación en el ángulo de la cámara. El otro factor que hace que el seguimiento sea un desafío es el movimiento de la cámara. Las madrigueras son objetos fijos. No se mueven mientras la cámara se mueve hacia adelante, lo que dificulta el seguimiento de las madrigueras. El seguimiento y conteo de las madrigueras de *Nephrops* se propone utilizando los valores espaciotemporales de cada madriguera. La técnica espaciotemporal propuesta rastrea cada madriguera en función de sus valores espaciales y temporales y cuenta las madrigueras únicas. Las madrigueras únicas se cuentan utilizando los valores de intersección de las madrigueras detectadas en fotogramas consecutivos. Los experimentos se realizaron en diferentes conjuntos temporales de un video de muestra de la estación FU 30. El trabajo también implementó los algoritmos de seguimiento de objetos *OpenCV* de última generación para rastrear las madrigueras de *Nephrops* y descubrir resultados falsos positivos. El trabajo comparó los resultados del algoritmo propuesto con estos algoritmos y encontró que la solución propuesta proporciona resultados mucho más precisos que los algoritmos de seguimiento ya disponibles. Los resultados se presentan cuantitativa y cualitativamente. Los resultados muestran una precisión de conteo de hasta el 90%. El sistema desarrollado es un marco completo que anota, detecta, corrige, rastrea y cuenta las madrigueras de las *Nephrops*. El sistema está actualmente entrenado con FU 30, pero el trabajo se prueba en un conjunto de datos diferente de Islandia, Escocia, Italia y Aberdeen, Reino Unido, y obtuvo algunos resultados iniciales prometedores.

Conclusiones

A modo de resumen del resumen, nuestra contribución radica en comparar la detección de madrigueras de *Nephrops* con diferentes modelos, el análisis profundo y la aplicación del algoritmo de refinamiento de la detección mediante el cálculo de la precisión, el recuerdo y la puntuación F1, y la comparación del algoritmo de seguimiento propuesto también se compara con los algoritmos de seguimiento de *OpenCV*. Todos estos experimentos se realizaron para diferentes combinaciones de conjuntos de datos y diferentes niveles de parámetros. Los resultados muestran que nuestro enfoque tiene mejores resultados en cuanto a detección, refinamiento, seguimiento y recuento de madrigueras de *Nephrops*.

This page intentionally left blank.

List of Figures

This page intentionally left blank.

List of Tables

This page intentionally left blank.

List of Abbreviations

AP	Average Precision
AUVs	Autonomous Underwater Vehicles
BPNN	Background Propagation Neural Network
CCC	Concordance Correlation Coefficient
CNN	Convolution Neural Network
DOG	Distance Over Ground
FOV	Field Of View
FP	False Positive
FU	Functional Unit
GPS	Global Positioning System
GSA	Geographical Sub Areas
GT	Ground Truths
ICES	International Council for the Exploration of the Sea
IEO	Instituto Español de Oceanografía
IoU	Intersection Over Union
KCF	Kernelized Correlation Filters
KLT	Kanade-Lucas-Tomasi
LSTM	Long Short-Term Memory
mAP	Mean Average Precision
MIL	Multiple Instance Learning
R-CNN	Region-Based Convolution Network
ROV	Remotely Operated Vehicle
RPN	Region Proposal Network
SGNEPS	Study Group on *Nephrops* Surveys
STF	Spatial–Temporal Filtering
SORT	Simple Online Real-time Tracking
TAC	Total Allowable Catch
TP	True Positive
UHD	Ultra High Definition
USBL	Ultrashort Baseline
UWTV	UnderWater TeleVision
WGNEPS	Working Group on *Nephrops* Surveys
YOLO	You Only Look Once

This page intentionally left blank.

Chapter 1: Introduction

The earth's ecosystem mainly comprises oceans, producing 50% oxygen and 97% water. Also, it is a significant source of our daily food as it provides 15% of proteins in the form of marine animals. There are many more studies on terrestrial ecosystems than on marine ecosystems because it is more challenging to study the marine ecosystem, especially in the deeper areas. Monitoring the habitats of marine species is difficult for biologists and marine experts. Environmental features such as depth-based color variations and the turbidity or movement of species make it a challenge [1]. Marine scientists used satellites, shipborne, and camera sensors several years ago to collect underwater species images. In recent years, with the advancement of technology, scientists have used underwater Remotely Operated Vehicles (ROVs), Autonomous Underwater Vehicles (AUVs), sledge and drop frame structures equipped with high-definition cameras to record the videos and images of marine species. These vehicles can capture high-definition photos and videos. Besides all this quality equipment, the underwater environment is still challenging for scientists and marine biologists. The two main factors which make it difficult are the free natural environment and variations of the visual content, which may arise from variable illumination, scales, views, and non-rigid deformations [2].

Thousands of underwater species are essential for marine scientists to monitor. One of them is *Nephrops norvegicus* (a Norway lobster, *Nephrops* from now on), an important European commercial species. Functional Units (FU) assess and manage *Nephrops* populations, where there is a specific survey for each FU. A survey is conducted every year across European countries to monitor the habitat of *Nephrops norvegicus*. This species lives in sandy-muddy sediments, creating burrows in the Atlantic NE waters and the Mediterranean Sea [3]. Special equipment is used in the survey. The abundance of *Nephrops* populations is currently monitored by underwater television (UWTV) surveys on many European grounds. The survey data is stored on disks in the form of high-definition images and videos. The data is analyzed manually using the TV survey to classify and count the *Nephrop* burrows. *Nephrops* spend most of their time inside the burrows, and their emergence behavior is influenced by several factors: time of year, light intensity, or tidal strength.

For this reason, abundance indices obtained from the commercial catch or the traditional bottom trawl surveys are considered poorly representative of the *Nephrop* population and are not considered appropriate. Currently, the *Nephrop* data are collected through the UWTV surveys and are reviewed manually by trained experts. Many of the data were

difficult to process due to complex environmental conditions.

AI is an emerging field that solves many detection problems, including classifying underwater species and detections. However, the literature cannot provide any concrete solution to detect and classify the *Nephrops* burrows system for habitat monitoring. One of the main reasons is the unavailability of *Nephrops* survey data. The complexity of data is also one of the reasons. This thesis is an effort to automate the existing method of *Nephrops* counting.

1.1. Importance

The *Nephrops* supports one of the most essential fisheries in Europe, with landings of almost 60,000 t [4] (1 t=1 Mg) and a first sale income of approximately 300 M€ annually [5]. Most *Nephrops norvegicus* are counted in the Northeast Atlantic fisheries, with the United Kingdom contributing more than 56%. The Mediterranean Sea contributes around 7% of total landings, with a comprehensive first-sale income of 86 million € annually [6].

1.2. Motivation

WGNEPS conducts the survey yearly through special equipment. The Spanish Institute of Oceanography in Cadiz and the Marine Institute Ireland have collected the data through these surveys. These stations are represented as FU 30 and FU 22 functional units. *Nephrops* data are collected, and trained experts review UWTV surveys manually. A ten-to-twelve-minute video was made on each point of interest, and the whole survey has more than 20-30 points of interest yearly. Many of the data were difficult to process due to complex environmental conditions. Each station's image data (which refers to video or stills data) is reviewed independently by at least two experts, and the counts are recorded for each minute onto log sheet records. Each row of the log sheet records the minute, the number of burrows system count, and the time stamp. Count data are screened to check for any unusual discrepancies using Lin's Concordance Correlation Coefficient (CCC) with a threshold of 0.5. Lin's CCC [24] measures the ability of counters to precisely reproduce each other's counts on a scale of 0.5 to 1, where 1 is perfect concordance. Only stations with a threshold lower than 0.5 were reviewed again by the experts. Figure. 1.1 shows the current methodology used for the counting of *Nephrops* burrows. This exercise costs a lot of time, human and money resources. No system is available to help them solve their current problem. Spanish Institute of Oceanography and Marine Institute Ireland is willing to collaborate in this project and provide all the dataset and technical support throughout the research project.

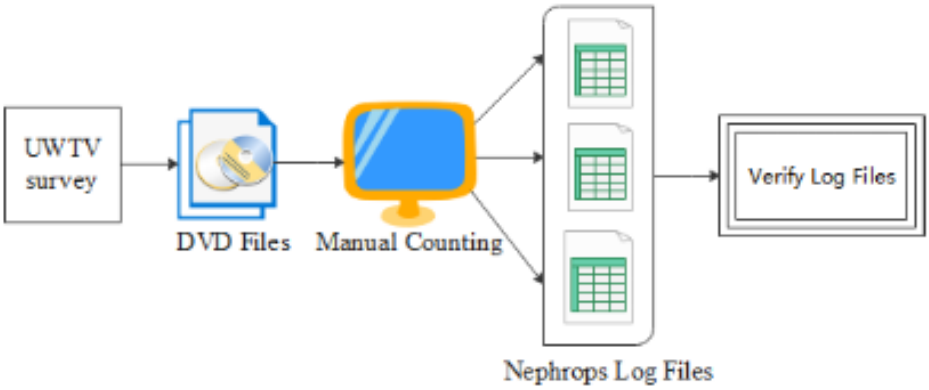


Figure 1.1: Current methodology for Nephrops norvegicus burrows count

1.3. Problem Statement

From the engineering point of view, some of the significant research problems in identifying and classifying the burrows of *Nephrops* are:

a) To understand the dataset and its limitations.
b) To prepare the dataset for model training and testing.
c) To determine the mechanism for pre-processing underwater videos to improve the quality.
d) To mark the ground truth image annotation on the dataset.
e) To explore the state-of-the-art deep learning-based underwater object detection and recognition algorithm.
f) To identify the pattern of burrows and classify between different burrows.
g) To propose a deep learning algorithm to automatically detect and classify the pattern of burrows and provide the assessment of the species.

The outcomes of this research are aimed at benefiting marine biologists, ecologists, and fisheries management organizations.

1.4. Research Objectives

To build a better system for *Nephrops norvegicus* stock assessment, the core objectives of this work are:

a) To annotate and validate the available dataset for model training.
b) To train the Faster RCNN models for automatic detections of *Nephrops norvegicus* burrows
c) To develop a mechanism for rectifying detections.
d) To develop a mechanism to track and count the *Nephrops* burrows automatically.

1.5. Thesis Contribution

This thesis proposes multiple contributions to improving the monitoring methodology and counting the *Nephrops norvegicus*.

a) **Data Preparation and Annotation of *Nephrops norvegicus* burrows:** Currently, the marine experts working with *Nephrops* burrows are not using any annotation tool to annotate *Nephrops* burrows, as this is time-consuming. At first, a semi-automation tool was developed to annotate the burrows. Still, that tool does not help much as it annotates many false positives initially, and rectifying each annotation was very time-consuming. Then, a tool named Microsoft VOTT image annotation tool [26] was adopted to annotate the burrows manually using Pascal VOC format. The saved XML annotation file from this tool contains the image name, class name (*Nephrops*), and bounding box details of each object of interest in the image. The annotated images are validated by marine sciences experts from the Gulf of Cadiz, Spain and Ireland. The validation of annotation is essential to obtain high-quality ground-truth information. This process took a long time as confirming every annotation is time-consuming and sensitive. A dataset is created for different stations with annotations that help the scientists to work in future to build an AI-based system for *Nephrops*.

b) **Automatic Detection of *Nephrops norvegicus* Burrows**: The work proposed deep learning models to automatically detect, classify, and count *Nephrops* burrows. Some current state-of-the-art deep learning models like Inceptionv2, MobileNetv2, ResNet50, ResNet101 and YOLOv3 have been adopted to detect *Nephrops* burrows. The training used transfer learning to train these models. Two different datasets from FU 30 and FU 22 are used for the model training. The proposed work can achieve a mAP higher than 80%, which is a positive indication to change the current paradigm of manual counting of *Nephrops* burrows. This work makes a significant advancement for all the groups working on the *Nephrops norvegicus* counting for stock assessment, where it is shown to automatically detect and accurately count the *Nephrops* burrows.

c) **Detection Refinement Technique for Accurate Identification of *Nephrops norvegicus* Burrows**: The following contribution is to develop a detection refinement technique for deep learning models to improve the quality of results. This work proposes a detection refinement mechanism based on spatial–temporal information to enhance the detection of missed true positives and suppress false positive detections. In the current problem, the low-level tracking cannot be applied as the *Nephrops* burrows are on the ground, where the characteristics are very different from the natural image. In the proposed approach, spatial and temporal information is used to suppress the false positives and recover the missed detections.

The work is divided into two parts. At first, the model is trained using state-of-the-art Faster RCNN models Inceptionv2, ResNet50, and ResNet101 to detect *Nephrops* burrows. The work's second part applies the proposed spatial–temporal-based detection refinement algorithm. Each detected burrow's spatial and temporal information is obtained in a video sequence. This information is used across multiple frames to refine the *Nephrops* burrow detections. The spatial-temporal mechanism helped in suppressing the FP burrows. It allowed us to find the missed TP detection, achieving better accuracy and tracking and counting burrows in a video sequence. To address the detector's challenges, the work proposed "A Novel Detection Refinement Technique for Accurate Identification of *Nephrops norvegicus* Burrows in Underwater Imagery" based on spatial-temporal analysis that enhances the mAP of a generic detector. The results show an improvement in the detections after suppressing the false positives and recovering the missing detections.

d) **Tracking and Counting *Nephrops norvegicus* Burrows:** As a last contribution, a mechanism is proposed for tracking and counting *Nephrops* burrows. The proposed tracking and counting mechanism used the spatial-temporal values of each *Nephrops* burrow. The proposed spatial-temporal technique tracks each burrow based on its spatial and temporal values and counts the unique burrows. The unique burrows are counted using the intersection values of detected burrows in consecutive frames. The proposed methodology is a three-step process that starts from collecting and processing data and detecting and counting burrows. The data is collected from UWTV surveys. This study considers data from the FU 30 station to detect the *Nephrops* burrows. In this work, we trained the model using YOLOv3 (You Only Look Once), a real-time object detection algorithm to identify the *Nephrops* burrows. YOLOv3 is a single-stage and extremely fast and accurate model. We performed experiments on the videos from the FU 30 station. The results show a mAP of *Nephrops* burrow detection of more than 80%. Also, counting TP *Nephrops* burrows using the proposed spatial-temporal algorithms gives accurate results up to 100%.

1.6. Thesis Organization

The dissertation consists of seven chapters and has been organized as follows:

Chapter 2 introduces the marine ecosystem, *Nephrops norvegicus* and its burrows characteristics, ICES, FU, and WGNEPS definitions. Next, the UnderWater TeleVision Survey and the *Nephrops* survey design and timings are discussed. The *Nephrops* study area and observation methodology are discussed in detail in this chapter.

Chapter 3 provides the details about the dataset used in the thesis. The data collection equipment of FU 22 and FU 30 are discussed in detail. The data collection procedure is also mentioned in the chapter. The data of both stations is discussed in terms of their characteristics. The data pre-processing and annotation procedure is discussed in detail. Finally, the chapter concludes with the dataset preparation.

Chapter 4 discusses the deep learning models used in the thesis for the *Nephrops* burrows detections. This chapter introduces the concepts of deep learning, supervised learning, and transfer learning. The complete methodology of *Nephrops norvegicus* burrows detections is presented in this chapter. It also includes the details of each model used in the study, along with their architecture and parameter values. The model training environment and validation process are also presented in the chapter.

Chapter 5 presents a novel technique for detection refinements, tracking and counting *Nephrops* burrows. Section 5.1 introduces the *Nephrops norvegicus* Burrows Detection Refinement concept, followed by the background study. The methodology is presented in section 5.1.2. The detailed algorithm is discussed in section 5.1.4. The *Nephrops norvegicus* Burrows Tracking and Counting mechanism is presented in section 5.2. This chapter also presented the comparative analysis of a few tracking techniques in the underwater environment.

Chapter 6 presented all the experiments performed for *Nephrops* burrows detections, refinements, tracking and counting of burrows. Section 6.1 introduced the experiments about the *Nephrops* burrows detections. The quantitative and qualitative analysis of the results are presented. The performance of models is measured by mAP. The precision and recall curves show the performance of different models on different datasets. Section 6.2 introduced the results of detection refinements. The results are presented quantitatively using precision, recall and F1 score. The results are also shown visually. Section 6.3 discussed the results of tracking and counting *Nephrops* burrows. The results are compared with multiple OpenCV tracking algorithms. The proposed tracking and counting algorithms perform well in all datasets.

Chapter 7 summarizes the dissertation and provides conclusions with the future directions.

1.7. Related Publications

Journals

- Naseer, A.; Nava Baro, E.; Daud Khan, S.; Vila, Y.; Doyle, J. "Automatic detection of *Nephrops norvegicus* burrows from underwater imagery using deep learning," *Computers, Materials & Continua*, vol. 70, no.3, pp. 5321–5344, 2022. https://doi:10.32604/cmc.2022.020886.
- Naseer, A.; Nava Baro, E.; Khan, S.D.; Vila, Y. “A novel detection refinement technique for accurate identification of *Nephrops norvegicus*

burrows in underwater imagery". *Sensors* 2022, *22*, 4441. https://doi.org/10.3390/s22124441.

- Aguzzi, J.; Damianos, C.; Robinson N.J.; Naseer A.; Navarro, J.; Vila Y.; Weetman, A.; Doyle J. (2022). "Advancing fishery-independent stock assessments for the Norway lobster (*Nephrops norvegicus*) with new monitoring technologies". *Frontiers In Marine Science*, 9, 969071 (18p.). Publisher's official version: https://doi.org/10.3389/fmars.2022.969071. Open Access version: https://archimer.ifremer.fr/doc/00797/90879/.

Conference Papers

- A. Naseer, E. N. Baro, S. D. Khan and Y. V. Gordillo, "Automatic Detection of *Nephrops norvegicus* Burrows in Underwater Images Using Deep Learning," *2020 Global Conference on Wireless and Optical Technologies (GCWOT)*, 2020, pp. 1-6. https://doi.org/10.1109/GCWOT49901.2020.9391590.

Technical Reports

- Aguzzi, J.; Aristegui-Ezquibela, M.; Burgos, C.; Doyle, J.; Fifas, S.; Firmin, C.; Jónasson, J.; Jonsson, P.; Lundy, M.; Martinelli, M.; Medvešek, D.; Naseer, A.; O'Connor, J.; Pereira, B.; Silva, C.; Sköld, M.; Vacherot, J-P.; Vila, Y.; Weetman, A.; Wieland, K. (2022). Working Group on *Nephrops* Surveys (WGNEPS; outputs from 2021). International Council for the Exploration of the Sea (ICES). ICES Scientific Report Vol. 4 No. 29 https://doi.org/10.17895/ices.pub.19438472.
- Aristegui-Ezquibela, M.; Aguzzi, J.; Burgos, C.; Doyle, J.; Fallon, N.; Fifas, S.; Jónasson, J.; Jonsson, P.; Lundy, M.; Martinelli, M.; Masmitja, I.; McAllister, G.; Medvešek, D.; Naseer, A.; Reeve, C.; Silva, C.; Simon, J.; Vacherot, J-P.; Vigo-Fernández, M.; Wieland, K. (2021). Working Group on *Nephrops* Surveys (WGNEPS; outputs from 2020). International Council for the Exploration of the Sea (ICES). ICES Scientific Report Vol. 3 No. 36. https://doi.org/10.17895/ices.pub.8041.
- Wieland, K.; Weetman, A.; Aristegui-Ezquibela, M; Aguzzi, J.; Burgos, C.; Chiarini, M.; Cvitanić, R. et al. "Working Group on *Nephrops* Surveys (WGNEPS; outputs from 2019)." (2020).
- Aristegui-Ezquibela, M. et al. "ICES. 2020. Working Group on *Nephrops* Surveys (WGNEPS; outputs from 2019). ICES Scientific Reports". 2020. https:// doi.org /10.17895/ices.pub.5968.

This page intentionally left blank.

Chapter 2: Marine Science

2.1.Introduction

This chapter will introduce the marine sciences and its ecosystem. The *Nephrops norvegicus* and their characteristics are discussed in detail in the chapter, followed by the importance of *Nephrops* species. The *Nephrops* burrow system has specific features discussed in detail in this chapter. This chapter also includes the significant terminologies used by the marine experts during their survey and *Nephrops* counting. This chapter discusses the *Nephrops* study area and observational methodology, followed by the counting procedure and quality control.

2.2.Marine ecosystem

The earth's ecosystem mainly comprises oceans with 97% of water. Marine ecosystems include the open and deep oceans and marine species. The environment has high levels of dissolved salts. The marine ecosystem is one of the primary sources of our daily food. The marine species have different physical and biological characteristics. Coral reefs are an excellent example of an ecosystem associated with other marine life, such as fish and turtles. The oceans cover 70% of our planet, so the marine ecosystem covers most of our earth. There are more studies in terrestrial ecosystems than in marine ecosystems because it is more challenging to study the marine ecosystem, especially in the deeper areas. The environment of the marine ecosystem has specific challenges like color variations, species movement, and turbidity. Monitoring the habitats of marine species is difficult for biologists and marine experts. Marine scientists have been monitoring the environment for decades by collecting underwater species images using satellites, shipborne and cameras. With the advancement of technologies, scientists use several new techniques like ROVs and AUVs to record images and videos of marine ecosystems.

2.3. *Nephrops norvegicus*

The Norway lobster, *Nephrops norvegicus*, is one of the leading commercial crustacean fisheries in Europe, where in 2018, the Total Allowable Catch (TAC) was set at 32,705 tons for International Council for the Exploration of the Sea (ICES) areas 7, 8 and 9 [7]. Figure. 2.1 shows the species of *Nephrops norvegicus*. A *Nephrops* specimen ranges from 2 – 5.5 cm to a maximum length of 24.0 cm. The most common length is about 19.0 cm [8]. Norway norvegicus females undergo ovary ripening mainly in spring, spawn in summer and autumn, and recover in winter. [9]. Figure. 2.2 shows

some of the different sizes of *Nephrops norvegicus* species. This species can be found in sandy-muddy sediments from 90 m to 800 m depth in the Atlantic NE waters and the Mediterranean Sea [3], where the sediment is suitable for constructing their burrows. *Nephrops* spend most of their time inside the burrows, and their emergence behavior is influenced by time of year, light intensity, and tidal strength. These burrows can be detected through optimal lighting set-up during video recordings of the seabed. The burrows can be easily identified from surface features once specialist training has taken [10].

Figure 2.1: Nephrops norvegicus

Figure 2.2: Different sizes of Nephrops norvegicus

Nephrops burrow system

Nephrops norvegicus live inside the burrows that are created on the seabed. They require silt and clay to construct stable burrow systems [11-13]. A burrow system is composed of one or multiple entrances. Usually, a burrow system has a distinct U-shape with at least two openings and a connecting tunnel approximately 20-30 cm below the seabed [14]. Over time, additional entrances and the existing burrow system were also created. The development of the burrow system is influenced by both biotic (e.g., the capability of the animal to both maintain and defend a complex) and abiotic factors (e.g., sediment type, burrow density/available space, hydrographic and benthic morphology, benthic disturbance by trawling) [15].

A *Nephrops* burrow system typically can have single to multiple openings to different tunnels. A unique individual is assumed to occupy a burrow system [16]. Burrows show signature features specific to *Nephrops*, as shown in Figure. 2.3. The characteristics of the burrow system can be summarized as follows:

a) At least one burrow opening is particularly half-moon shape.
b) There is often proof of expelled sediment, typically in a wide delta-like 'fan' at the tunnel opening, and scratches and tracks are frequently evident.
c) The centre of all the burrow openings has a raised structure.
d) *Nephrops* may be present (either in or out of the burrow).

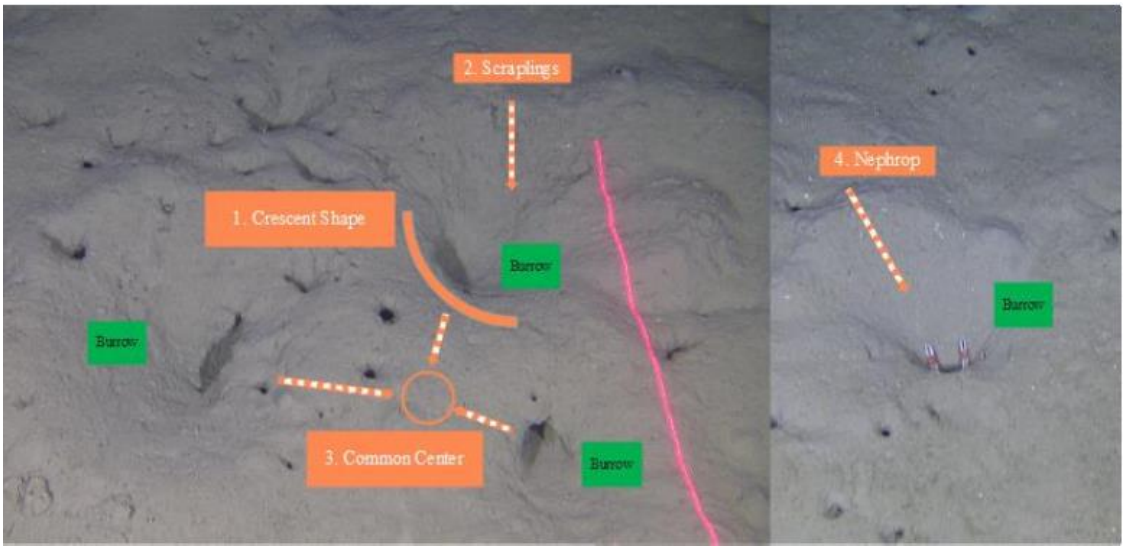


Figure 2.3: Nephrops burrow signature features

2.4. International Council for the Exploration of the Sea, ICES

The International Council for the Exploration of the Sea (ICES) is an intergovernmental marine science organization meeting societal needs for impartial evidence on the state and sustainable use of our seas and oceans. They aim to advance

and share the scientific understanding of marine ecosystems and their services. They use this knowledge to generate state-of-the-art advice for meeting conservation, management, and sustainability goals. The ICES works with 6000 scientists in 700 marine institutes in 20 member countries [17]. ICES is meeting annually to monitor the progress and observations of each member country. ICES is currently working in the Atlantic Ocean, the North Pacific Ocean, the Mediterranean Sea, and the Black Sea.

2.5. Working Group on *Nephrops* Surveys, WGNEPS

The Working Group on *Nephrops* Surveys (WGNEPS), formerly the Study Group on *Nephrops* Surveys (SGNEPS), is the co-ordinating expert group for *Nephrops* UWTV and trawl surveys. The first WGNEPS meeting took place in Barcelona in 2013. The group aims to provide international coordination for *Nephrops* UWTV and trawl surveys in the North Atlantic. WGNEPS has focused on planning, protocols, quality control, design, and survey development issues.

The primary objective of WGNEPS is to generate quality-assured estimates of *Nephrops* absolute abundance within defined areas with a coefficient of variation (CV) or relative standard error of less than 20%. Some of the secondary objectives of the group are:

- To collect multibeam and sediment data to improve the definition of *Nephrops* habitat.
- Collect hydrographic and environmental parameters (e.g., temperature, salinity, turbidity, oxygen, etc.)
- To monitor anthropogenic activity on *Nephrops* grounds (including trawl marks, litter, oil and gas-related impacts, fishing gears, etc.) to comply with the MSFD and OSPAR requirements.
- To integrate benthic and ecosystem monitoring requirements under the MSFD and OSPAR into existing UWTV surveys.
- To monitor various biological parameters and to provide LFD time series data for *Nephrops* if combined with beam trawl or bottom trawl sampling.

The WGNEPS is very important as it standardized all the protocols for *Nephrops* burrows counting and TV surveys. The WGNEPS meeting is held every year to review the *Nephrops* surveys of each station and discuss the possible recommendations for the upcoming year.

2.6. UnderWater TeleVision Survey (UWTV)

Every year, the UnderWater TeleVision (UWTV) and Trawl surveys are conducted all

over Europe to estimate the abundance of *Nephrops norvegicus* species. The surveys are used to provide population estimates for *Nephrops* based on Functional Units (FU) in ICES areas and in a preliminary and exploratory way in some Geographical sub-areas (GSA) in the Mediterranean (Figure 2.4). The UWTV methodology aims to identify and count the *Nephrops norvegicus* burrow systems within a known surface area to obtain absolute abundance estimates, and this involves the use of seabed video footage or high-definition images captured by a camera mounted on a towed sledge [18]. The UWTV methodology aims to identify and count *Nephrops norvegicus* burrow systems within a known surface area to obtain absolute abundance estimates, and this involves the use of seabed video footage or high-definition images captured by a camera mounted on a towed sledge [18]. These surveys are later used in the manual counting of *Nephrops* burrows. UWTV surveys were first carried out on the Fladen ground in 1992 by Marine Scotland Science. Since then, the number of stocks with routine *Nephrops* UWTV surveys has increased over time and in 2017, around 18+ *Nephrops* grounds were surveyed. Estimating Norway lobster populations using this method involves identifying and quantifying burrow density over the known area of *Nephrops* distribution that can be used as an abundance index of the stock [15,19]. *Nephrops* abundance from UWTV surveys is the basis of assessment and advice for managing these stocks [19]. To conduct a standard UWTV survey, a specific protocol is required across different stations. The Spanish Institute of Oceanography Cadiz and Marine Institute Ireland conducted the UWTV survey yearly. This study uses the yearly survey of 2018-19 for the Gulf of Cadiz and Ireland. These stations are represented as FU 30 and FU 22 functional units. Figure 2.4 shows the *Nephrops* UWTV survey coverage in 2019 by the Gulf of Cadiz and Marine Institute Ireland.

2.7. Functional Unit (FU)

There is a data collection framework for *Nephrops norvegicus* all over Europe. In the Northeast Atlantic, stocks of *Nephrops norvegicus* are managed by Total Allowable Catches (TACs) and quotas set at an ICES sub-area level [20]. The sub-areas are aggregations of ICES statistical rectangles, including spatially explicit and uniquely numbered regions referred to as Functional Units (Figure 2.4). The International Council assesses the FU status for the Exploration of the Sea (ICES) through their expert Working Groups [16].

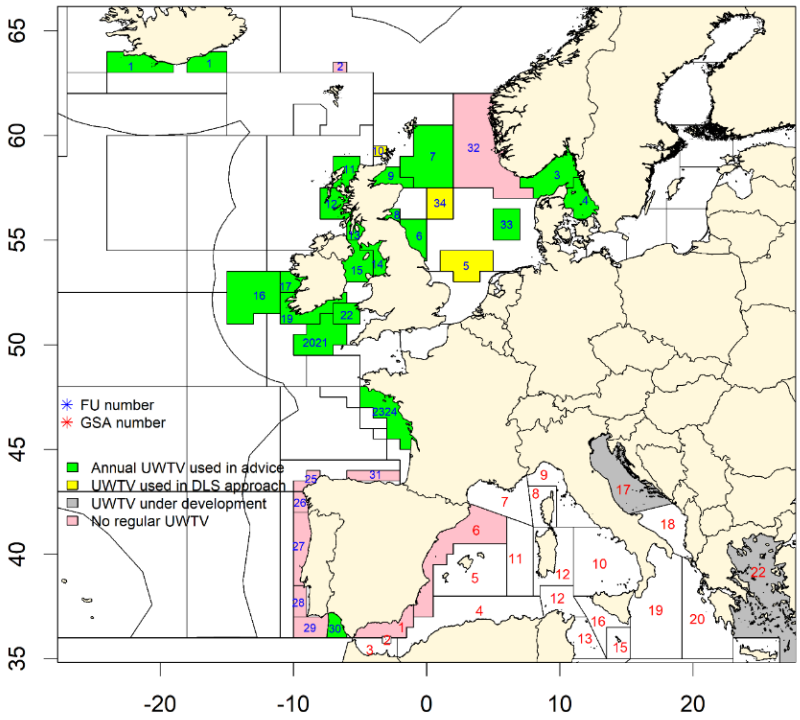


Figure 2.4: Fishery Units (FUs, from ICES) in the Atlantic and Geographical Subareas in the Mediterranean (GSAs, from FAO) as of 2022. FUs and GSAs are used for the fishery assessment of Norway lobster performed by UWTV surveys or by trawling, including areas.

2.8. *Nephrops* Survey Sampling Design

2.8.1. Survey Design

UWTV surveys are focused on suitable habitats for *Nephrops*. Prior knowledge of the *Nephrops* ground is required based on available information such as sediment distribution maps, ground data or local knowledge. The marine experts at different FUs currently use two main UWTV survey design approaches. The first approach is grid (fixed or randomized), while the second is stratified random design. In some UWTV surveys, there is a buffering between stations to ensure better spatial coverage. Table 2.1. shows the statistics for an average number of stations, ground area, density design and CV relative standard error of the UWTV *Nephrops* survey. Both approaches are applied at different stations to estimate the abundance of *Nephrops*. The grid approach is extended adaptively until boundaries are established. The stratified random approach uses a priori data on sediment and or integrated VMS data to define strata with more similar densities.

2.8.2. Survey Timing

Surveys should be carried out annually when water clarity conditions are optimal, and weather conditions are likely to be calm. Surveys are not restricted to a particular time of day, and 24-hour operations can occur. For exploratory purposes or in cases of limited time for dedicated UWTV surveys, the utility of

Table 2. 1: Summary of UWTV survey statistics

Name	FU	Area of Ground (km^2)	Recent number of stations/ 1000 km2	Design	CV-Relative Standard Error (based on 2017)
South off Iceland	FU 1				
Kattegat & Skaggerak	FU 3-4	13104		Random stratified	
Botney Gut & Silver Pit*	FU 5	1000	43	Grid	Na
Farn Deeps	FU 6	2750	39.3	Grid	3.00%
Fladen Ground	FU 7	28153	2.5	Random Stratified	6.40%
Firth of Forth	FU 8	915	52.5	Random Stratified	10.00%
Moray Firth	FU 9	2195	20.5	Random Stratified	12.00%
Noup*	FU 10	400	15	Random Stratified	Na
North Minch	FU 11	2908	13.1	Random	7.30%
South Minch	FU 12	5072	6.9	Random Stratified	10.30%
Clyde (not including Sound of Jura)	FU 13	2081	18.7	Random Stratified	6.10%
Irish Sea East	FU 14	1043	34.5	Grid	10.00%
Irish Sea West	FU 15	5275	23.7	Grid	3.10%
Porcupine Bank	FU 16	7108	8.4	Grid	3%
Aran Grounds	FU 17	926	65.9	Grid	3.10%
SW & South of Ireland	FU 19	1572	22.3	Random Stratified	Na
Labadie	FU 20-21	10014	5.4	Grid	4.40%
Smalls	FU 22	2800	27.1	Grid	5.50%
Bay of Biscay	FU 23-24	11680	14	Grid	Na
Gulf of Cadiz	FU 30	3000	21.3	Randomised isometric Grid	8.70%
Off Horns Rev	FU 33			Random stratified	
Devils Hole	FU 34	1753	10.8	Fixed stations (VMS based)	Na

standard monitoring surveys (potentially during the hours of darkness) or chartered vessels should be considered. To support the expansion of survey coverage to stocks with no or developing UWTV surveys, it is recommended to have technology and methodology transfer through staff exchanges where possible.

2.9. *Nephrops* Study Area

Nephrops are carried out in each FU yearly at different times. Each FU has its geostatistical location in the ocean. In this work, the data from the Smalls (FU 22) and Gulf of Cadiz (FU 30) UWTV surveys are obtained to conduct the experiments to detect *Nephrops* burrows automatically. Figure. 2.5 shows the map of MI-Ireland with stations carried out in 2018 to estimate the burrows. A station is a geostatistical location in the ocean where the *Nephrops* survey is conducted yearly.

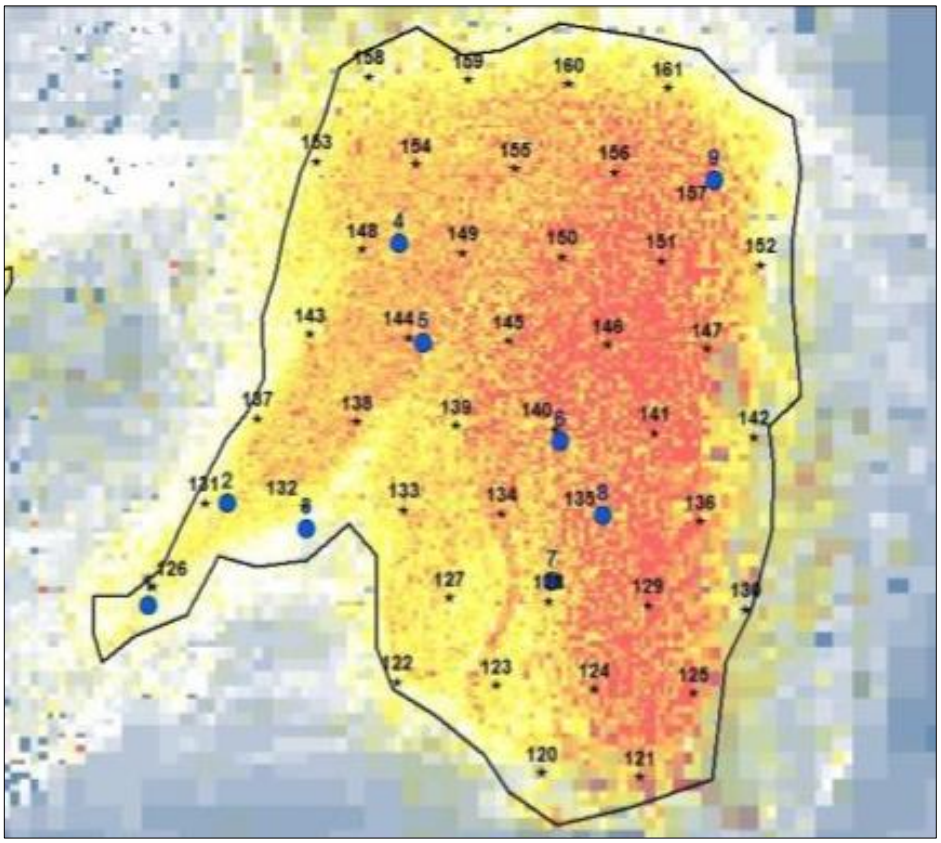


Figure 2.5: Study Area of Nephrops at MI-Ireland in 2018.[23]

Figure. 2.6 shows the Gulf of Cadiz's map with stations carried out in 2018 in ISUNEPCA 0618 and the *Nephrops* burrow density obtained using the manual count.

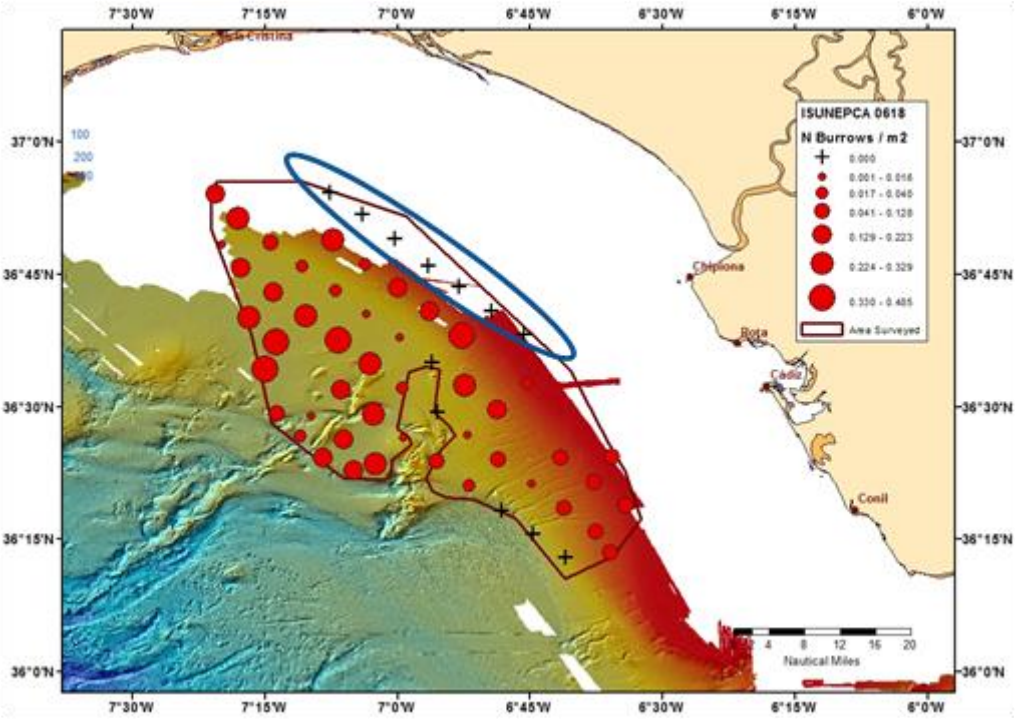


Figure 2.6: Nephrops burrow density at the Gulf of Cadiz in the 2018 survey

Figure 2.7 shows the seabed at different stations of geostatistical locations in the Gulf of Cadiz. Some stations have a high density of *Nephrops*, and some are not included in the survey count due to very little or no density.

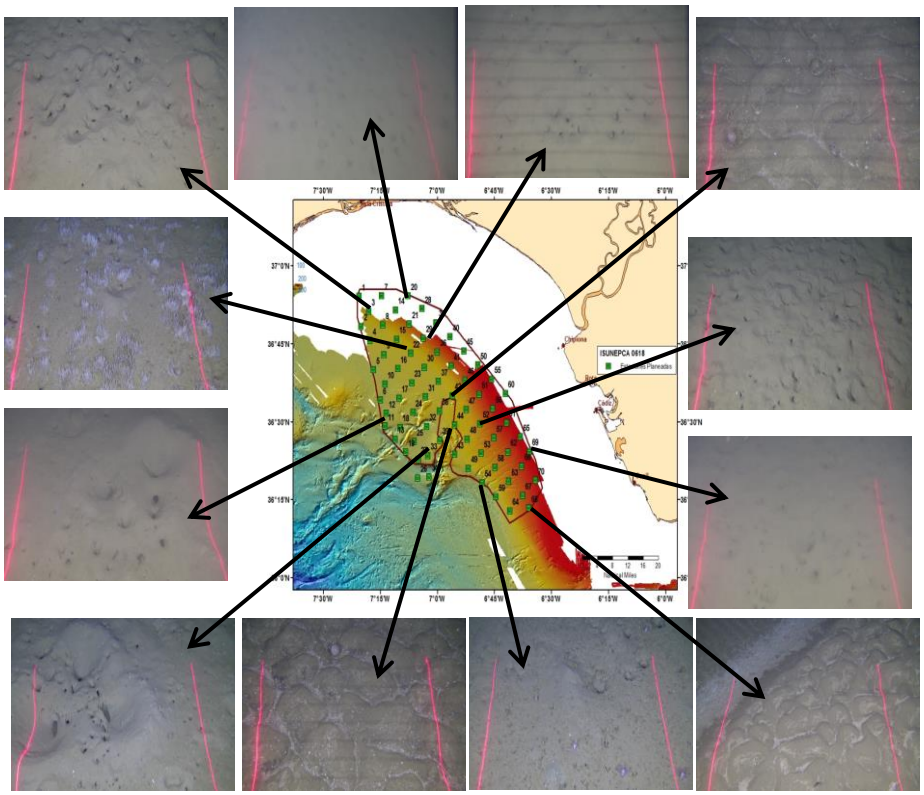


Figure 2.7: Seabed at different stations

2.10. *Nephrops* Observation Methodology

To observe the habitat of *Nephrops*, a survey is designed every year at each FU. At every FU, special underwater equipment with a camera and lights is used for the underwater survey. This equipment varies at every station. The work discusses the general sledge design, camera settings and other equipment. Also, the equipment details of FU 22 and FU 30 stations are presented in our work.

2.10.1. Sledge Design

A sledge is used in the *Nephrops* survey. The sledge's design is based on the Scottish sledge, where the sledge frame should be wide enough (typically around 1.6 m) so that any sediment clouds will not obscure the field of view. To avoid the sledge sinking when deployed on soft sediment, wide sacrificial skis, a lightweight frame, and appropriate flotation should be used [21]. The sledge should be robust enough to secure all instruments, but the system must be flexible sufficient to adjust balance. Figure 2.8 shows the blueprint of the sledge used in the surveys.

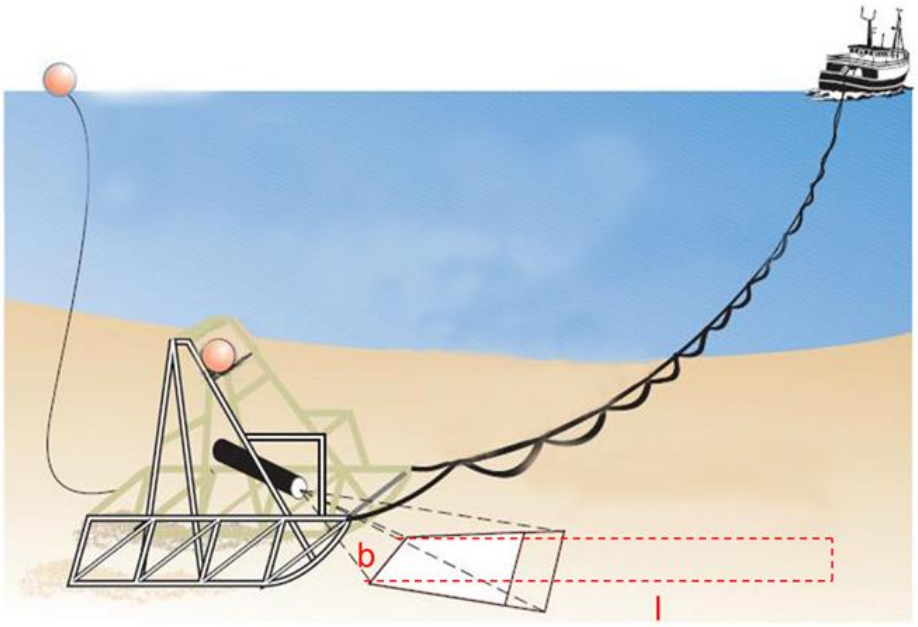


Figure 2.8: Sledge design with floatation and angled camera set-up showing the field of view (b) and distance of TV track (l). K. Mutch, Marine Scotland, Science, Crown Copyright [21]

The sledge used in the survey has mounted cameras that help to record the videos and good-quality images. The camera is mounted at a certain angle with laser lights that show the field view. The camera's field of view should be between 0.7 and 1.0 metres. A field of view of 1 metre plus does not allow sufficient definition to detect and identify burrows. A narrow field of view may be required in high-density grounds, whereas in low-density grounds, a wider area may be

more appropriate. The field of view needs to be calibrated regularly during the survey to ensure a known field of view.

2.10.2. Lighting

The lighting plays an essential role in the survey as it helps capture good-quality images and videos. The light system must evenly distribute light over the entire field of view. The strength of the light power should be adjustable to cope with turbidity and particle reflection. The angle, strength and number of lights should also be able to add a three-dimensional element to the images to assist in correctly identifying subject matter.

2.10.3. Estimation of Vessel and Sledge Distance over Ground

It is essential to accurately calculate the exact distance the sledge travelled over the ground during the video recording. This data should be presented in meters and obtained using various methods such as vessel Global Positioning System (GPS), Ultrashort Baseline (USBL) or an odometer mounted on the sledge.

2.10.4. Timing and Frequency of Sampling

Analysis of tow duration has shown that the mean and variance of burrow count density per tow stabilised after around 5 minutes, provided that the sledge had covered at least half the required distance within the full 10 minutes of the run, approximately 100 m [21-22]. It is recommended that each tow should be at least 10 minutes in duration or longer if poor viewing conditions are experienced. To allow detailed examination of the seabed, vessel speed should be approx. 0.7 knots. This will ensure that a distance of around 200 m is covered at each station. Maintaining constant ground contact during the TV track is essential, which can be facilitated by using the winch.

2.10.5. Recording of Footage, Storage of Footage and Footage Review

It is crucial to synchronise in-time navigation and video files to ensure the link between the video, geographical position and towed distance. This can be done, e.g., by using video overlays and time-related file naming. Many devices, such as DVD recorders, DV tape recorders and hard disc drives in various formats, may record video footage. Video footage should be backed up regularly during the UWTV survey. The type of review monitors used will also depend on the camera system. Analogue TV signal is best reviewed on CRT monitors, whereas high definition can be checked on laptops and flat-screen monitors.

2.10.6. Verification of Video Footage

Each FU in WGNEPS should create a reference set containing ten videos, where each video footage is 5 minutes. The footage is tagged based on visibility (poor,

medium, and good) and *Nephrops* density (low, medium, and high). The reference sets prepared by each functional unit are distributed among all the members of WGNEPS in DVD formats. Agreed consensus counts must be made using independent national experts or international exchanges on these reference sets. The reference set shows the current survey statistics and is updated every year.

2.10.7. The Training Procedure for Counting

Before counting survey footage, all the scientists at every station must be trained using the training material. The training procedure is:

- Each station must provide a one-minute annotated video of their area before counting the survey footage. The video footage covers the range of density visibility and shows the *Nephrops* burrows features [23].
- The footage also shows the identification of burrows that help train the scientists. The expert reviewer should assist all the new training staff members.
- All the reviewers review and validate the counts using Lin's concordance correlation coefficient (CCC – minimum threshold of 0.5) before counting the survey footage [24].

2.11. Counting Procedure and Quality Control

Each institute adopts a standard operating procedure for burrow counting. This includes details of how many minutes are to be counted, warm-up session details, where to count on the screen and removal of minute counts where footage quality deteriorates. Before the counting procedure, a training session helps the reviewers count the burrows independently. The following procedure is adopted for counting and quality control.

a) On completion of the training process, survey counts must be conducted as blind and independent counts; a minimum of two counters should do this. Datasheets should be separate for each counter.

b) Each minute block will count the number of *Nephrops* burrow systems.

c) A warmup count is required for the first minute of each station. Then, a minimum of 7 good visibility minutes should be counted.

d) Suppose counters resume counting after a break of more than 3 hours. In that case, they should be re-familiarised again by reviewing an entire 10-minute run before restarting counting (using random footage from the same area).

e) Only count burrow systems (and partial burrow systems) that pass off the bottom flat edge of the monitor and within the field of view.

Figure 2.9 shows the *Nephrops* review timestamps used during the counting procedure at FU 30.

Timestamp Review Nephrops

Station: **Time:**

ID: Min	Burrow Count	Time Stamp
1		15C 25L-R, 35R, 45C?, 55L glide
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		

Figure 2.9: Nephrops burrows count Timestamp.

This page intentionally left blank.

Chapter 3: Materials and Methods

3.1.Introduction

As discussed in Chapter 2, the Marine experts count the *Nephrops norvegicus* manually. Also, one of the biggest challenges is the unavailability of *Nephrops* dataset. This chapter will provide an overview of the proposed approach and discuss the data collection, preprocessing and preparation in detail. This chapter is divided into two sections. Section 3.2 discusses the overall methodology of this thesis, and section 3.3 discusses the data collection and preparation techniques.

3.2. Proposed Methodology

The proposed methodology of the work is presented in Figure 3.1. The first part of the work is to collect and prepare the dataset. The dataset is collected from the yearly UWTV surveys at different FUs. After performing certain pre-processing, the data is annotated and ready for model training, data collection, and preparation. Section 3.3 discusses in detail the data collection and preparation framework. The second part of the methodology is the detection of *Nephrops* burrows. The *Nephrops* burrows are detected by applying the deep learning technique. The deep learning models are trained and tested on the different datasets. Chapter 4. presented the details about the deep learning techniques, model training and testing mechanism and obtained the results. The third part of the proposed methodology is *Nephrops* burrow detection refinement. This part refines the detections using the proposed detection refinement algorithm. The last part of the proposed methodology is the tracking and counting the *Nephrops* burrows. Multiple tracking algorithms are applied to track the *Nephrops* burrows, but they cannot count them correctly. The work presented a new tracking and counting mechanism based on the spatial values of the *Nephrops* burrows. The details of the algorithm are presented in chapter 5.

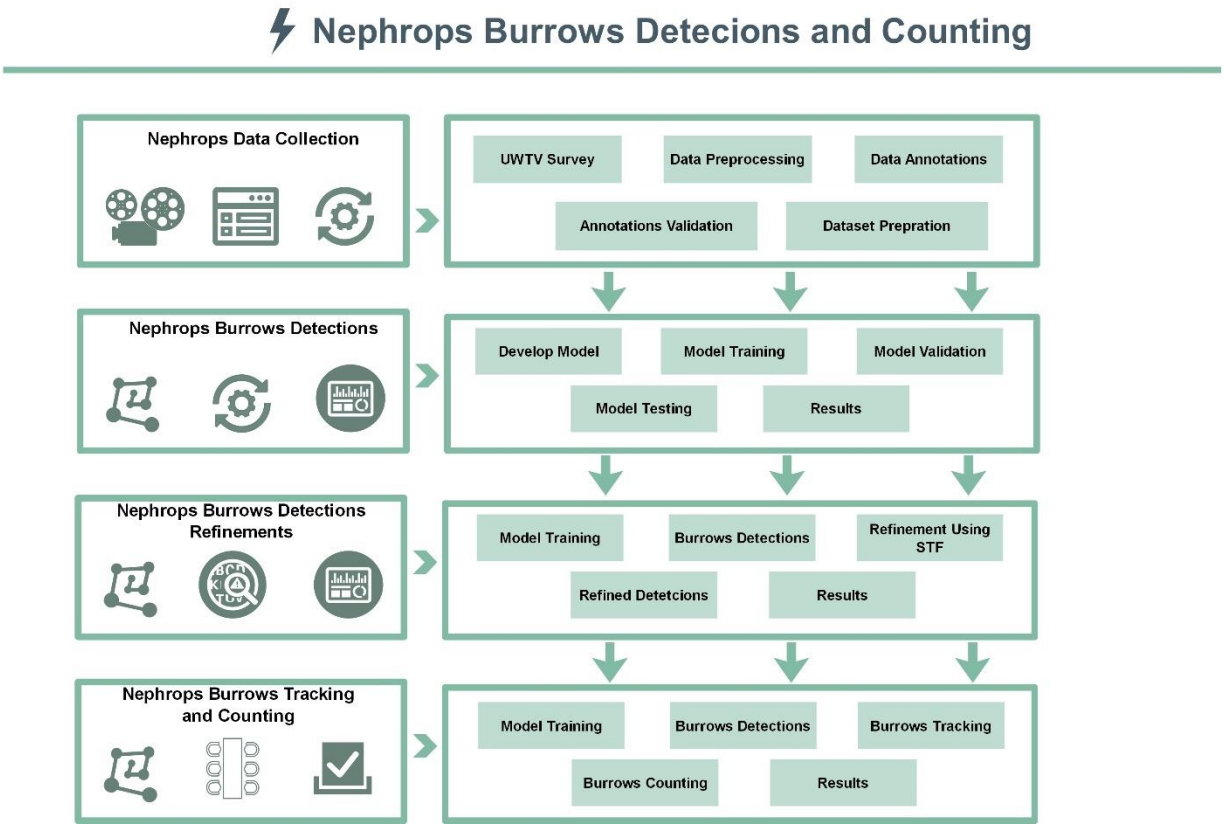


Figure 3.1: Proposed Methodology for Nephrops Burrows Detections and Counting.

3.3. Data Collection and Preparation

Currently, the *Nephrops* data are collected through the UWTV surveys and are reviewed manually by the trained experts. The data is collected using the sledges with cameras and lights. Many of the data were difficult to process due to complex environmental conditions. The data is collected in the form of still images and videos. Many stations have good data quality, but some have a lot of noise and need pre-processing before data usage. The collected data is stored in DVDs and external drives for survey counts. The *Nephrops* burrow systems are quantified following the protocol established by the ICES. The image data (which refers to video or still data) for each station is reviewed independently by at least two experts, and the counts are recorded for each minute onto the log sheet records.

With the massive amount of data collected for videos and images, manually annotating and analysing is laborious and requires much data review and processing time. Currently, all the stations are analysing the UWTV surveys manually to classify and count the *Nephrops*. Due to limited human capabilities, the manual review of image data requires a lot of time by trained experts to process the data to be quality-controlled and ready for use in stock assessment. Due to these factors, only a limited amount of collected data is used for analysis that usually does not provide deep insights into a

problem. Also, in some stations, it is tough for the human eye to classify and detect the burrows from a running video.

3.3.1. Data Collection Equipment

FU 22 Station

At FU 22, a sledge mounted with an HD video and stills CathX camera and lighting system at 75˚ to the seabed with a field view of 75 cm, confirmed using laser pointers, was used [25]. High-definition still images were collected with a frame rate of 12 frames per second with a resolution of 2048 x 1152 pixels for 10 – 12 min. The image data was stored locally in an SQL server and then analysed using different applications. Figure 3.2. shows the sledge used in data collection at FU 22.

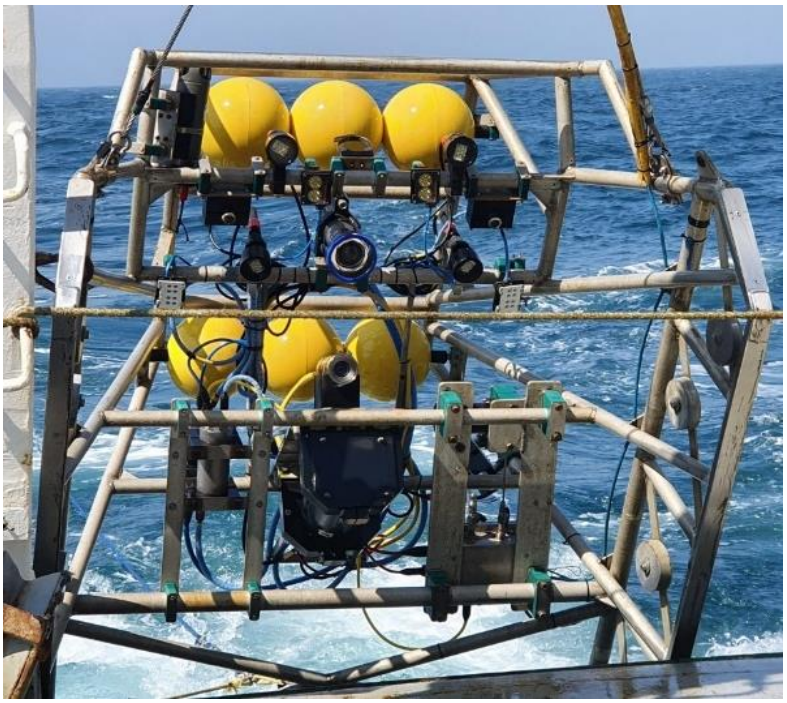


Figure 3.2: Sledge and equipment use in 2018 UWTV survey at FU22

FU 30 Station

At FU 30, a sledge was used to collect the data during the survey. Figure. 3.2 (a and b) shows the sledge used in data collection at FU 30. The camera is mounted on top of the sledge with an angle of 45º and a height of 80 cm from the sledge base. Videos were recorded using a 4K Ultra High Definition (UHD) camera (SONY Handycam FDRAX33) with a Lens of ZEISS® Vario-Sonnar 29.8 mm and an optical zoom of 10x. The sledge has a definition video camera and two reduced-sized CPUs with 700 MHz, 512 Mb RAM, and 16 GB storage. Four spotlights with independent intensity control are used to record the video with good lighting conditions. The equipment also has a two-line laser separated

by 75 cm to confirm the field of view (FOV) and a Li-ion battery of 3.7 V & 2400 mAh (480 Watt) to support the system's power. Segments of 10-12 minute video duration were recorded at 25 frames per second, with a 3840 x 2160 pixels resolution. The data were stored on hard disks and reviewed later manually by experts. Figure 3.3 shows the setup of the instruments mounted in the sledge and a sample image, and a complete description is presented in Table 3.1.

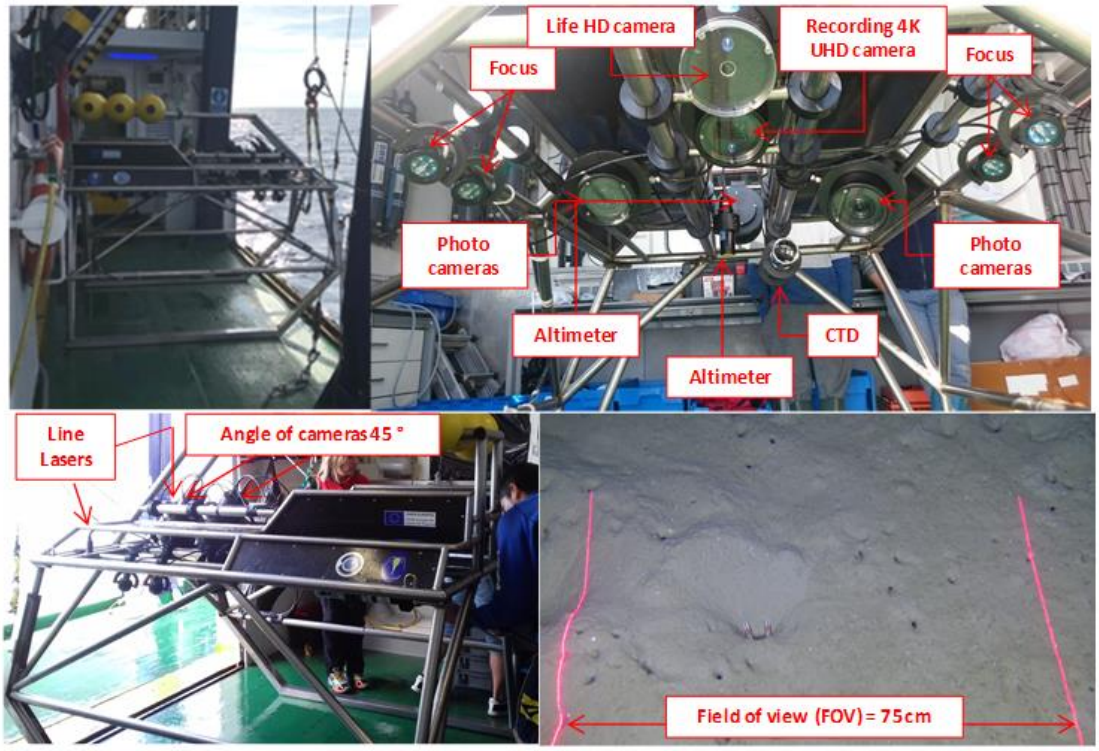


Figure 3.3: FU30 Equipment details used in data collection.

Table 3. 1: FU 30 Equipment details used in data collection.

Image System
Life Camera
Full HD (1920 × 1080) @ 30 fps Mounting angle 45°
***Recording Camera:* SONY FDRAX33**
4K Ultra HD (3840 × 2160) and Full HD (1920 × 1080) @ 50 fps Mounting angle 45°
***Photo camera:* SONY ILCE QX1**
20.1 MPixel Mounting Angle variable
Lighting System
28,640 lumens, distributed in 4 spotlights with individual intensity system TST-OFL 7000 (Thalassatech—Oil Filled LED)

Photogrammetry System
3-point lasers (5 mW & λ = 670 nm) forming a triangle of side 70 mm 2-line lasers (200 mW & λ = 670 nm) separated by 75 cm (Field of view)
Auxiliary System
Battery (Li-ion, size 18,650, 3.7 V & 2400 mAh = capacity 480 Wh)
Sensors
Altimeter: Tritech PA500 CTD (conductivity, temperature, and depth): AML Oceanographic MINOS X

3.3.2. Data Collection Procedure

At FU 30, the survey is conducted on 70 different stations. A station is a geostatistical location where the *Nephrops* burrow density is estimated to obtain the *Nephrops* abundance index over the known survey area using geostatistical analysis. At each station, the sledge was deployed and towed with constant speed between 0.6–0.7 knots to obtain the best possible conditions for counting *Nephrops* burrows. Once the sledge is stable on the seabed, video footage of 10–12 min at 25 frames per second is recorded, corresponding to approximately 200 m swept. Vessel position (dGPS) and sledge position, using a HiPAP transponder, are recorded every 1 to 2 s. The distance over ground (DOG) is estimated from the position of the sledge in all stations, and the field of view of the video footage is 75 cm (FOV), confirmed using two-line lasers. Out of all these 70 stations, seven are selected based on better lighting conditions, high contrast, high density of *Nephrops* burrows, and better visibility. The recorded footage was saved into hard disks for further analysis on *Nephrops* density. Each sledge used at different FU

Table 3. 2: Data collection equipment details at FU 22 and FU 30

Data Collection Equipment	**FU 22 (Ireland)**	**FU 30 (Gulf of Cadiz)**
Equipment Type	Sledge	Sledge
Camera	HD CathX	Sony FDRAX33
FPS	12	25
Field view	75 cm	75 cm
Image Resolution	2048 x 1152	3840 x 2160
Recording Duration	10-12 Minutes	10-12 Minutes
Density Range	0.31	0.35
Domain Area	3063 Km^2	3000 Km^2
No of Stations	42	70

needs specific equipment settings before its operation. Table. 3.2 summarizes the techniques and equipment used to collect data from FU 22 and FU 30 stations.

Camera Settings and Field of View

The sledge used in the survey has mounted cameras that help to record the videos and good-quality images. In the Gulf of Cadiz (FU 30) survey, the camera is mounted at 45◦ at 80 cm from the sledge base. The camera recorded high-definition videos and still images. The camera is equipped with 512 MB RAM and 16 GB storage. At the Marine Institute of Ireland (FU 22), the camera is mounted at an angle of 75◦ with a height of 75 cm from the sledge base. The camera is mounted at a certain angle with laser lights that show the field view. The camera recorded HD videos and still images and stored them in a local device. Table 3.3 shows the camera settings and field of view at FU 22 and FU 30 stations.

Table 3. 3: Camera and Field of View setting at FU 22 and FU 30

Institute	**Camera mounting angle°**	**Height centre camera lens to deck (cm)**	**Field of View (cm)**
MI-Ireland (FU 22)	40	75	75
Gulf of Cadiz (FU 30)	45	80	75

Laser set up

When lasers define the field of view, these have to be set up vertically, parallel and visible with a known horizontal separation distance on the bottom of the image on the monitor. At FU 22, the laser lights are not used to specify the field of view. Instead of these lights, laser pointers show the field of view. Figure 3.4 shows the laser pointers on a sample image of FU 22.

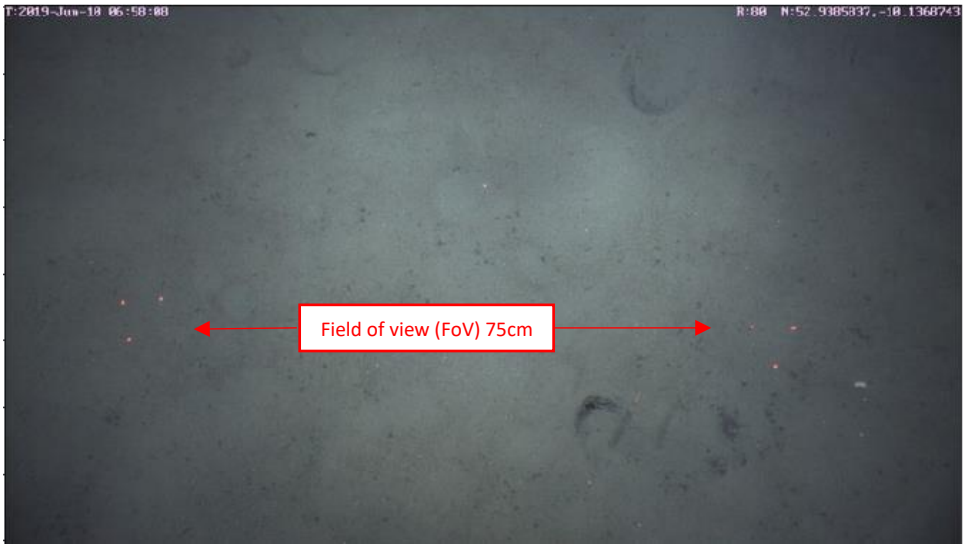


Figure 3.4: FU 22 Sample Image with laser pointers showing the field of view.

Figure 3.5 shows the field of view with red laser lights on the sample image of the FU 30 survey.

Figure 3.5: FU 30 Sample Image with laser lights showing the field of view

3.3.3. Data Characteristics

The data is collected in the form of still images and videos. The collected data is stored in DVDs and external drives during the survey and later viewed, arranged and analyzed accordingly.

FU 22 Data Characteristics

At FU 22, 42 UWTV stations were surveyed in 2018. The 10-12 minute video was recorded at different frame rates ranging from 15 fps, 12 fps, and 10 fps at Ultra HD. Also, the high-definition images were captured with the camera. The images were recorded with a resolution of 2048 x 1152 pixels. Figure.

3.6 shows the high-definition still images from the 2018 UWTV survey HD camera. The top image shows a burrow system composed of three holes in the sediment, whereas in the bottom image, a single *Nephrops* individual is seen outside the burrows. Illumination is better near the center of the field view and decreases to the borders of the images. The camera angle shows 75 degrees with a ranging laser (red dots) on the screen. A *Nephrops* burrow system may be composed of more than one entrance, and in this paper, our focus is to detect the individual *Nephrops* burrow entrances.

Figure 3.6: High definition still images from 2018 UWTV survey for FU 22 station

FU 30 Data Characteristics

At FU 30, the data is only collected through videos. The videos are recorded at 25 frames per second in good lighting condition. Every station at FU 30 has recorded video footage of 10-12 minutes. Figure. 3.7 shows the high-definition images from the 2018 UWTV survey of FU 30. FU 30 images show better illumination (in terms of contrast and homogeneity) than FU 22. Pink lines on the images correspond to red laser lighting to 75cm width searching areas (the red color is pink due to the distortion produced by different attenuation of light wavelength in water.

Figure 3.7: High definition still images from 2018 UWTV survey for FU30 station

3.3.4. Data Preprocessing

The underwater environment is hard to analyze as it presents formidable challenges for Computer Vision and machine learning technologies. The image classification and detection in underwater images differ significantly from other visual data. Also, data collection in an underwater environment is the biggest challenge. One reason for this is light, as light and water are not considered good friends, because

when light passes through the water, it cannot absorb and reach the sea surface, which makes the images or videos a blurring effect. Also, scattering and non-uniform lighting make the environment more challenging for data collection.

Poor visibility is a common problem in the underwater environment. The poor visibility is due to the ebb and flow of tides, which causes fine mud particles to suspend in the water column. The ocean current is another factor that causes frequent luminosity change. The visual features like lightning conditions, color changes, turbidity, and low pixel resolution make it challenging. So, we need some preprocessing before the use of data.

Data collected from FU 22 and FU 30 is converted into frames. The collected data set has a lot of frames with low and non-homogeneous lightning and poor contrast. The frames without burrows or poor visibility are discarded during the annotation phase, and consecutive frames with similar information are discarded. Figure. 3.8 shows a mosaic of images that contains much noise and is hard to analyze. Some images are blurry due to the underwater environment and lightning conditions, while others have very rough surfaces and almost nothing or significantly less density of *Nephrops*.

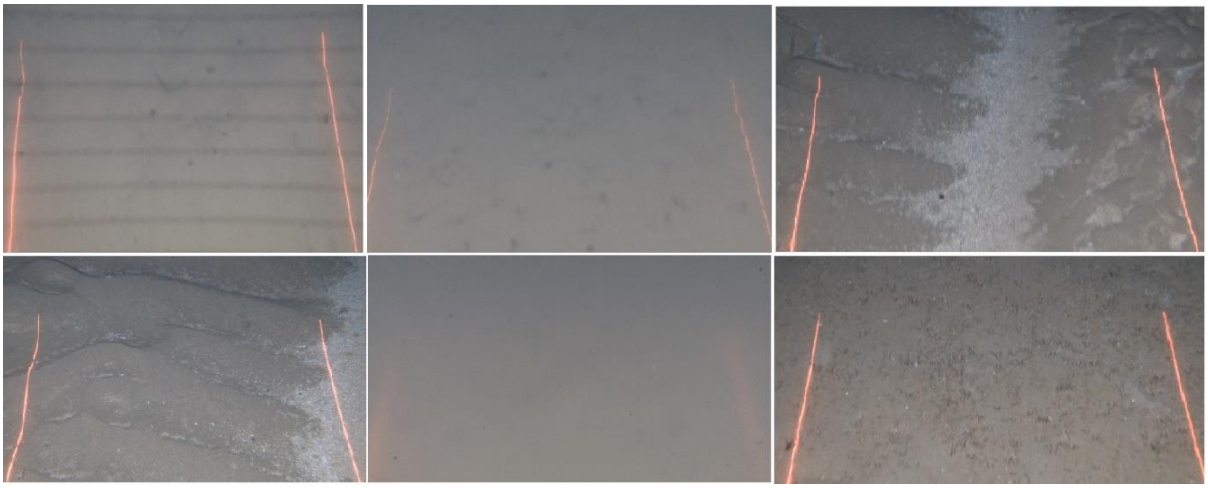


Figure 3.8: Images with poor visibility or low Nephrops density

3.3.5. Image Annotations

Image annotation is a technique Computer Vision uses to create training and test ground truth data, as supervised deep learning algorithms require this information. Usually, any object is annotated by drawing a bounding box around it.

Currently, the marine experts who work with *Nephrops* burrows are not using any annotation tool to annotate *Nephrops* burrows, as this is a time-consuming job. In this phase, the images are annotated to overcome this challenge, and all recorded annotations are validated by the marine experts from Ireland and Cadiz institutes before training and testing processes.

With the massive amount of data collected for videos and images, manually annotating and analyzing it is laborious and requires much data review and processing time. Due to limited human capabilities, the manual evaluation of image data requires a lot of time by trained experts to process the data to be quality-controlled and ready for stock assessment. Due to these factors, only a limited amount of collected data is used for analysis that usually does not provide deep insights into a problem [25].

In this work, two different mechanisms are adopted to annotate the *Nephrops* burrows because the aim is to provide a mechanism that is easy to understand by Marine scientists. Here are some of the methods.

Image Annotation using Semi-Auto Annotation

A semi-auto annotation tool is developed in MATLAB. This tool inputs a frame and draws the bounding boxes on the potential *Nephrops* burrows. This tool used hand-crafted features to read the characteristics of burrows and annotate the burrow as *Nephrops*. Figure 3.9 shows the outcome of an annotated frame from that tool. Some of the major problems of this tool are a high False Positive (FP) ratio, restricted to only one frame at a time, and not providing any option to modify the already annotated burrows. The tool initially annotated specific burrows based on the hand-crafted features of each burrow on the given frame, leading to many false annotations.

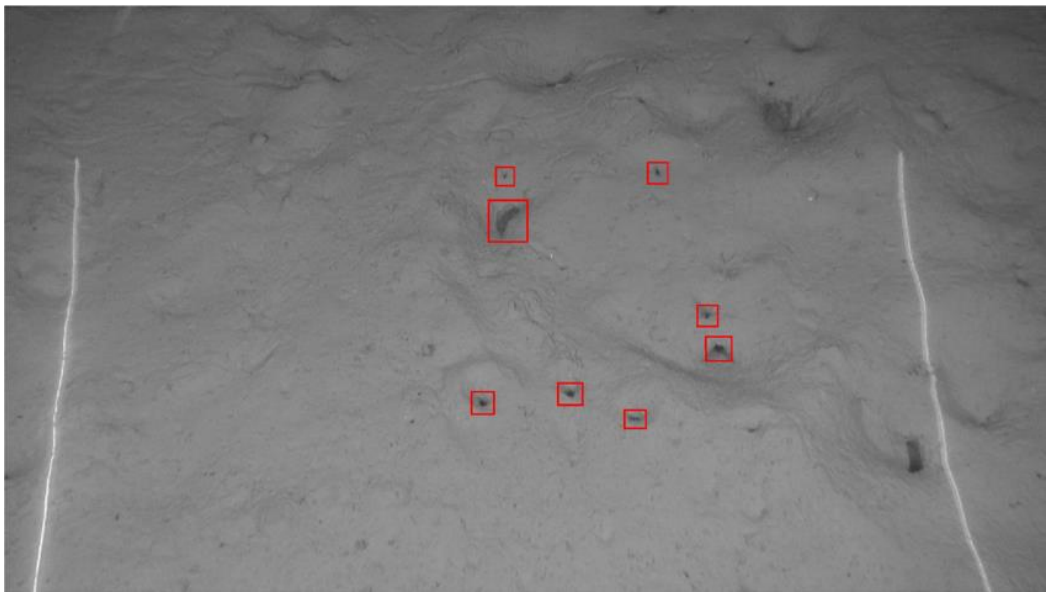


Figure 3.9: Nephrops burrows annotation using Semi-Automation tool

Image Annotation using Microsoft VOTT

We adopt the mechanism to annotate the burrows manually in the Microsoft VOTT image annotation tool [26] using Pascal VOC format. The saved XML

annotation file contains each object's image name, class name (*Nephrops*), and bounding box details. The annotated frames led to formulating the ground truths (GT) for model training. To create the datasets for training and testing, from the set of annotated frames (more than 100,000), only frames with *Nephrops* burrows are selected, using the criteria of using only one frame per individual object, selected to increase the diversity of its appearance, which the aim of creating a small dataset which contained most of the typical cases of *Nephrops*

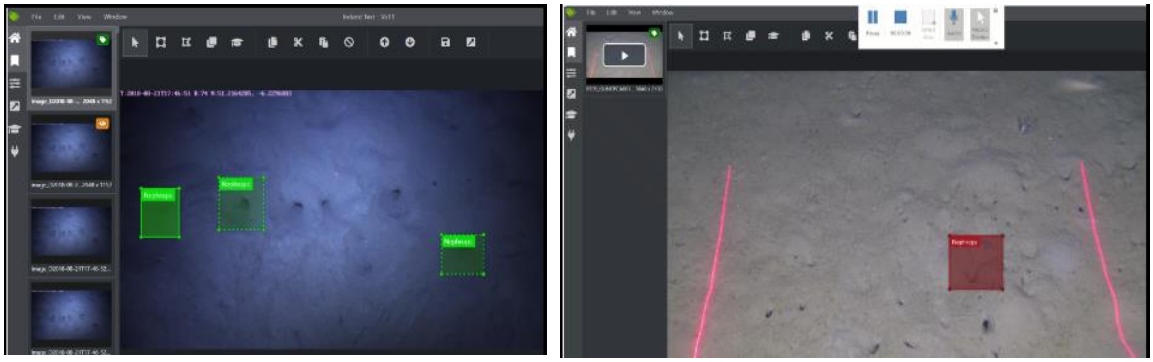


Figure 3.10: Manual Annotation in a frame from (a) FU 22 and (b) FU 30 UWTV survey using VOTT

burrows. The image annotation was one of the most consuming and sensitive jobs we finished in many months. Figure. 3.10 shows two screenshots from the FU 22 and FU 30 UWTV surveys that are manually annotated.

Seven stations are annotated for FU 30 and seven for FU 22. A total of 248 images were annotated for seven stations of FU 30, and 978 images were annotated for seven FU 22 stations before the validation stage. In general, there is a higher density of *Nephrops* burrows from FU 22 compared to FU 30, which is a factor of population dynamics.

3.3.6. Validation of Annotation

The *Nephrops* burrows annotation is a tedious job, and it requires a lot of experience to annotate a burrow because different species build burrows with similar appearance on the bottom of the sea. Once all the burrows are annotated, it is essential to validate each with the advice of marine experts from the IEO institution, Gulf of Cadiz. This process took a long time as confirming every annotation is time-consuming and sensitive. After validating each annotation, a curated dataset is used for training and testing the model.

3.3.7. Dataset Preparation

Dataset preparation is essential to train any model. In 2018, 19 surveys covered the 25 FU's in ICES and one geographical subarea (GSA) in the Adriatic Sea [27]. These surveys were conducted using standardized

equipment and agreed protocols under the remit of WGNEPS. At FU 22, thousands of images were recorded in the 2018 survey. FU 22 provided a few hundred images for experimental purposes. Out of thousands of recorded images, 1133 high-definition images were manually annotated from FU 22.

At FU 30, 70 UWTV stations were surveyed in 2018. Out of 70 surveyed stations, 10 were rejected due to poor visibility and lighting conditions. Seven stations are selected for experimentation, with good lighting, low noise and few artefacts, higher contrast, and high density of *Nephrops* burrows. The data from seven stations are considered for annotations. So, each video is around 15,000 - 18,000 frames. 105,000 frames were recorded from seven different stations of the 2018 data survey.

After validating all the annotations, the annotated images are recorded into XML files and converted to TFRecord files, a sequence of binary strings that TensorFlow requires to train the model. The dataset was divided into two independent groups, the first for training and the second for testing. Table 3.4 shows the details of the dataset used in the experimentations.

Table 3. 4: Dataset Preparation

Dataset Distribution			
Functional Unit	Training Images	Testing Images	**Total**
FU 30 Dataset	200 (80%)	48 (20%)	**248**
FU 22 Dataset	906 (80%)	227 (20%)	**978**

This page intentionally left blank.

Chapter 4: *Nephrops norvegicus* Burrows Detections Using Deep Learning

4.1. Introduction

Nephrops data are collected, and trained experts review UWTV surveys manually. Many of the data were difficult to process due to complex environmental conditions. Burrows systems are quantified following the protocol established by ICES [3][6]. In this chapter, an automated system is proposed based on deep learning techniques that detects and counts the *Nephrops* burrows in video footage with high precision. The proposed method introduces a deep-learning-based automated way to identify and classify the *Nephrops* burrows. This research uses the current state-of-the-art Faster RCNN models Inceptionv2, MobileNetv2, ResNet50 and ResNet101 for object detection and classification. Also, the *Nephrops* burrows are detected using YOLOv3, which is later used for burrow tracking and counting.

This chapter considers data from the Gulf of Cadiz (FU 30) and the Smalls (FU 22) *Nephrops* grounds to detect the *Nephrops* burrows using the image data collected from different stations in each FU using our proposed methodology. This chapter aims to automatically apply deep learning models to detect and classify the *Nephrops* burrows. Current state-of-the-art Faster RCNN [28] models Inceptionv2 [30] and MobileNetv2 [31], ResNet50 [32], and ResNet101 [33] and YOLOv3 [34] are trained for *Nephrops* burrows detection.

This chapter makes a significant advancement for all the groups working on the *Nephrops norvegicus* counting for stock assessment, where it is shown to detect and accurately count the *Nephrops* burrows automatically. The rest of the chapter is sectioned as follows. Data Learning and Supervised Learning is discussed in section 4.2 and 4.3. The Neural network is explained in section 4.4. Transfer Learning is defined in section 4.5. The complete architecture of *Nephrops norvegicus* Burrows detections is described in section 4.6.

4.1.1. Deep Learning

Deep learning is an advanced machine learning form that uses an Artificial Neural Network (ANN) with three or more layers. The deep neural networks

learn from a large amount of data and support the AI applications to perform analytical and physical tasks without human interventions. The conventional machine-learning techniques have limited capabilities in terms of processing, and it requires a lot of engineering by hand and domain expertise to build a feature extractor that can extract the information from raw data. [35]. On the other hand, deep learning did not require much pre-processing typically needed for the machine learning process. Machine learning and deep learning models are also capable of different types of learning, usually categorized as supervised, unsupervised, and reinforcement learning. [36]. The neural networks are a typical example of supervised learning.

4.1.2. Supervised Learning

Supervised learning, also called supervised machine learning, is the most common machine and deep learning type. In this learning, the labelled data is used to train the algorithms. The algorithms are used to classify the objects like a house, an animal, or a person. In supervised learning, an objective function measures the error between the output scores and the desired pattern of scores. The machine algorithms adjust these parameters to reduce the model error, and these parameters are often called the 'weights. A deep learning algorithm can have hundreds of millions of adjustable weights. These weights are adjusted using a gradient vector. Various algorithms and computational techniques are used in supervised learning like Neural networks, Naive Bayes, Linear regression, Logistic regression, Support vector machine (SVM), K-nearest neighbour, and Random Forest. The current research uses deep learning models to classify and count the *Nephrops* burrows.

4.1.3. Neural Network

Neural networks, or artificial neural networks (ANNs), are the core of deep learning algorithms. The human brain activity inspires these networks. [35]. The ANN comprises an input layer, an output layer and one or more intermediate hidden layers. Each layer comprises specific nodes, and each node is connected to another layer node with an associated weight and threshold value. The essential parts of neural networks are [36]:

- Neurons: The neuron is a set of inputs, weights, and an activation function that takes the output from the layer ahead of it.
- Hidden Layers: These layers have many neurons with many hidden layers.
- Input Layer and Neurons: The input layer consists of many input neurons.
- Output Layer: The output layer relates to the hidden layer and generates the desired results.

- Synapse: The synapse connects the neuron and the layers.

Deep learning is like an extensive neural network with multiple hidden layers. Figure 4.1 shows the basic structure of a deep neural network.

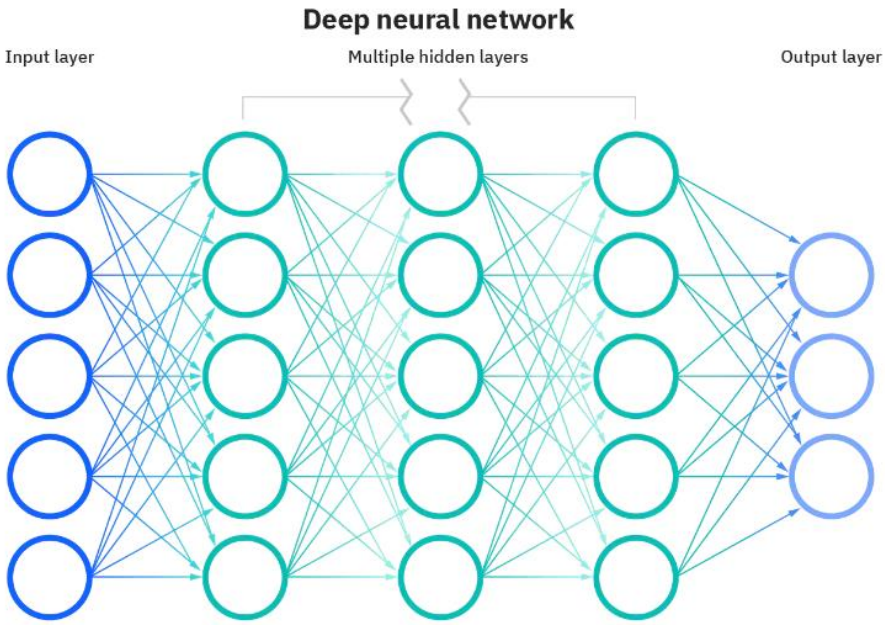


Figure 4.1: Deep neural network [35]

4.1.4. Transfer Learning

Transfer learning is a technique to reuse the already pre-trained model on a new problem. Transfer learning is a prevalent approach in deep learning where the pre-trained models are used as the starting point and trained with your prepared dataset. It is beneficial as most problems do not have enough labelled data to train those models. In transfer learning, the model is trained for some problem and is used in a way for any other problem. Transfer learning decreases the training time. [37].

4.2. Background Study

Object detection and classification is a challenging computer vision problem. Researchers have developed many methods for object detection and classification tasks. The existing object detection approaches use handcrafted feature-based models [55-58] and deep features models [59]. The hand-crafted features models use essential features such as shape [60], texture [61–63], and edges [64,65] to train the classifier. On the other hand, convolutional neural networks automatically learn hierarchical features from the training set. Deep learning replaces the handcrafted parts and introduces efficient object detection and classification algorithms. Over the last few years, deep learning models have enjoyed tremendous success in various object

detection and classification tasks. Due to this reason, deep learning models are also employed in the detection and classification of underwater species. Although the aquatic environment is complex and challenging compared to the ground, deep learning algorithms perform much better than conventional and handcrafted features. State-of-the-art deep learning-based object detectors include region-based convolution networks (R-CNN), Fast R-CNN [29], and Faster R-CNN [28]. R-CNN uses deep ConvNet to classify the object proposals. R-CNN algorithm is computationally expensive as it uses a selective search [66] strategy to generate many object proposals, followed by the object proposal classification step. On the other hand, Fast R-CNN improves R-CNN, where a faster training process is achieved compared to R-CNN. Fast R-CNN uses multitasking to update all the network layers and handle the loss, improving the network's speed and accuracy. Compared to both methods, Faster R-CNN introduces a region proposal network (RPN), combining the RPN with Fast R-CNN into a single network.

Li et al. [67] developed a deep-learning model for detecting marine objects. The model detects and recognizes fish using a deep convolutional network. They applied the Fast R-CNN algorithm to classify the twelve different classes of underwater fish. They also introduced a dataset of 24,272 images of all these classes. They achieved more than 90% accuracy in detection.

Similarly, Villon et al. [68] applied deep learning algorithms to the Fish4Knowledge dataset project to detect and classify the fishes. Rathi et al. [69] combined Faster R-CNN with three classification networks (ZF Net, CNN-M, and VGG16) to detect 50 fish and crustacean species from Queensland beaches and estuaries. The regional proposal method comprises a regional proposal network and a classifier network. Xu et al. [70] applied the YOLO deep learning model to recognize the fishes in underwater videos. They used three different types of datasets that were recorded at real-world waterpower sites. They achieved an mAP of up to 53.92%. Mandal et al. [71] presented a Faster R-CNN approach using deep neural networks to identify the fishes and their different species. Gundam et al. [72] also proposed a fish classification technique based on the Kalman filter that used partial automation of fish classification from underwater videos. Jalal et al. [73] proposed a hybrid approach that combines YOLO-based object detection with optical flow and Gaussian matrix models to detect and classify fish from underwater videos. A similar method based on YOLO to detect and classify the fish was proposed by Sung et al. [74]. They used 892 images and achieved the fish classification accuracy of up to 93%. Jager et al. [75] proposed a deep CNN approach based on AlexNet architecture to classify fish species. They used the dataset of LifeCLEF 2015. Zhuang et al. [76] proposed a deep learning model based on an SSD detector to identify the fishes and their species automatically. In their approach, they used ResNet-10 as a classifier for species identification. Zhao et al. [77]

proposed an automatic detection and classification method for fish and underwater species. The proposed method, "Composed FishNet", is based on the composite backbone and a path aggregation network. The combined backbone method is the improvement of ResNet. The enhanced path aggregation network is designed to improve the semantic information caused by unsampling. The results show they achieved an average precision (AP) of 75.2%. Labao et al. [78] proposed a multilevel object detection network that used R-CNN as a network framework. Their proposed network contained two region proposal networks and seven CNNs connected by long short-term memory (LSTM). The proposed network improved performance over the simple one-stage detection networks. Salman et al. [79] proposed an R-CNN-based two-stage automatic fish detection and location method. They combined the fish motion information with the background and optical flow information to generate the candidate region of the fish. Their proposed model requires a fixed-size input image, and the candidate region ex-traction also needs substantial disk space.

Deep learning models also have been employed to detect marine objects other than fishes, such as plankton and corals. These two are also significant components of the underwater marine ecosystem. Plankton are the basics of aquatic food. Dieleman et al. [80] used a deep neural network to classify plankton. They introduced the inception module for image information extraction. Lee et al. [81] also proposed a deep neural network for plankton classification on a large dataset. Their convolutional neural network used three convolutional layers and two fully connected layers. The problem with the coral classification is its color, size, texture, and shape. Shiela et al. [82] introduced a local binary pattern for texture and color coordination. For classification purposes, they used the neural network with three backpropagation layers. Elawady et al. [83] used supervised CNN to classify corals. Table 4.1 summarizes the key findings of the papers discussed in this section.

In recent years, some researchers also started to implement machine learning and deep learning models to classify and detect the *Nephrops* norvegicus species. Sokolova et al. [136] introduce specialized software called NepCon, which is based on machine learning techniques to classify, detect, and count the *Nephrops* norvegicus. Their work contributes to the underwater image acquisition of *Nephrops* species. The proposed system is a cost-effective, stable, and robust underwater imaging system designed for automatic detection and tracking of *Nephrops*. The work has been tested in a controlled environment and is not capable of detecting and tracking the burrows of *Nephrops*. Avsar et at. [137] proposed a deep learning-based detection and counting mechanism for *Nephrops* norvegicus individuals. They used a public dataset from Denmark and applied the YOLOv4 model for detecting *Nephrops* individuals. Later they used the SORT algorithm for tracking and counting individual *Nephrops*. The

result shows a promising mAP of 97.82%. This work solely focuses on the classification, detection and counting of *Nephrops* individuals.

Table 4. 1: Underwater object detection with key findings

Approach	Year	Object Detection	Dataset	Performance Parameters
Deep Convolutional Network [67]	2015	Marine Objects	ImageCLEF_Fish_TS dataset 24272 Images	mAP
HOG, SVM and Deep Learning [68]	2016	Fish Detection	Fish4Knowledge 13000 fish thumbnails	Precision, Recall, F-Score
Faster R-CNN (ZF Net, CNN-M, VGG16) [69]	2018	Fishes & crustacean species	Fish4Knowledge 27,142 Images	AP
YOLO [70]	2018	Fishes	Three datasets	mAP
Faster R-CNN [71]	2018	Fishes	Uni of Sunshine Coast 12365 Images	mAP
YOLO based Hybrid approach [73]	2020	Fish Classification	LifeCLEF 2015 93 Videos	F-Score
YOLO [74]	2017	Fish detection	892 Images	Precision, Recall, FPS
CNN AlexNet [75]	2016	Fish Classification	LifeCLEF2015	AP, Precision, Recall,
ResNet-10 [76]	2017	Underwater Species	SEACLEF2017	AP
Composed FishNet [77]	2021	Fish and Underwater Species Detections	SeaCLEF 2017 20,0000 images	AP, F-Measure
Multilevel R-CNN [78]	2019	Fish detection	300 Underwater Images	Precision, Recall, F-Score
Two-stage R-CNN [79]	2019	Fish detection	Fish4Knowledge, LCF-15	Precision, Recall, F-Score
Three layers CNN [81]	2016	Plankton detection	WHOI-Plankton database 3.2 Million Images	F1-Score
Mask R-CNN [136]	2021	Nephrops	University of Denmark (DTU) repository	JI, mAP
YOLOv4 [137]	2023	Nephrops	University of Denmark (DTU) repository	Precision, Recall, mAP

4.3. *Nephrops norvegicus* Burrows detections

Many scientists employ Artificial Intelligence-based tools to analyze marine species with the advancement of artificial intelligence and computer vision technology. Deep convolutional neural networks have shown tremendous success in the tasks of object detection [38,39], classification [40,41], and segmentation [42]. These networks are data-driven and require a lot of labelled data for training. A deep learning-based system is proposed to automatically detect and classify the *Nephrops* burrow systems that take underwater video data as input. The network learns hierarchical features from the input data and detects the burrows in each input video frame. The cumulative sum of detections, all video frames give the final count of *Nephrops* burrows. The FU 30 and FU 22 datasets were collected using different image acquisition systems (Ultra HD 4 K video camera and HD stills camera, see section 3.3) from diverse *Nephrops*

populations. The image data is annotated using the Microsoft VOTT image annotation tool [26].

4.3.1. Proposed Framework of *Nephrops* Burrows Detections

The actual methodology used to count *Nephrops* is explained in the previous section. In this work, the old paradigm of counting *Nephrops* burrows is replaced with an automated framework that automatically detects and counts the number of *Nephrops* burrows quickly and accurately. Figure. 4.2 shows the high-level diagram of the proposed methodology. Video files are converted to frames using OpenCV, then images are manually annotated using the VOTT image annotation tool. The marine experts verify the annotated data before being used for training the deep neural network. Figure. 4.3 shows the detailed steps of the research methodology used in this work.

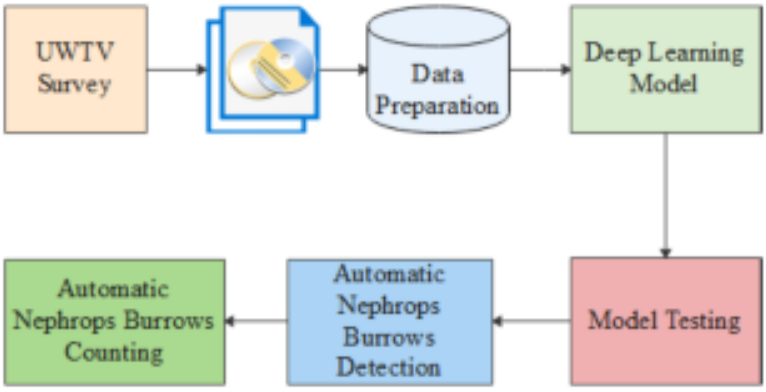


Figure 4.2: Block diagram of proposed methodology

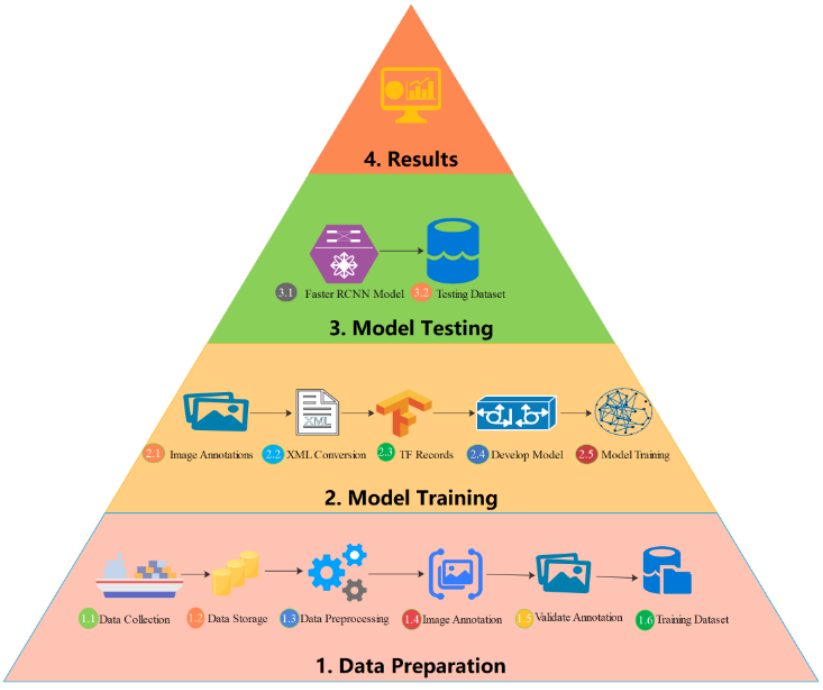


Figure 4.3: Architecture of proposed methodology

4.3.2. Model Training

To train the models, the transfer learning [37] is utilized to fine-tune the Faster R-CNN Inceptionv2 [30], MobileNetv2 [31], ResNet50[32], ResNet101[33] and YOLOv3 [34] models in TensorFlow [36]. Inceptionv2 is one of the architectures that have a high degree of accuracy. The basic design of Inceptionv2 helps to reduce the complexity of CNN. A pre-trained version of the network model trained on the COCO dataset [43] is used.

Inception v2

Inception networks are considered one of the big milestones in CNN. Before the Inception networks, CNN only added layers to deepen the networks. The Inception network, on the other hand, used complex engineering to increase the accuracy and speed of the network. Inception v1 was the first network of the Inception series. Inception v2 is the second network in the generations of Inception convolutional neural network architectures. Inception v2 reduces the "representational bottleneck". The representational bottleneck happens, which means the loss of information due to drastic alteration in the input. For computational complexity, the Inception v2 network used a smart factorization method to factorise the 5x5 convolution to two 3x3 convolutions. In our project, we implemented the Inceptionv2 architecture to detect and classify the "Nephrops burrows" from two different datasets: FU22 and FU30, with image resolutions of 1229x691 and 3840x2160, respectively. The architecture starts with a series of six convolutional layers, designed for initial feature extraction from the input images:

The first 3x3 convolutional layer processes the input image, producing an output feature map. We added padding to maintain the size of the output. So, if the input is from the FU22 dataset, for example, the output could remain same to 1229x691. The second 3x3 convolutional layer further processes the data. Again, assuming standard padding and stride, the size might remain similar to the output of the first layer. The third layer, a 3x3 convolutional padded layer, explicitly maintains the dimensionality due to padding, ensuring the output feature map doesn't reduce in spatial dimensions. Following this, the fourth 3x3 convolutional layer again processes the feature map. Depending on padding and stride, the size could remain consistent with the previous layers' outputs. The fifth and sixth 3x3 convolutional layers continue this pattern, each building on the feature maps produced by the previous layer, typically maintaining the size if padding is applied to counteract the natural reduction from the convolution process.

In between and following these convolutional layers, there are two pooling layers aimed at reducing the dimensionality and computational requirements. Pooling layers, typically with a 2x2 filter and a stride of 2, would reduce the height and width of the feature map by half, enhancing the model's efficiency and focus on relevant features. For example, if a pooling layer follows the initial convolutional layers, the output dimension might reduce from 1229x691 to approximately 614x345 for images from the FU22 dataset.

After processing through the initial convolutional and pooling layers, the data then enters the Inception modules. Each module processes inputs with a variety of convolutional operations and pooling, leading to outputs that capture a wide array of features at different scales.

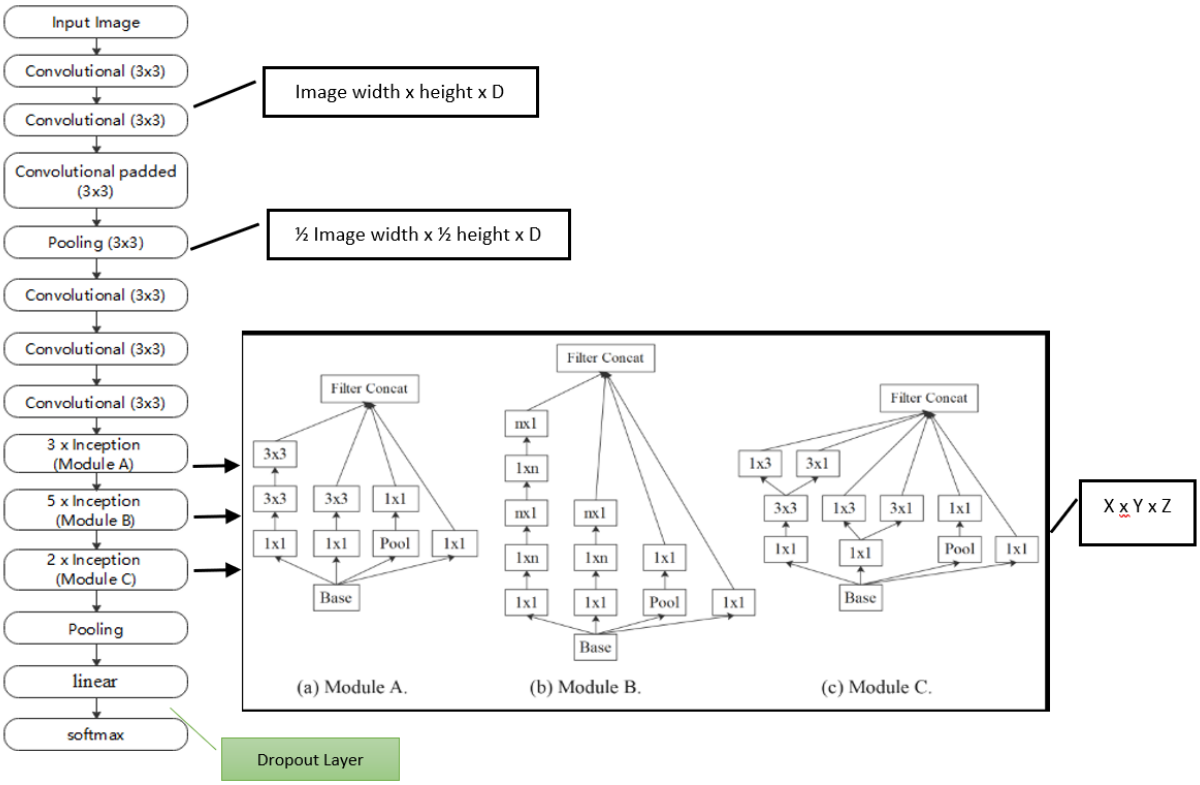


Figure 4.4: Inceptionv2 layers and architecture [30]

For fine-tuning, we utilized batch normalization after each convolutional layer to ensure more stable and faster training. Dropout layers were included to prevent overfitting. The entire model was trained using the Momentum optimizer [44] with a decay rate of 0.9 and a learning rate of 0.01, processing images one at a time. We carefully set the max pooling parameters and applied gradient clipping [45] with a threshold of 2.0 to ensure smooth and effective training. Figure 4.4 shows the Incetionv2 layers and its architecture. The figure shows the output of each layer in general. Each convolution layer output will be (Image width x Image height x D), where D is the depth of convolutional layer based on the number of filters. In this work a filter of size 64 is used. The X and

Y are the spatial dimensions. These values represent the width and height of the output feature map. For example, if the original image size is 1229x691 and a pooling layer with a stride of 2 is applied, then both dimensions would roughly halve, making X ≈ 614 and Y ≈ 345 after the first pooling layer. Z is concatenation of each path of the inception module.

MobileNetv2

With the revolutionization of neural networks, the accuracy, speed, and performance cost a lot in terms of resources and capabilities. The devices with fewer resources cannot cope with the immense computational power of neural networks. The MobileNet v2 is tailored for a constrained resources environment. The MobileNet v2 CNN architecture was proposed by Sandler et al. [31]. The main contribution of this network is the layer module, i.e., they introduce the inverted residual with a linear bottleneck. One of the main reasons for choosing the MobileNetv2 architecture was the relatively small training dataset from FU 30. This architecture optimizes the memory consumption and execution speed with minor errors. MobileNetv2 architecture has depth-wise separable convolution instead of conventional convolution. This architecture initially has a convolution layer with 32 filters, followed by 17 residual bottleneck layers (Figure 4.5).

The initial convolutional layer uses a 3x3 kernel and used for initial feature extraction. Each bottleneck block contains three layers convolutional, Depthwise convolutional and convolutional. The 1x1 convolutional layer after the bottleneck block is used to serves additional feature combination and reduction, preparing features for classification. The output size of fully connected projection layer is equals to the number of classes in the classification task. The pooling reduces each channel in the feature map to a single value to summarizes the spatial information. Our experiments achieved the best model result with RMSProp [46] momentum with a decay of 0.9. This work uses a learning rate of 0.01 to balance fast convergence with the stability of the training process, a batch size of 24 is set to optimizing the trade-off between computational efficiency and model performance. We employed a truncated normal initializer for its role in preventing layer activation outputs. The L2 regularization is used with Rectified Linear Unit (ReLU) as an activation function to mitigate overfitting while preserving non-linearity in activations. The box predictor used in the MobileNet model was the Convolutional box predictor. Table. 4.2 shows the parameter list and values used in the MobileNet v2 and Inception v2 models.

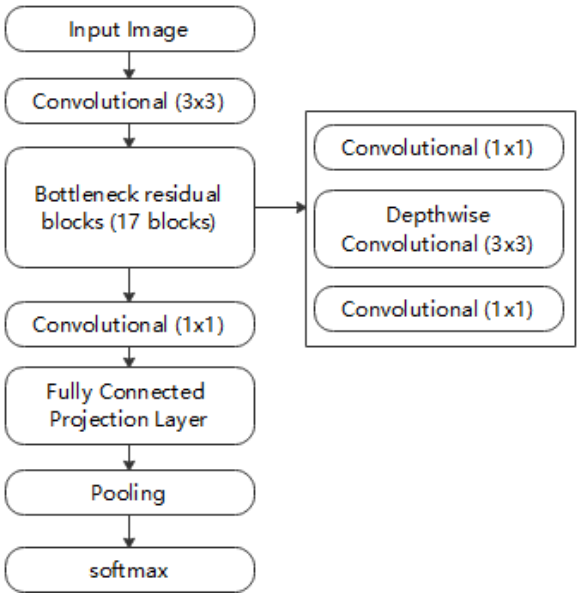


Figure 4.5: MobileNet v2 model architecture [31]

Table 4. 2: Inception v2 and MobileNet v2 Model Training Parameters

Parameters	Inceptionv2	MobileNetv2
Number of Classes	01	01
Optimizer	Momentum	RMSProp
Momentum Rate	0.9	0.9
Learning Rate	0.01	0.01
Batch Size	1	24
Initializer	truncated_normal_initializer	truncated_normal_initializer
gradient_clipping_by_norm	10	-
Regularization	L2	L2
Activation Function	Softmax	RELU
Maxpool kernel size	2	-
Maxpool stride	2	-
Box Predictor	Mask RCNN box predictor	Convolutional box predictor

ResNet50

ResNet50 [32] is a variant of the model ResNet. The ResNet50 has 48 convolutional layers, one max pool, and one average pool layer, so it is a 50-layer-deep convolutional network. Out of these 50 layers, one layer is used in the first convolution with a kernel size of 7 × 7 64 kernels with stride 2 and a max pool of size 3 × 3 with stride 2. Nine layers are used in the second convolution with a kernel size of 1 × 1 64 kernels and 3 × 3 128 kernels. In the next step, 12 layers are used with 1 × 1128; after that, a kernel of 3 × 3,128, and, at last, a kernel of 1 × 1,512.

The fourth convolution uses 18 layers with a kernel of 1 × 1,256 and two more kernels with 3 × 3,256 and 1 × 1,1024. The fifth convolution uses nine layers with a 1 × 1,512 kernel with two more of 3 × 3,512 and 1 × 1,2048. Finally, the last layer is used for the avg pool and a softmax function. ResNet50 is a widely used ResNet model.

ResNet101

The ResNet101 [33] is a dense convolutional neural network that is 101 layers deep. The first convolution has a kernel size of 7 × 7 64 kernels with stride 2 and a max pool of size 3 × 3 with stride 2. Nine layers are used in the second convolution with a kernel size of 1 × 1 64 kernels and 3 × 3 128 kernels. In the next step, 12 layers are used with 1 × 1,128; after that, a kernel of 3 × 3,128, and, at last, a kernel of 1 × 1,512. The fourth convolution uses 69 layers with a kernel of 1 × 1,256 and two more kernels of 3 × 3,256 and 1 × 1,1024. The fifth convolution uses 9 layers with a 1 × 1,512 kernel with two more of 3 × 3,512 and 1 × 1,2048. Finally, the last layer is used for the avg pool and a softmax function. The ResNet50 and ResNet101 have better accuracy when compared to the other models for our problem.

Table. 4.3 shows the parameter list and values used in the ResNet50 and ResNet101 models.

Table 4. 3: ResNet50 v2 and ResNet101 Model Training Parameters

Parameters	ResNet50	ResNet101
Number of Classes	01	01
Optimizer	Momentum	Momentum
Momentum Rate	0.9	0.9
Initial Learning Rate	0.0003	0.0003
Batch Size	1	24
Initializer	truncated_normal_initializer	truncated_normal_initializer
gradient_clipping_by_norm	10	10
Regularization	L2	L2
Activation Function	Softmax	Softmax
Maxpool kernel size	2	2
Maxpool stride	2	2
Box Predictor	Mask RCNN box predictor	Mask RCNN box predictor

YOLOv3

The YOLO detector was introduced in 2016 with its first version, called YOLO v1, by Redmon et al. [34], achieved 63.4 mAP on the Pascal VOC data set; the second version, called YOLOv2 and YOLO900 was introduced in 2017 by Redmon et al. [47] that achieve 78.6 mAP on the same data set used in YOLOv1. In 2018, Redmon et al. [34] introduced YOLOv3, which used a complex MS COCO data set with 80 classes and achieved 57.9 mAP. In 2020, multiple versions of YOLO appeared from different authors that used the same MS COCO data set and achieved a higher mAP as compared to YOLOv3, they listed as YOLO v4 by Bochkovskiy et al. [48] achieve 65.7 mAP, Scaled YOLO v4 by Chein et al. [49] achieve 66.2 mAP, PP-YOLO from Xiang et al. [50] achieve 65.2 mAP and YOLO v5 by Ultralytics achieve 68.9 mAP. In 2021, YOLOX was introduced by Zheng et al. [51], which achieved 69.6 mAP with the MS COCO dataset. YOLOv3 uses darknet to train the model. The darknet has originally 53 layers. In YOLOv3, another 53 layers are added to the darknet for detection, making 106 layers of fully convolutional architecture. Figure. 4.7 shows the YOLOv3 architecture. YOLOv3 [34] gives the best results compared to other neural networks. The previous neural networks used region-based convolutional neural networks, which require thousands of evaluations to predict an object from an image. On the other hand, YOLO only passes the image once to the neural network, that is why it is called "You Only Look Once". YOLOv3 has five layer types in general; they are: "convolutional layer", "upsample layer", "route layer", "shortcut layer", and "yolo layer". Table. 4.3 shows the parameter list and values used in the YOLO v3 model.

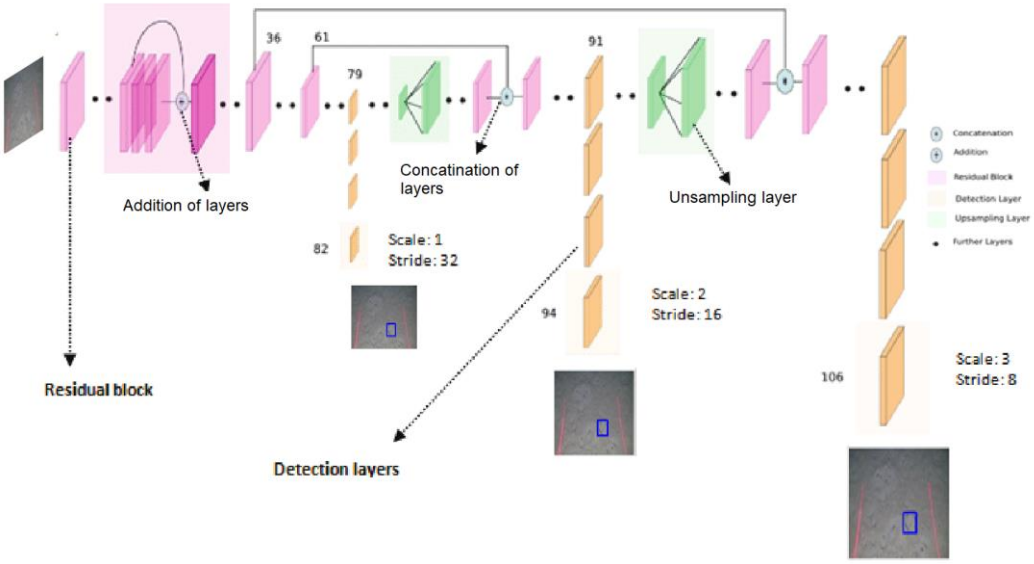


Figure 4.6: YOLOv3 architecture

Table 4. 4: YOLO v3 Model Training Parameters

Parameters	YOLO v3
Number of Classes	01
Train Batch Size	4
Train Input Size	416
Train_Transfer	True
Batch_Norm_Decay	0.9
Batch_Norm_Epsilon	1e-05
Activation Function	Leaky RELU
Train Epochs	100
stride	1

4.3.3. Model Training Environment

This work conducts model training, validation, and testing on a Linux Machine powered by an NVIDIA TitanXP GPU. Transfer learning [37] is utilized to fine-tune the models in TensorFlow [52]. Multiple combinations for model training are used, i.e., trained separate models for Cadiz and Ireland datasets, training a model by combining both the datasets and training and testing the model with different datasets. For FU 30, 200 images are used, and for FU 22, 619 images are used for training the model. The Inception and MobileNet models used two classes (one for *Nephrops* and one as background) L2 regularization for training and were trained with 70k steps. MobileNetv2 is two times quicker in training as compared to the Inceptionv2 model.

Precision can be seen as how robustly the model identifies *Nephrops* burrows' presence, and Recall is the rate of TP over the total number of positives detected by the model [53]. Generally, when the recall increases, the precision decreases, and vice versa, so precision vs. recall curves P(R) are valuable tools for understanding model behaviour. The mean average precision (mAP), defined in Eq. (1), is used to quantify how accurate the model is with a single number.

$$mAP = \int_0^1 P(R)dR \qquad (1)$$

In our problem, ground truth annotation and model findings are rectangular areas that usually don't fit perfectly. In this paper, it is considered a TP detection if both areas overlap more than 50%. This is computed by Jaccard index J,

defined in Eq. (2)

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|'} \quad (2)$$

A and B are the set of pixels in the truth annotation and model finding rectangular areas, respectively, and | . | means the number of pixels in the set. When J ≥ 0.5, a TP is detected, but if J < 0.5, detection fails with an FN. This methodology calculates P and R values, and mAP is used as a single number measure of the model's goodness. Usually, this parameter is named mAP50, but in work, it used mAP for simplicity.

4.3.4. Models Validation

Models were trained using a random approximately 70-75% sample of the annotated dataset. The remaining is used for testing. The turning checkpoints are recorded after every 10k iterations, and the mAP50 is computed on the validation dataset. The model is evaluated using mAP, precision and recall curve, and visual inspection of the images with automatic detections. Figure. 4.8 shows the model evaluation life cycle.

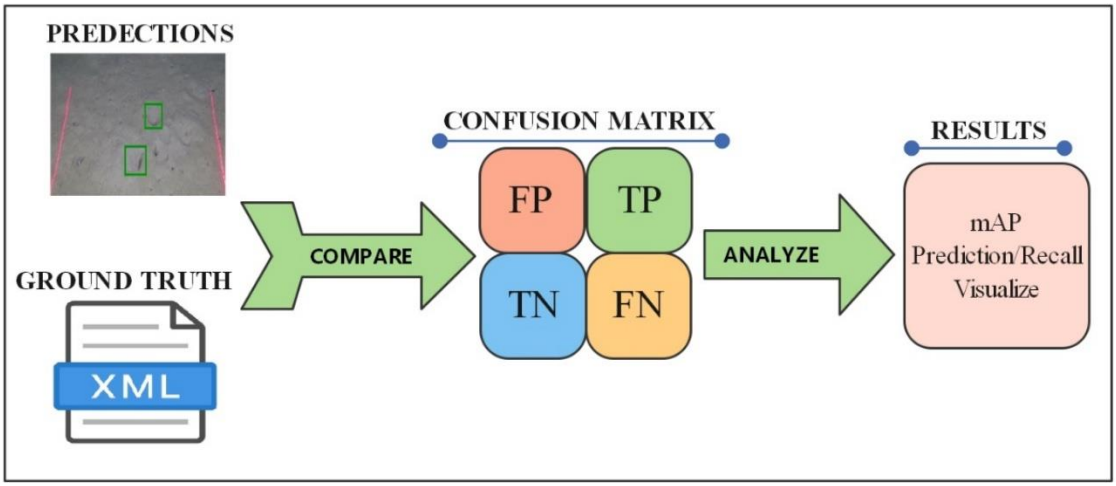


Figure 4.7: Model Evaluation Life Cycle.

This page intentionally left blank.

Chapter 5: *Nephrops norvegicus* Burrows Detections Refinement and Counting

5.1.Introduction

This chapter significantly improves the detection of *Nephrops* burrows, tracking and counting of *Nephrops* burrows. With the evolution of the convolutional neural network (CNN), object detection in the underwater environment has gained much attention. However, due to the complex nature of the underwater environment, generic CNN-based object detectors still face challenges in underwater object detection. These challenges include image blurring, texture distortion, color shift, and scale variation, which result in low precision and recall rates. A detection refinement algorithm is proposed in this chapter to tackle this challenge. The algorithm is based on spatial–temporal analysis to improve the performance of generic detectors by suppressing false positives and recovering the missed detections in underwater videos.

Object counting is also one of the biggest challenges in vision problems. One of the major causes of these challenges is the nature of an object, like its shape, size and movement in the underwater environment. This chapter presented the challenges faced while tracking *Nephrops* burrows and a new tracking and counting mechanism based on the spatial-temporal values of the *Nephrops* burrows.

This chapter is mainly divided into two sections. Section 5.2 details the detection refinement mechanism, the proposed algorithm and its working. Section 5.3 is about the tracking and counting of *Nephrops* burrows. This section describes the tracking and counting challenges, background study and proposed tracking and counting algorithm.

5.2. *Nephrops norvegicus* Burrows Detection Refinement

The proposed detection refinement mechanism is based on spatial–temporal information to enhance the detection of missed true positives and suppress false positive detections. The work presented in [54] used temporal information to track the faces and suppress false positive detections. Their approach used low-level tracking to detect the faces in real images. Furthermore, their system does not recover the missing detections. In this problem, low-level tracking cannot be applied due to the complex

underwater data nature, and the object characteristics vary with each appearance. The *Nephrops* burrows are not a real species, and their characteristics differ significantly from the natural image. The previous work integrates temporal information to track the faces and suppress the false positives. The proposed approach uses spatial and temporal information to suppress the false positives and recover the missed detections. The work is divided into two parts. At first, the model is trained using state-of-the-art Faster RCNN models Inceptionv2 [30], ResNet50 [32], and ResNet101 [33] for the detection of *Nephrops* burrows. The work's second part presents a spatial–temporal-based detection refinement algorithm. The burrows are detected in each frame in a video sequence, and then the spatial and temporal information across the multiple frames to refine the *Nephrops* burrows detections. The spatial–temporal mechanism helped in suppressing the FP burrows. It allowed us to find the missed TP detection that led to achieving better accuracy and tracking and counting burrows in a video sequence. Figure 5.1 shows the result of the detectors trained using the Inception model. In Figure 5.1(a), the bounding boxes in blue show the ground truth, while the red bounding boxes show the detections from the Inception model. Due to variations in camera direction and the appearance of burrows, the detector accumulates FPs and missed detection in some frames. The figure clearly shows the missed detection in the intermediate frames. Figure 5.1 (b) shows the ground truth bounding boxes in blue color while the false positive detections in orange color.

To address these challenges, a detection refinement approach based on spatial–temporal analysis is proposed that enhances the mAP of a generic detector. The proposed detection refinement mechanism identified these missed detections, recovered them, and suppressed the false positives. Generally, the proposed approach has the following contributions:

i. The first contribution is the spatial–temporal filtering (STF) model that extracts the spatial and temporal information of all the detections of the consecutive frames of an input video by suppressing the false positives and recovering the missed detections. The proposed method will improve the performance of the generic detectors (such as Inception and ResNet, in our case).
ii. The 2nd contribution is the performance evaluation of the proposed framework on our proposed novel dataset. The experimental results demonstrated the effectiveness of the proposed approach.

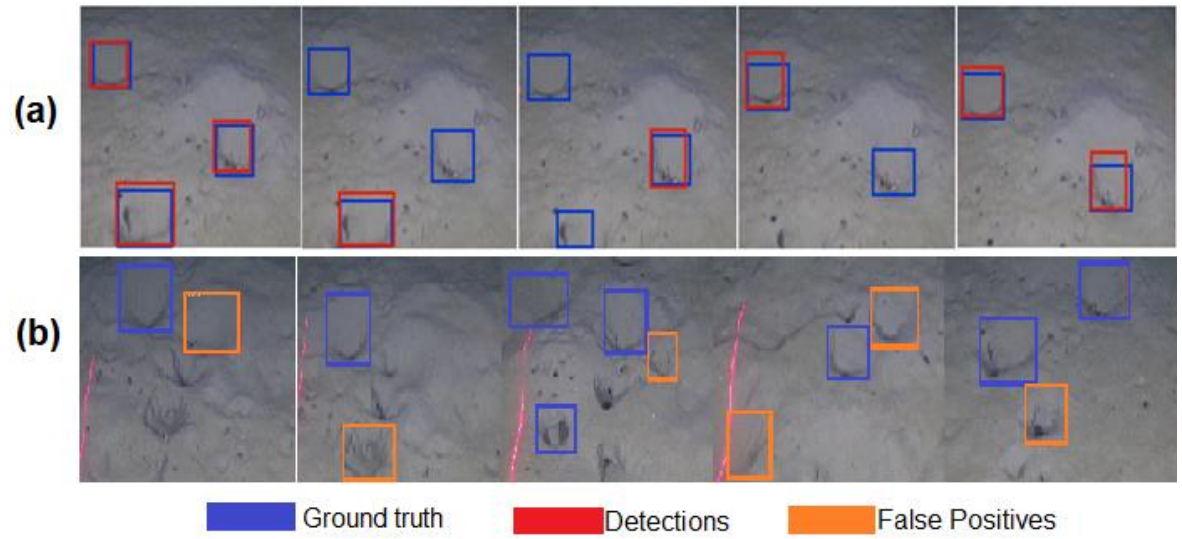


Figure 5.1: Ground truth (blue color, bounding boxes). (a) The result of detector (Inception) (red color, bounding boxes). Shows missing detections in consecutive frames. (b) FP Detections (Orange color, bounding boxes) shows FP detections in random frames

5.2.1. Detection Refinement Methodology

Figure 5.2 shows the pipeline of the proposed framework. The proposed framework has two sequential stages. The first stage is object detection, while refinement is performed during the second stage. During the first stage, state-of-the-art generic detectors, for example, Faster RCNN, Inception, ResNet50, and ResNet101, are used to detect the *Nephrops* burrows. For this purpose, the input video sequence is divided into temporal segments, each consisting of N frames. The state-of-the-art detectors are applied to each temporal segment to detect *Nephrops* burrows. The obtained results are passed to the refinement module that will employ spatial–temporal filtering (STF) to recover the missed detections from the frames and suppress the false positive detections. This process improves the mean average precision (mAP) of the results obtained from the detectors.

5.2.2. Detection Refinements Parameters

After detecting *Nephrops* burrows, a post-analysis performance of the obtained results is carried out. After a critical analysis of the results, it has been observed that the detectors encounter many FP and miss many TP, which degrades accuracy. To recover missed detections and suppress FP, the work proposes a detection refinement algorithm that exploits the spatial–temporal information among consecutive frames of the given temporal segment. The Inception, ResNet50, and ResNet101 models are tested on a video of five minutes in length. The proposed detection refinement algorithm takes inputs V, λ, and W,

where V is the video, λ, is a threshold value for displacement vector, the threshold value is the value of IoU (intersection over union) that is compared later with the IoU of detected *Nephrops* burrow, and W is a size of the temporal window which determines the number of frames in the temporal window. These models provide a set of TP, FP, and missed detections. The criteria for defining TP and the FP in the proposed detection algorithm are discussed in the next sections.

True Positives (TP)

The algorithm considers every detection as a TP if it is continuously detected by the detector within the temporal window and its average IoU in all the frames in the temporal window is more than or equal to the threshold value λ. Therefore, if the detector marks any FP detection as TP and the detection continues to occur in all consecutive frames, our algorithm considers it a TP detection.

False Positives (FP)

The FP detections are those not detected in consecutive frames, and their combined IoU is less than the threshold value λ. These FP detections are also declared as FP in the ground truth dataset. The detectors detect them as TP because of the camera angle (45°) and the position and angle of the burrow.

Missed Detections

The missed detections are those which are TP and are detected in some frames by the detector but missed in some intermediate frames due to the position or visibility of the burrow. The missed detections are very important to identify because without identifying them, the burrows cannot be tracked. The performance of models is increased by recovering the missed detections.

Detection Refinement Algorithm

The proposed algorithm presented in Algorithm 5.1 shows the refinement mechanism using spatial-temporal data analysis. This algorithm is divided into two sections, i.e., suppression of false positives and identification of missed detections. Figure 5.2 shows the basic processing steps of false positive suppression and missed detection identification and recovery.

Suppression of False Positives

The first step towards the refinement of detections is to suppress the FP. Let Fi = {B1, B2,…, Bn} be the frame i with n detections obtained with a deep learning model . Let sF be the set of consecutive frames within a temporal window with size W. The algorithm takes Bj for frame Fi as an input for

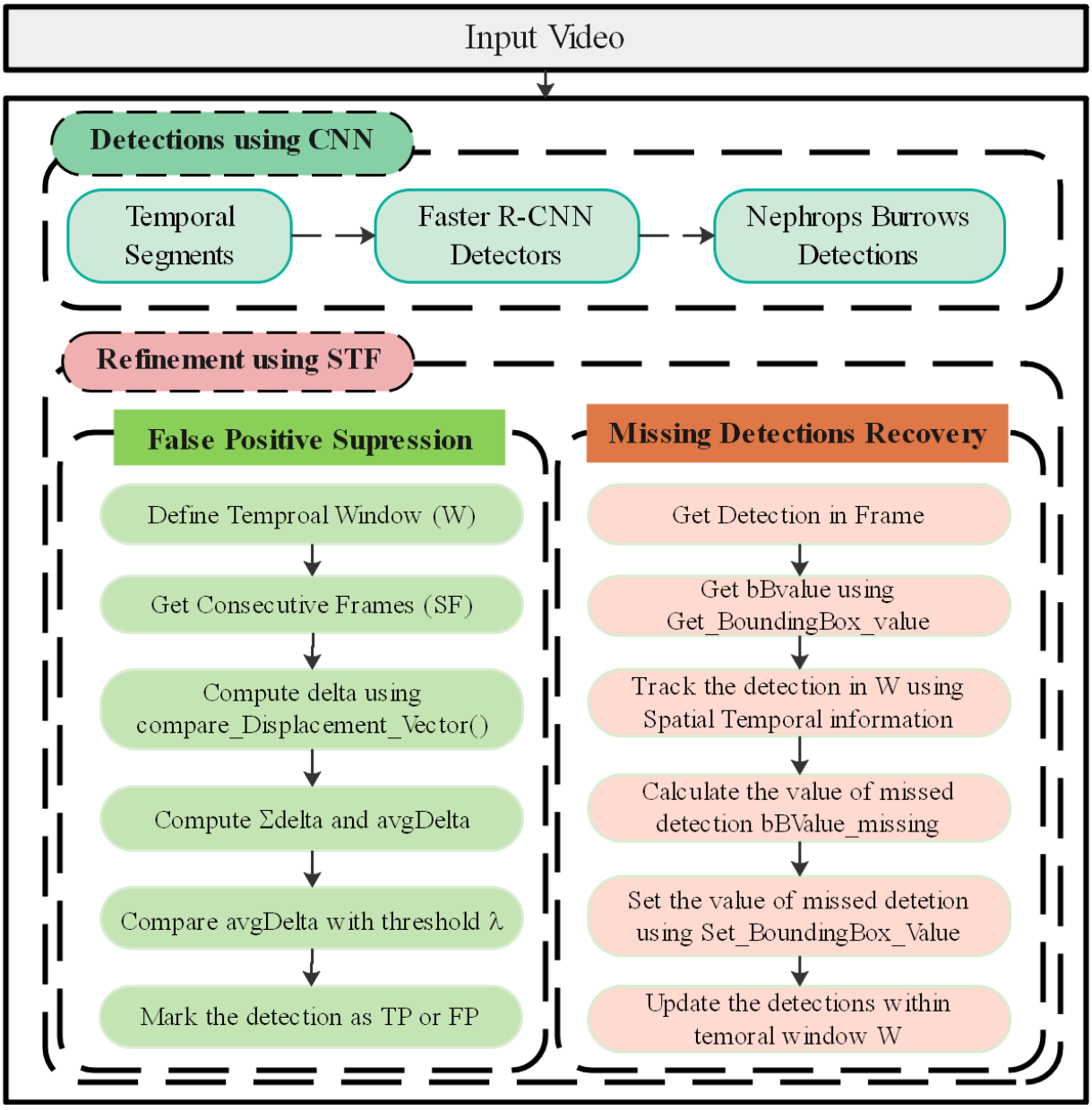


Figure 5.2: Detection refinement algorithm

refinement and provides a refined output as FR. To suppress the FP in the current frame i, the work computed the overlapping of each detection Bj of the current frame and the detection in the next frame from sF.

Algorithm 5.1: Detection Refinement

Input Data V, λ, W, where V is an input video, λ is a threshold value for the displacement vector, W is the size of the temporal window

Results $F_R = \{F_1, F_2, ..., F_n\}$, where F_R is a list of refined frames

begin

F = Extract_Frames_With_BoundingBox(V) // $F = \{F_1, F_2, ..., F_n\}$ where F is the list of frames and each one $F_i = \{B_1, B_2, ..., B_n\}$ has n bounding boxes $B_j = \{x_j, y_j, w_j, h_j\}$, where (x_j, y_j) are coordinates of initial pixel of the bounding box j and w_j, h_j are width and height.

T = Extract_Duration(V) // $T = \{T_1, T_2, ..., T_n\}$ *where T is total time of the video*

Foreach *frame* $f \in$ F **do**

F_R = Add_Frame(f)

sF = Create_Subset_Frame_W_Range(F) // sf is list of frames that need to compare with the current frame till the 'W' temporal window size

deleteFlag = Set (FALSE)

Foreach *boundingbox* $b \in$ f **do**

b_Index = Get_Bounding_Box_Index(b)

Foreach *frame* $fc \in sF$ **do** //*where* $fc = f+1$

delta += Compare_Displacement_Vector (f_{b_Index}, fc_{b_Index})

endFor

avgDelta = delta/W

if avgDelta < λ **then**

deleteFlag = Set (TRUE)

endif

if deleteFlag is FALSE **then** F_R= Add_ Bounding_Box_in_Frame (f, f_{b_Index})

endif

endFor

Foreach *boundingbox* $b \in$ f **do**

indexSet = Identify_Missing_Detection(b, F_R)

endFor

lastIndex = 0

for index in F_R do

if index is in indexSet

```
            j = index
        For lastIndex to j
                bBValue += Get_BoundingBox_Value(b, f lastIndex)
            endFor
            bBValue_missing = bBValue/j
      Set_BoundingBox_Value(b, fj, bBValue_missing)
            lastIndex = j;
                        endif
                    endFor
                endFor
        return FR
end
```

The algorithm receives three inputs: an input video with detections V, threshold value λ, and temporal window size W. For each detection in the current frame b ∈ Bj at frame Fi, the algorithm first identify the current detection location in the next frame of sF and then computes δk = IoU value of current detection with consecutive k frame's detection in sF using Compare_Displacement_Vector(fb_Index, fcb_Index) method (k = 1,…,W). Then, δavg = 1/W ∑δk is the estimated average within the temporal window. The detection is marked as FP if δavg < λ, and as TP if otherwise, suppressing the FP. The work processes the whole video V detections in the same way.

Identification of Missed Detections

After refining the detections by suppressing the FP in the previous step, the next step is identifying the detections that our detector missed. For this purpose, the algorithm tracks each detection Bj ∈ Fi to identify the missed detection. If the detection is found in frame i + 1, the algorithm continues to track it till the temporal window size W. If the current detection is not tracked in any frame, the algorithm marks that as missed detection and stores it in the set indexSet. The algorithm defines the Set_BoundingBox_Value( ) method to calculate the missed detection value. The location of the missed detection is first computed from the indexSet. Letting Bj be the current detection and indexSetj the missed detection, the algorithm calculates the accumulative value of detection from the current frame till the indexSet location and then calculates the average, called bBValue_missing. As the work maintains the number of frames N between the

current and missed detection, the algorithm calculates the missed detection value by adding the N value to the bBValue_missing. The missed detection information is then filled in and updated in the refined output FR.

5.3. *Nephrops norvegicus* Burrows Tracking and Counting

Object counting is also one of the biggest challenges in vision problems. One of the major causes of these challenges is the nature of an object, like its shape, size and movement in the underwater environment. The other major factor is the underwater environment. The underwater environment has a complex background structure, poor visibility, the turbulence of water, and the complex seabed, which causes the object counting a complex and challenging problem. With the advancement of AI, many automated tools and mechanisms are available to count objects in underwater and ground objects. In underwater environments, the counting algorithm used a regression-based [84] or detection-based [85] approach to count the objects. There are many tracking techniques used in the literature which are based on the OpenCV KCF tracker [86], Optical flow [87], SORT [88], or Kalman Filter [89].

The techniques proposed for tracking underwater objects perform well with the objects in motion, like fishes and other underwater species. In the current problem, three major challenges are faced while tracking the *Nephrops* burrow. The first challenge is the camera's movement; our objects are not moving, but the camera is moving in the forward direction, leaving the object behind. The second challenge is the characteristics and size of burrows that are not fixed, and each new burrow can vary in size and other characteristics. The third challenge is the angle/opening of the burrow. Each burrow opening can vary in direction, and the angle of the burrow can also change. Due to these challenges, the traditional object-tracking mechanism is not very effective.

This work proposes an efficient tracking technique based on the object's spatial and temporal values. The proposed tracking and counting of *Nephrops* burrows used the spatial-temporal values of each burrow. The proposed spatial-temporal technique tracks each burrow based on its spatial and temporal values and counts the unique burrows. The unique burrows are counted using the intersection values of detected burrows in consecutive frames. The proposed methodology is a three-step process starting from Data collection and processing, Detection and Counting of burrows.

5.3.1. Background and related work

Object detection and classification is a challenging computer vision problem. Researchers have developed many methods for object detection and classification tasks. The existing object detection approaches use handcrafted feature-based models [90-93] and deep features models [94]. There are many approaches presented for underwater object tracking that are very useful in tracking and counting objects having concrete features and moving in the water.

Object counting is also one of the biggest challenges in vision problems. One of the major causes of these challenges is the nature of an object, like its shape, size and movement in the underwater environment. The other major factor is the underwater environment. The underwater environment has a complex background structure, poor visibility, turbulence of water, and the complex seabed, which causes the object counting a complex and challenging problem. The counting algorithms used regression-based [95-100] or detection-based [101-103] methods to count the underwater objects. The Regression-based method generates a density map for images and later integrates it with the image maps to count the objects. Most regression-based counting algorithms are useful in counting objects from a single image. The detection-based methods use two-stage detectors [104] or single-stage [105-106]. The two-stage detector, also called a sparse detector, uses two steps for image detection. The first step generates the boxes in the image using the region proposal network, while the second step evaluates these proposals and generates the detections. Some of the most popular two-stage detectors are RCNN [28], Fast RCNN [29], Faster RCNN [28], SPPNet [107] and Pyramid Networks [108]. On the other hand, the single-stage detectors used single-shot or dense-shot architectures. This type of detector processes the image only once and uses the feature pyramid networks to detect the objects. Some of the state-of-the-art single-stage detectors are OverFeat [109], SSD [110], YOLO [102], Retina-Net [111] and Effiient-net [112]. The YOLO detector was introduced in 2016 with its first version, YOLO v1, by Redmon et al. [102], achieving 63.4 mAP on the Pascal VOC data set. The second version, YOLOv2 and YOLO900, was introduced in 2017 by Redmon et al. [47] and achieved 78.6 mAP on the same data set used in YOLOv1. In 2018, Redmon et al. [34] introduced YOLOv3, which used a complex data set of MS COCO, which had 80 classes and achieved 57.9 mAP. In 2020, multiple versions of YOLO appeared from different authors that used the same MS COOC data set and achieved a higher mAP as compared to YOLOv3; they listed as YOLO v4 by Bochkovskiy et al. [113] achieved 65.7 mAP. Scaled YOLO v4 by Chein et al. [114] achieved 66.2 mAP, PP-YOLO from Xiang et al. [115] achieved 65.2 mAP and YOLO v5 by Ultralytics achieved 68.9 mAP. In 2021, YOLOX was introduced by Zheng et al. [116], which achieved 69.6 mAP with the MS COCO dataset.

Object Counting in Images

In literature, people used many methods to automatically count marine objects. In a recent survey by Li et al. [117], they present three ways based on sensors, vision and acoustic technology for counting underwater objects. For

computer vision methods, people proposed counting methods for still images and videos.

In underwater images, object segmentation is one of the most used techniques for object counting. It is an important method to differentiate the object of interest from the background based on its intensity value. Solahdin et al. [118] and Labuguen et al. [119] adopted threshold-based segmentation methods to count the shrimps and fishes. Jing et al. [120] proposed an edge detection-based Sobel Operator to detect fish's edges to obtain the fish count estimation. The detection-based method is the other common method used in detecting and counting underwater objects. The key point in this method is to select the appropriate classifier to identify and detect the objects accurately. Culverhouse and Pilgrim [121] used an Artificial Neural Network to count the fish from the underwater images. Fan et al. [122] used Background Propagation Neural Network (BPNN) to classify the number of fish for counting. Object detection methods also used some hand-crafted features for object classification and counting.

Object Counting in Videos

Counting the objects from underwater videos is more challenging due to varying framerates and background changes. Lau et al. [123] presented segmentation and SVM-based detection and tracking methods to count the Norway lobster. Sharif et al. [124] used Kalman and Hungarian methods to count the fish from underwater videos on Shutterstock. Chuang et al. [125] presented a multi-object tracking algorithm based on the deformable multiple kernel method to track and count fishes from multiple locations and habitats. After the evolution of deep learning, the tracking and counting mechanism became more efficient. Huang et al. [126] presented a combination of deep learning and a 3D Kalman filter to count the fish from underwater stereo images captured from stereo cameras. Spampinato et al. [127] combined the blob-shape features and histogram matching to track the fish. They used the moving average algorithm to achieve an accuracy of up to 85%.

Modasshir et al. [86] presented a mechanism to identify and count the corals. They used Retina-Net [111] to identify and localize the coral samples from the dataset. For tracking corals, they used the OpenCV KCF tracker [129]. Mohammed et al. [130] proposed a fish farm monitoring system to detect, track and count the fish. They used YOLOv3 to detect the fish in the farm and an optical flow algorithm to track the fish movements in each frame using the fish trajectories. Wageeh et al. [131] presented a method to detect the fish from the

fish farm. They also count and make trajectories from fish detections. The proposed method is used for monitoring fish farms using the combination of fish counting and trajectories. They used distance to calculate the distance measured between the fishes in consecutive frames and track them according to the distance between them. Li et al. [132] proposed an adaptive multi-appearance model and tracking strategy for real-time fish tracking. Tanaka et al. [133] presented a fish tracking and counting method used to count the fish on the deck. They trained Yolov3 with 13789 different images of fish for detection. The fish counter proposed in their approach is based on the Simple Online Real-time Tracking (SORT) [134], which uses the Kalman filter to approximate the displacement of the fish in consecutive frames. Their approach detects and tracks the fish. Later, they apply the post-processing algorithm to suppress the false positives. Gaude et al. [89] proposed a method to track the fish in a varying turbidity environment. Kalman filter is used to track fish. Avsar et at. [137] proposed a deep learning-based detection and counting mechanism for Nephrops norvegicus individuals. They used the SORT algorithm for tracking and counting individual Nephrops. Table 5.1. Shows the summary of some tracking algorithms that people apply to track objects in the underwater environment.

Object Tracking Algorithms

OpenCV also provides multiple tracking algorithms that perform very well in tracking the objects in the videos. These algorithms are very fast compared to the detection algorithms. Here, the work presented all the OpenCV tracking algorithms and found out the pros and cons of these algorithms. Also, the current work will apply these algorithms to track the *Nephrops* burrows and will show the results in detail.

- BOOSTING Tracker

The boosting tracker works on the HAAR cascade-based detector. This tracker trained itself on the runtime based on the initial values of the bounding box provided on the first frame. This very old algorithm works fine in tracking objects, but the tracking usually fails with this tracker.

- MIL Tracker

MIL tracker also works on the same idea as the BOOSTING tracker, but during the tracking, it also considers the neighbourhood location of the object to get positive examples. It uses Multiple Instance Learning (MIL). The MIL tracker performs very well compared to the BOOSTER tracker, but it also leads to false positive tracking and not recovering from full occlusion.

- KCF Tracker

Kernelised Correlation Filters (KCF) tracker is the most commonly used tracker. It performs better in speed and accuracy than the previous two trackers, but it also leads to the false positive tracking of the object.

- TLD Tracker

TLD tracker used detection and learning in the tracking. It follows the object in every frame and localises it for tracking. This algorithm gives the best video results but leads to many false positives, degrading this tracker's efficiency.

- MEDIANFLOW Tracker

MEDIALFLOW tracker tracks the object in forward and backward directions, which enables this tracker to track the object more accurately. This tracker keeps track of the object and can identify when the tracking of the object fails. It works well with the object motion but fails when the camera moves and loses tracking.

- MOSSE tracker

MOSSE stands for Minimum Output Sum of Squared Error and uses adaptive correlation between objects. This robust tracker can resume tracking if the object is lost in some frames.

- CSRT tracker

CSRT tracker uses the spatial reliability map for adjusting the filter and tracking the object by localising the selected region. This tracker uses HoGs and Color names to find out the features of objects.

Some of the other famous tracking algorithms are.

- DeepSORT

DeepSORT is one of the most widely used tracking algorithms. The object with the DeepSORT is trackable for a longer period because it integrates the appearance information of the object within the tracking algorithm.

- Object Tracking MATLAB

A Computer Vision toolbox in MATLAB provides object tracking in the videos. This toolbox uses CAMShift and Kanade-Lucas-Tomasi (KLT) for tracking the object.

- MDNet

MDNet is a CNN-based tracking algorithm. It is mainly used for real-time object tracking but is highly expensive regarding computational and speed.

The method proposed in this study is to track the burrows using a spatial-temporal technique. The techniques presented in the literature cannot track the burrows accurately and provide a lot of false positives due to variations in the angle and burrows characteristics.

Table 5.1: Comparative analysis of a few Tracking techniques in an Underwater environment

Approach	Year	Date set	Detection Algorithm	Tracking Algorithm
Identification and Counting of Coral [86]	2018	-	RetinaNet	OpenCV KCF tracker
Detection and Fish Tracking [130]	2020	400 goldfish images	YOLOv3	Optical flow
Detection and fish Tracking [131]	2021	2000 images of golden fish	YOLOv3	Optical flow
Fish Tracking and Counting [133]	2022	13789 images of fishes	YOLOv3	SORT
Fish Tracking [89]	2019	Custom dataset	Hybrid Algorithm	Kalman Filter
Neprophs Tracking [137]	2023	DTU repository	YOLOv4	SORT

5.3.2. Burrows Tracking and Counting Methodology

The proposed methodology is presented in Figure. 5.3. The data is collected through the annual WGNEPS survey. The collected data is passed through a preprocessing stage to the trained model. The proposed spatial-temporal algorithm is run with the trained model to track the detected burrows and count the unique burrows.

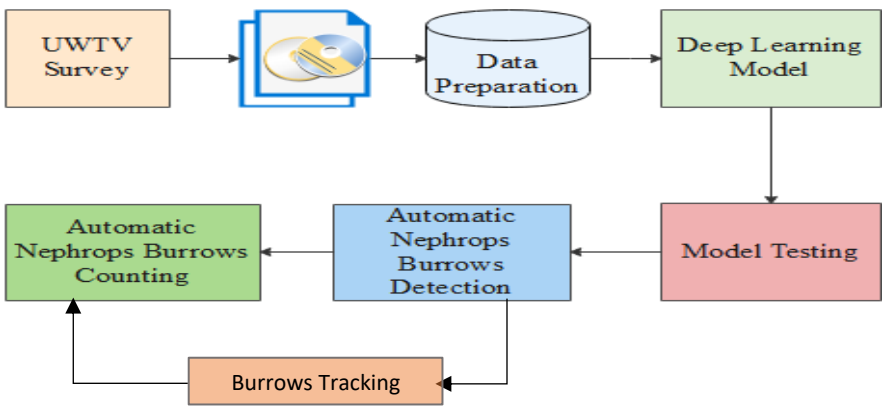


Figure 5.3: Burrows Tracking and Counting Methodology

5.3.3. Tracking and Counting of Burrows

The proposed algorithm used the spatial and temporal values of the object for tracking and counting. The proposed spatial-temporal technique tracks each burrow based on its spatial and temporal values and counts the unique burrows. The unique burrows are counted using the intersection values of detected burrows in consecutive frames. The proposed methodology is a three-step process starting from Data collection and processing, Detection and Counting of burrows.

The proposed algorithm, presented in Algorithm 5.2, shows the tracking and counting of burrows using the spatial-temporal value of each burrow. The tracking algorithm runs in parallel with the detection algorithm. Here, the work used the YOLOv3 model for the detection of burrows.

Tracking and Counting Algorithm

The algorithm receives two inputs: an input video V, and a threshold value λ (an overlap amount between predicted and ground truth detections). The algorithm's output will be the unique count of burrows in the given video. The first step is to detect the burrows in the video frames. The input video is passed to the method Detect_*Nephrops*_Burrows (*V*). The method converts the video into frames and uses a YOLO v3 detector to detect the burrows in each frame. The method's output is an individual frame (*I*) having a set of detected *Nephrops* burrows with spatial values. The spatial value of each detection is the bounding box values {*x, y, w, h*}, where (*x, y*) are coordinates of an initial pixel of the bounding box *j* and *w, h* are width and height. For each detection in the current frame $f \in I$ at frame I_i, the algorithm loops through each detection of the current frame and gets the current spatial value using the method Get_Spatial_Value(b). The current detection identified by the algorithm is stored in Indexfb and added to the list of burrows count *N*. The current detection is marked with a flag, and the algorithm continues to mark each detection of the current frame. Once all the detections of the current frame are marked, the algorithm saves the detections with their spatial values to $Index_{(f-1)b}$ and moves to the next frame. In the next consecutive frame, the algorithm again identifies the detection using Get_Spatial_Value(b). Now, each detection of the current frame is tracked by comparing with the previous frames detections stored in $Index_{(f-1)b.}$ The Compare_Overlapping ($Index_{fb}$, $Index_{(f-1)b}$) method compares the bounding box values of two detections. For comparison, this method did not use the traditional overlapping metric IoU because of variation in the position of a detected burrow in each frame due to the camera's movement. The overlapping method is modified in this

algorithm, and instead of calculating the IoU, the algorithm calculates the *Intersection* value of each comparison. This compared value is stored in the variable delta and is compared with the given threshold value λ. If the delta is greater or equal to the λ value, the same burrow is detected again and is not counted. Otherwise, the counter list of that frame is updated with a new burrow count. The work processes the whole video V detections in the same way. In the end, each counter value of frames is accumulated and returns the unique number of burrows.

Algorithm 5.2: Tracking and Counting

Input Data V, λ where V is an input video, and λ is a threshold value for object overlapping.

Results $N = \{N_1, N_2, ..., N_n\}$, where N are the unique objects, N_C is the count of unique burrows.

Begin

I = Detect_Nephrops_Burrows(V) // $I = \{I_1, I_2, ..., I_n\}$ where I is the list of frames and each one $I_i = \{B_1, B_2, ..., B_n\}$ has n bounding boxes and each box $B_j = \{x_j, y_j, w_j, h_j\}$, where (x_j, y_j) are coordinates of an initial pixel of the bounding box j and w_j, h_j are width and height.

count = 0

Foreach *frame* $f \in I$ **do**

Foreach *boundingbox* $b \in f$ **do**

$Index_{fb}$ = Get_Spatial_Value(b)

if (flag)

delta = Compare_Overlapping ($Index_{fb}$, $Index_{(f-1)b}$)

if delta < λ **then**

N_{fb}++

endif

endif

endFor

N.add(N_{fb})

flag = true

endFor

return N

Chapter 6: Experiments and Results

6.1. Introduction

This chapter summarizes all the experiments and results performed for *detecting, refining, tracking and counting Nephrops* burrows. Many experiments were performed on the dataset to evaluate the performance of the models and proposed work. This chapter is divided into three main sections. The first section discusses the experiments performed for burrows detections. The second part contains the detection refinement experiments, and the last section discusses the experiments and results of tracking and counting Nephrops burrows.

6.2. Experiments and Results of Nephrops Burrows Detection

6.2.1. Experiments

This section presents the performance of different networks in qualitative and quantitative ways. To detect the *Nephrops* burrows automatically, multiple experiments are performed. In this work, the models are trained on three datasets. The first dataset purely contains FU 30 images. The second dataset contains the images from the FU 22 dataset, and the third is the hybrid dataset containing images from both functional units. The dataset details and combination of training and testing data are shown in Table 6.1.

Table 6.1: Combination of Dataset for Training and Testing

Dataset	**Training**	**Testing**
Dataset-I	FU 30	FU 30
Dataset-II	FU 22	FU 22
Dataset-III	FU30+FU22 (Hybrid)	FU30+FU 22 (Hybrid)
Dataset-IV	FU 30	FU 22
Dataset-V	FU 22	FU 30
Dataset-VI	Hybrid	FU30
Dataset-VII	Hybrid	FU22

Thirty different combinations of sets of experiments are performed with varying models of training. Each set is iterated seven times. So, 210 experiments were carried out. The details of the experiments are shown in Table. 6.2. The YOLOv3 is trained and tested only with the FU 30 and FU 22 datasets separately.

Table 6.2: Experiments details for Detection

Experiment	Model	Training Dataset		Testing Dataset	
		Station	Images	Station	Images
Experiment-1	**MobileNet**	FU 30	200	FU 30	48
Experiment-2		FU 22	618	FU 22	359
Experiment-3		FU 30	200	FU 22	150
Experiment-4		FU 22	618	FU 30	200
Experiment-5		Hybrid	818	Hybrid	407
Experiment-6		Hybrid	818	FU 30	200
Experiment-7		Hybrid	818	FU 22	359
Experiment-8	**Inception**	FU 30	200	FU 30	48
Experiment-9		FU 22	618	FU 22	359
Experiment-10		FU 30	200	FU 22	150
Experiment-11		FU 22	618	FU 30	200
Experiment-12		Hybrid	818	Hybrid	407
Experiment-13		Hybrid	818	FU 30	200
Experiment-14		Hybrid	818	FU 22	359
Experiment-15	**ResNet50**	FU 30	200	FU 30	48
Experiment-16		FU 22	618	FU 22	359
Experiment-17		FU 30	200	FU 22	150
Experiment-18		FU 22	618	FU 30	200
Experiment-19		Hybrid	818	Hybrid	407
Experiment-20		Hybrid	818	FU 30	200
Experiment-21		Hybrid	818	FU 22	359
Experiment-22	**ResNet101**	FU 30	200	FU 30	48
Experiment-23		FU 22	618	FU 22	359
Experiment-24		FU 30	200	FU 22	150
Experiment-25		FU 22	618	FU 30	200
Experiment-26		Hybrid	818	Hybrid	407
Experiment-27		Hybrid	818	FU 30	200
Experiment-28		Hybrid	818	FU 22	359
Experiment-29	**YOLOv3**	FU 30	200	FU 30	48
Experiment-30		FU 22	618	FU 22	359

The MobileNet, Inception, ResNet50, ResNet101, and YOLOv3 models used 200 images from the FU 30 dataset for training the model, while 48 images were used for testing the models. Similarly, these models used 618 images from the FU 22 dataset for training and 359 for testing. Similarly, these models used the hybrid data set for training and testing by using 818 images and 407 images for testing the model.

6.2.2. Results and Analysis

Quantitative Analysis

The Mean Average Precision of all the models trained and tested by FU 22 and FU 30 stations are calculated during the quantitative analysis. The work trained all the models over 70k iterations. The models' performance is reported after every 10k iterations and achieves excellent precision on the trained dataset, as shown in Figure. 6.1.

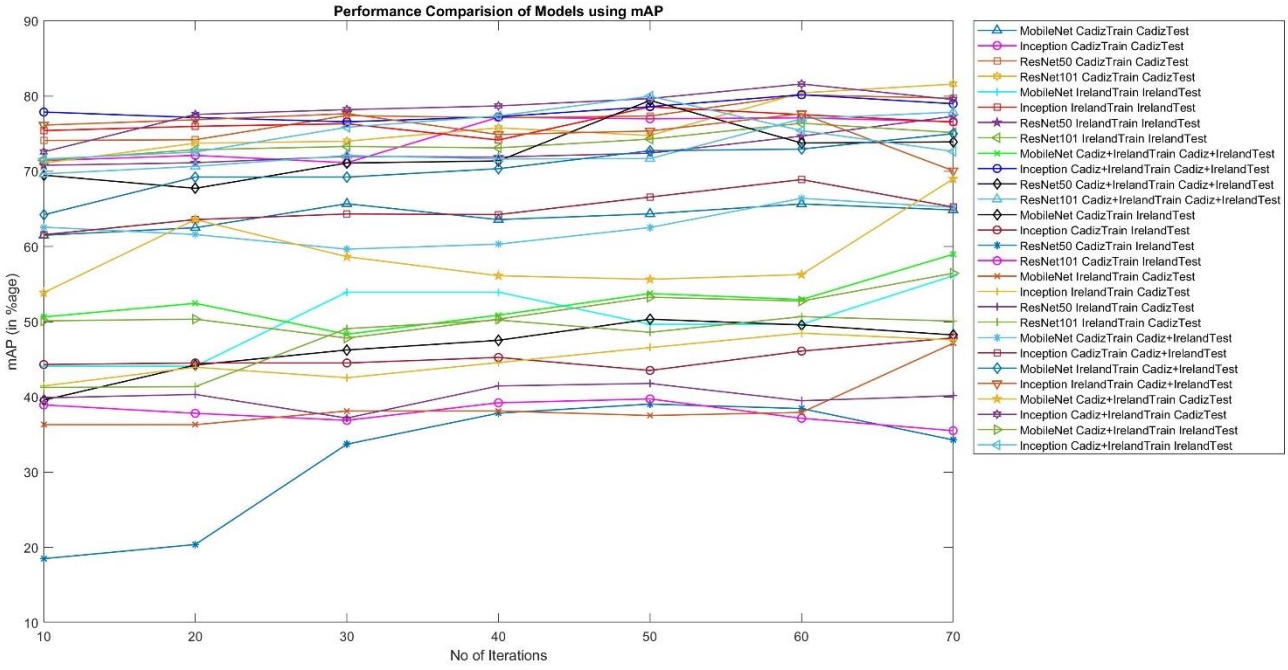


Figure 6.1: Mean Average Precision of models trained and tested by FU 22 and FU 30 stations.

The work evaluated the performance of mAP, a prevalent metric in measuring object detection algorithms' accuracy like Faster R-CNN, SSD, etc. Average precision calculates the average precision value for recall value over 0 to 1. Precision measures prediction accuracy, while Recall measures the positive predictions. The mAP is computed with the dataset of *Nephrops* from FU 22 and FU 30 stations over 70k iterations.

Tables. From 6.3 to 6.7 show the maximum mAP obtained by MobileNet,

Inception, ResNet50, ResNet101 and YOLOv3 models, respectively. Nine different training and testing combinations are used in these tables. At first, the models are trained by the FU 30 dataset and tested with the FU 30 dataset. Secondly, the models are trained using the FU 22 dataset and tested by the FU 22 dataset. In the following combination, the models are trained using a Hybrid dataset and tested by the Hybrid dataset. The fourth combination applied to the models is to train them using the FU 30 dataset and test them with the FU 22 dataset. The fifth combination is the opposite of the fourth, where the models used the FU 22 dataset for training while the FU 30 dataset for testing. Next, the sixth and seventh combinations are trained by FU 30 and FU 22 datasets and tested using the Hybrid dataset. The eighth combination used the Hybrid dataset for training and the FU 30 dataset for testing. The last combination was trained by the Hybrid dataset and tested by the FU 22 dataset.

Table 6.3 shows the maximum mAP obtained using the above dataset combination with the MobileNet model. The maximum mAP obtained using the MobileNet model is 75.12 when trained using the FU 22 data set and tested by the FU 22 data. The minimum mAP is 50.24 when the model is trained by the FU 30 dataset and tested by the FU 22 dataset.

Table 6.3: Summaries of mAP obtained using MobileNet Training Model

		Testing		
		FU 30	FU 22	Hybrid Dataset
Training	FU 30	65.69	50.24	66.14
	FU 22	57.14	75.12	56.11
	Hybrid Dataset	68.99	56.45	58.97

Table 6.4 shows the maximum mAP obtained while using the defined dataset combination with the Inception model. The maximum mAP obtained by the Inception model is 80.18 when the model is trained using the hybrid dataset and tested by the hybrid dataset. The mAP obtained was 80.18. The minimum mAP is 47.86 when the model is trained by the FU 30 dataset and tested by the FU 22 dataset.

Table 6.4: Summaries of mAP obtained using the Inception Training Model

		Testing		
		FU 30	FU 22	Hybrid Dataset
Training	FU 30	77.18	47.86	68.90
	FU 22	48.49	78.56	77.66
	Hybrid Dataset	80.16	79.99	80.18

Table 6.5 is about the ResNet50 model. This model is trained and tested by five different combinations of datasets. The maximum mAP achieved in this model is 80.16 when the model is trained and tested using the FU 30 dataset. The minimum mAP is 39.06 when the model is trained by the FU 30 dataset and tested by the FU 22 dataset.

Table 6.5: Summaries of mAP obtained using the ResNet50 Training Model

		Testing		
		FU 30	FU 22	Hybrid Dataset
Training	FU 30	80.16	39.06	-
	FU 22	41.08	77.30	-
	Hybrid Dataset	-	-	79.42

Table 6.6 summarizes the behaviour of the ResNet101 model. The maximum mAP achieved is 81.59 when the model is trained and tested by the FU 30 dataset. The minimum mAP is 41.04 when the model is trained by the FU 30 dataset and tested by the FU 22 dataset. The ResNet50 and ResNet101 models are not tested with Hybrid datasets when trained by FU 30 and FU 22 datasets.

Table 6.6: Summaries of mAP obtained using ResNet101 Training Model

		Testing		
		FU 30	FU 22	Hybrid Dataset
Training	FU 30	81.59	41.04	-
	FU 22	50.68	76.39	-
	Hybrid Dataset	-	-	77.87

In Table 6.7, the model used is YOLOv3. This model is only trained separately with the FU 30 and FU 22 datasets. The mAP achieved when trained and tested by the FU 30 dataset is 86.64, while the mAP of 78.54 is achieved when the model is trained and tested by the FU 22 dataset.

Table 6.7: Summaries of mAP obtained using YOLOv3 Training Model

		Testing		
		FU 30	FU 22	Hybrid Dataset
Training	FU 30	86.64	-	-
	FU 22	-	78.54	-
	Hybrid Dataset	-	-	-

Precision and Recall

This section shows the precision and recall curves obtained with the experiments performed for all the combinations shown in Table 6.1.

Figure 6.2 shows the results obtained with the models trained and tested by the FU 30 dataset. The best mAP is 81.59 with the ResNet101 model. The precision-recall results of all models with the FU 22 dataset are presented in Figure 6.3. The models are trained and tested by the FU 22 dataset. The maximum mAP obtained is 78.56 with the Inception model, and the minimum mAP obtained is 56.11 with the MobileNet model.

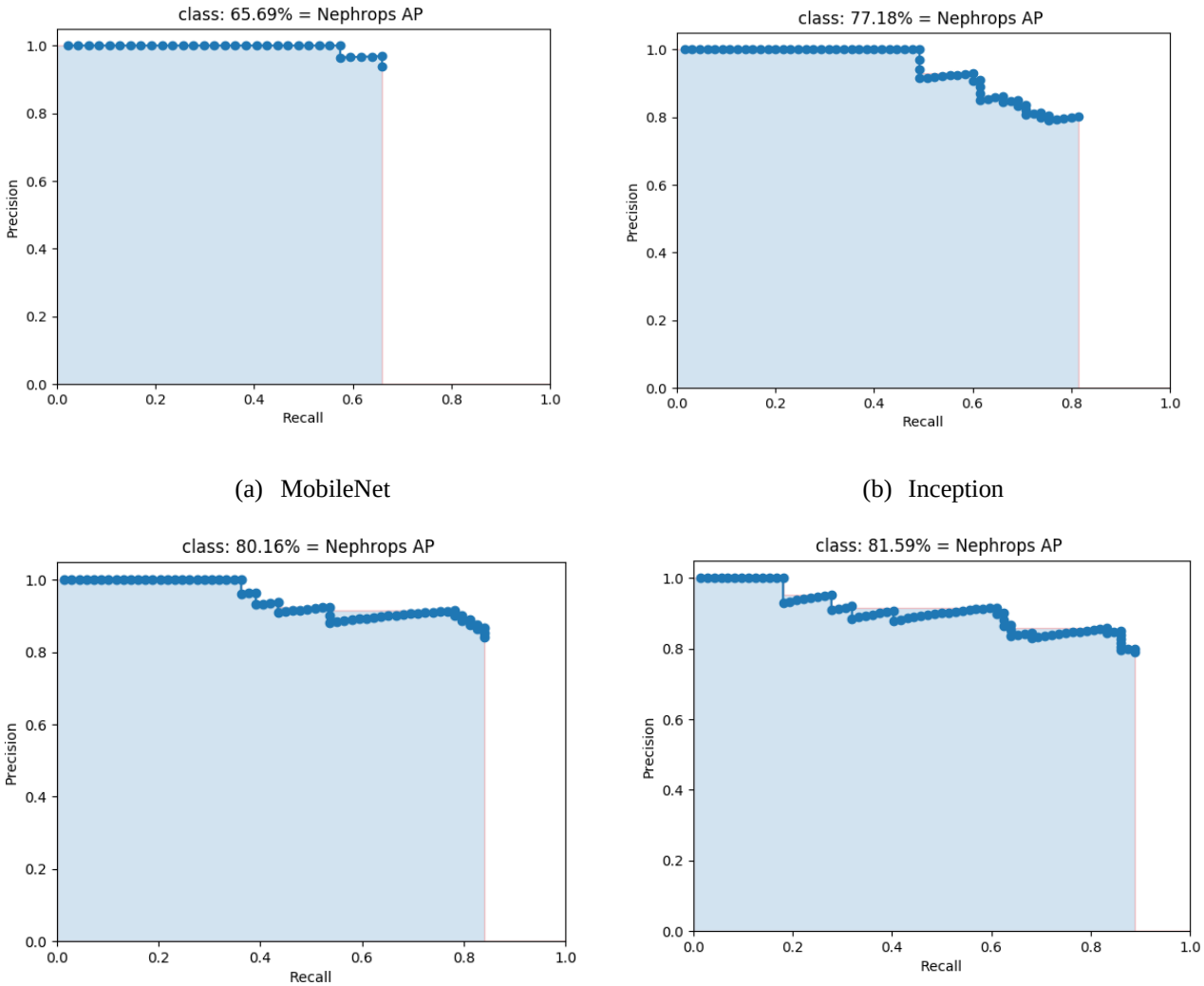
The next set of experiments is performed with the hybrid dataset. The models are trained and tested using the hybrid dataset. Figure 6.4 shows the maximum mAP obtained is 80.18 with the Inception model, and the minimum mAP obtained is 58.97 with the MobileNet model.

Figure 6.5 shows the results obtained by the models when trained by the FU 30 dataset and tested by the FU 22 dataset. As expected, the models do not perform well when trained by the FU 30 data set and tested by the FU 22 data set. The minimum value of mAP is 39.06, and the maximum is 47.86.

The models are trained by FU 22 and tested by the FU 30 dataset in the results shown in Figure 6.6. These models perform slightly better than those trained by the FU 30 dataset. The maximum mAP achieved during these experiments is 50.68, and the minimum is 38.14. The results show that the models did not perform well when trained and tested with different FU

datasets. The main reasons for low accuracy are the dataset environmental difference, burrow size variation, dataset size, lightning, and image quality.

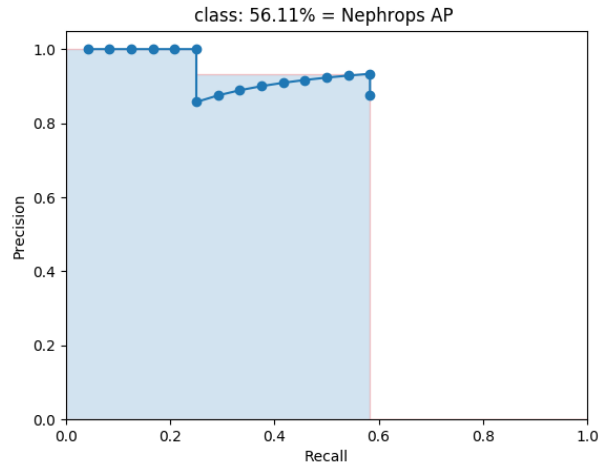
Figure. 6.7 shows the results obtained by the YOLOv3 model. The model is trained by the FU 30 and FU 22 datasets separately. The model performed exceptionally well in both cases, as the maximum mAP achieved is 86.64 and 78.54, respectively.



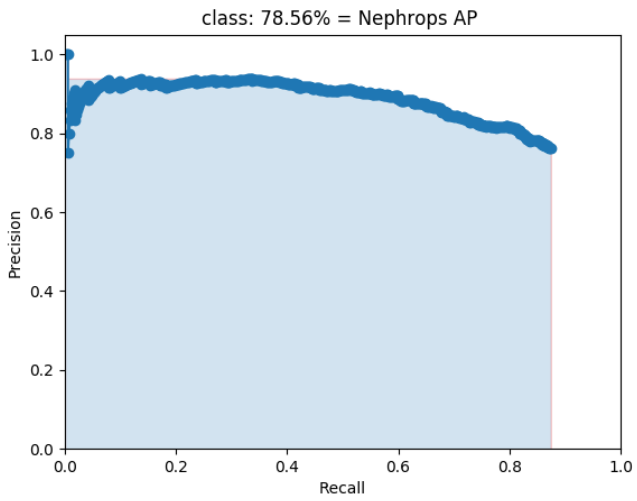
(a) MobileNet (b) Inception

(c) ResNet50 (d) ResNet101

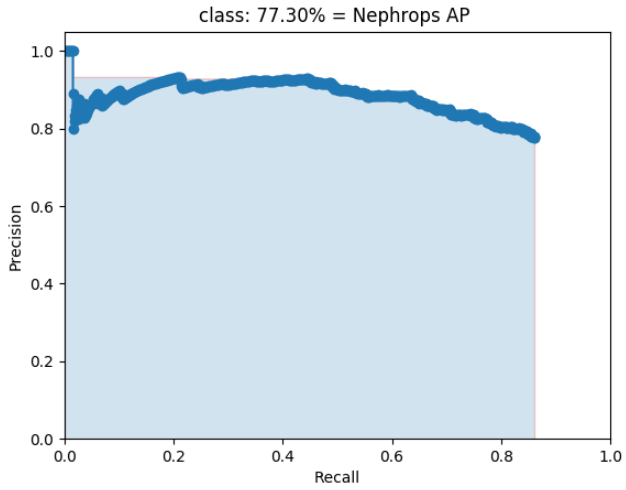
Figure 6.2: Precision-Recall curve obtained using FU 30 dataset (a) PR-curve of MobileNet, (b) PR-curve of Inception, (c) PR-curve of ResNet50, (d) PR-curve of ResNet101.



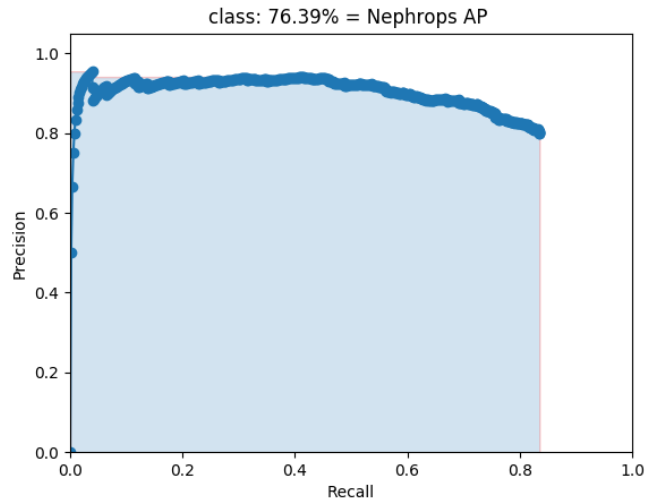
(a) MobileNet



(b) Inception

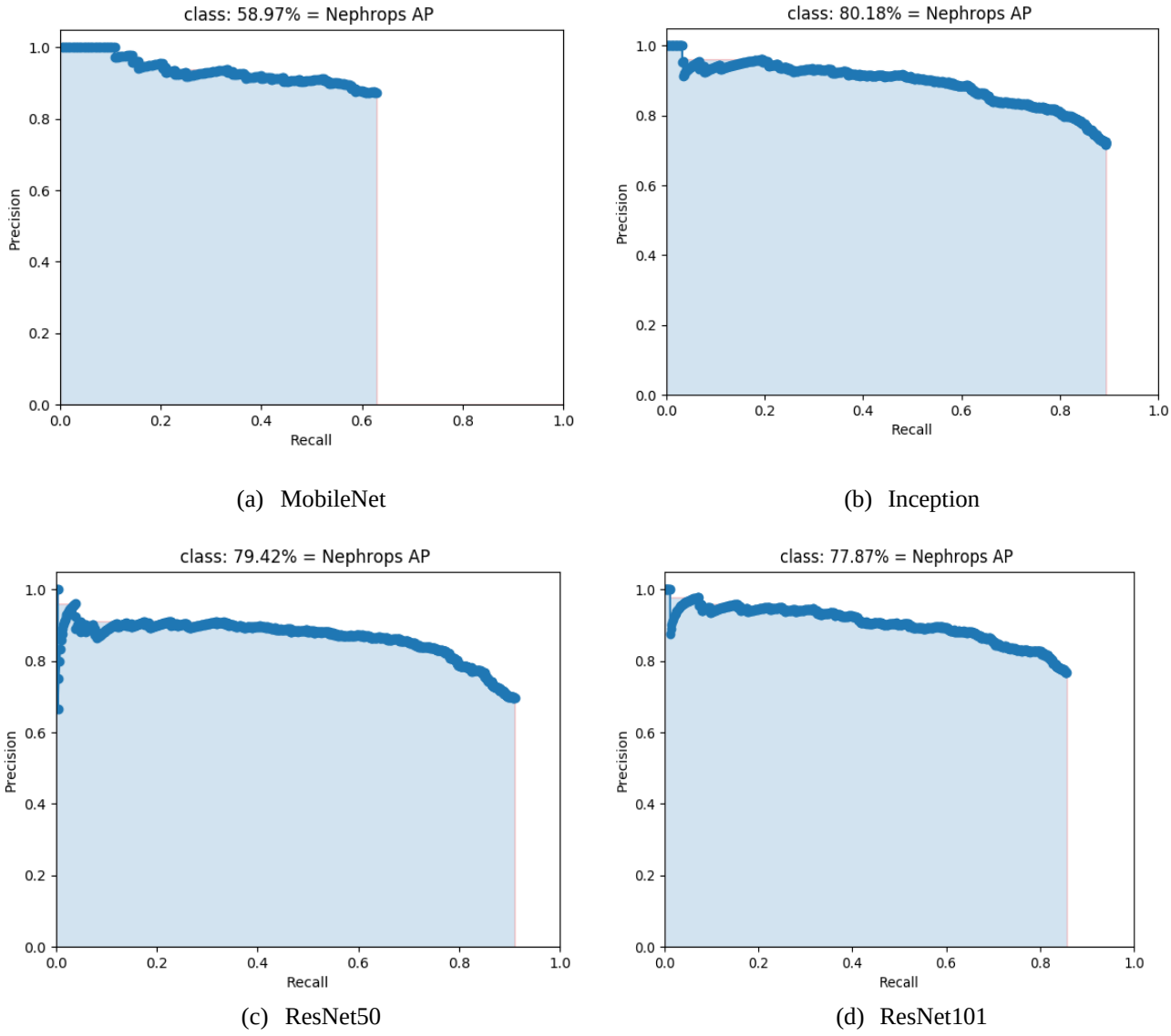


(c) ResNet50



(d) ResNet101

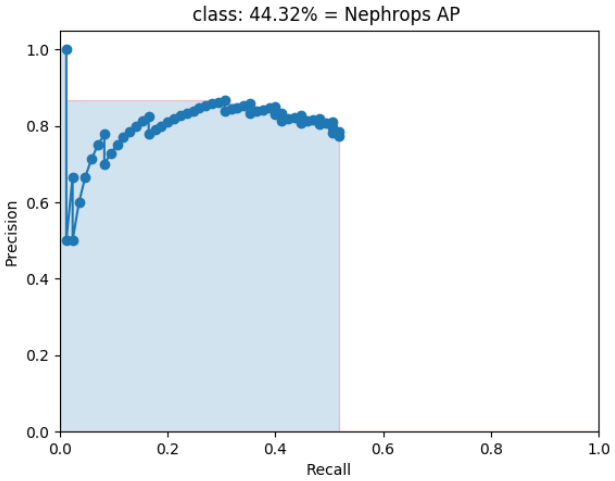
Figure 6.3: Precision-Recall curve obtained using FU 22 dataset (a) PR-curve of MobileNet, (b) PR-curve of Inception, (c) PR-curve of ResNet50, (d) PR-curve of ResNet101.



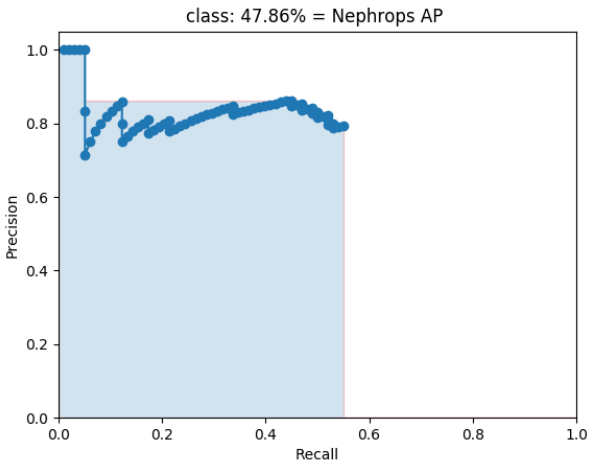
(a) MobileNet (b) Inception

(c) ResNet50 (d) ResNet101

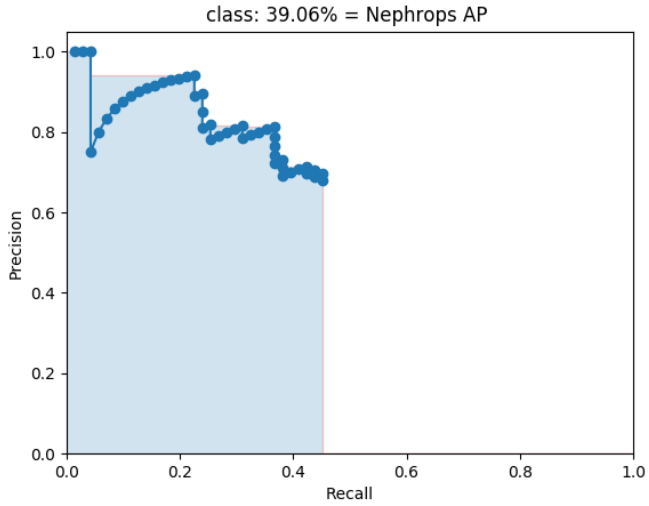
Figure 6.4: Precision-Recall curve obtained using Hybrid dataset (a) PR-curve of MobileNet, (b) PR-curve of Inception, (c) PR-curve of ResNet50, (d) PR-curve of ResNet101.



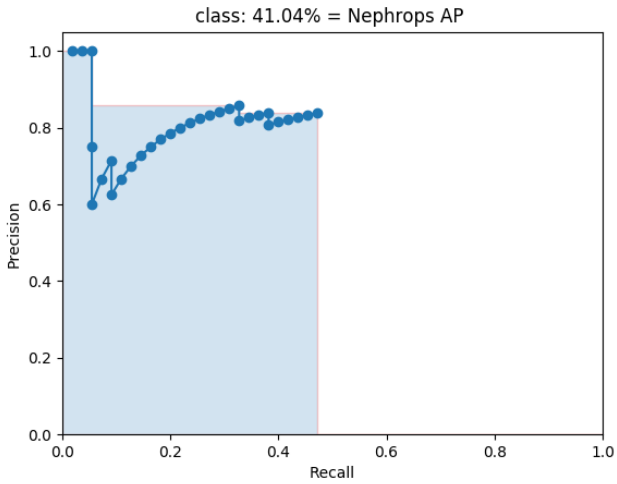
(a) MobileNet



(b) Inception

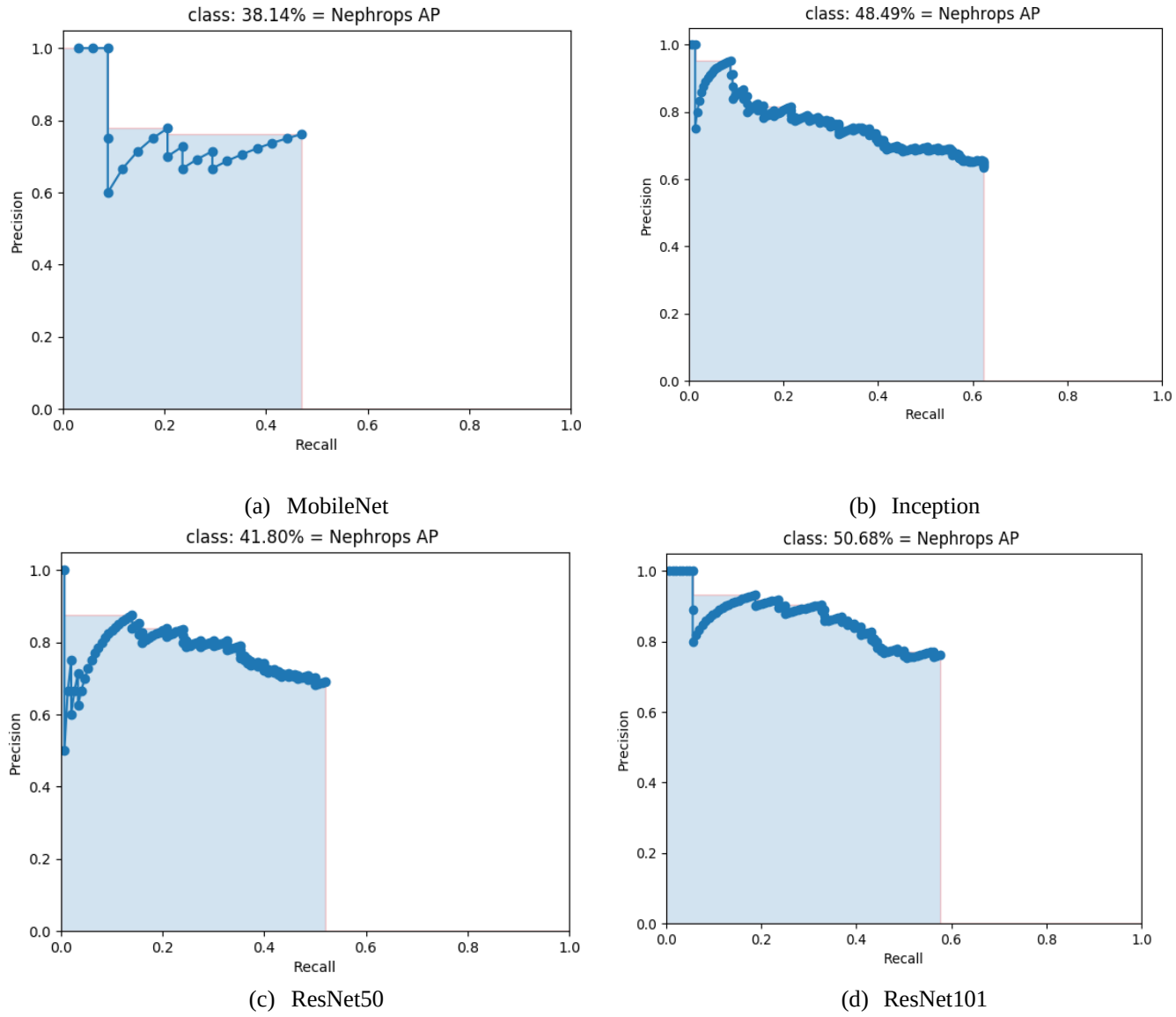


(c) ResNet50



(d) ResNet101

Figure 6.5: Precision-Recall curve obtained using FU 30 dataset [Train] and FU 22 dataset [Test] (a) PR-curve of MobileNet, (b) PR-curve of Inception, (c) PR-curve of ResNet50, (d) PR-curve of ResNet101.



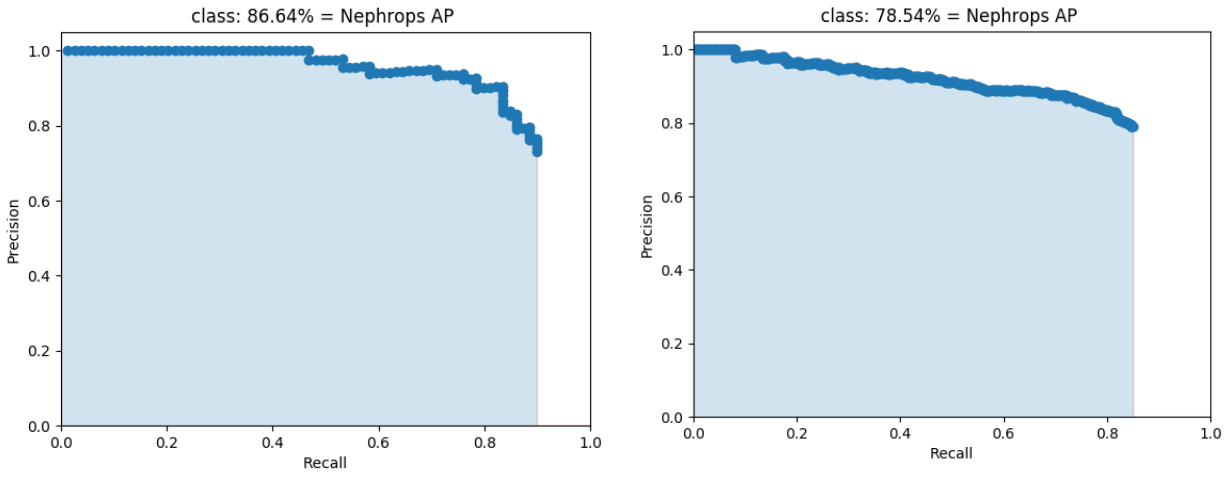
(a) MobileNet

(b) Inception

(c) ResNet50

(d) ResNet101

Figure 6.6: Precision-Recall curve obtained using FU 22 dataset [Train] and FU 30 dataset [Test] (a) PR-curve of MobileNet, (b) PR-curve of Inception, (c) PR-curve of ResNet50, (d) PR-curve of ResNet101.



FU 30 FU 22

Figure 6.7: Precision-Recall curve obtained using YOLOv3 model (a) Train and Test model using FU 30 dataset (b Train and Test model using FU 22 dataset

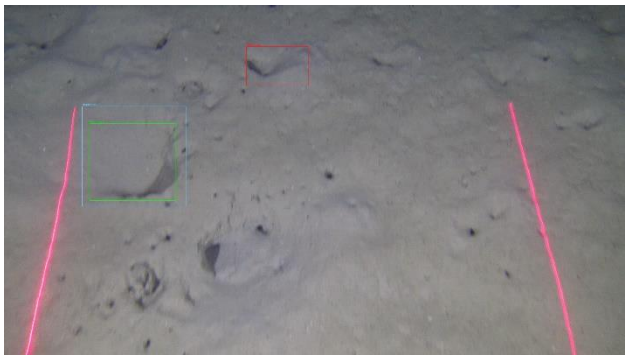
Qualitative Analysis

This section analyses the performance of different models on different datasets qualitatively. The visualization results are from the MobileNet, Inception, ResNet50, ResNet101 and YOLOv3 models, trained and tested using a different combination of the FU 30, FU 22 and hybrid datasets.

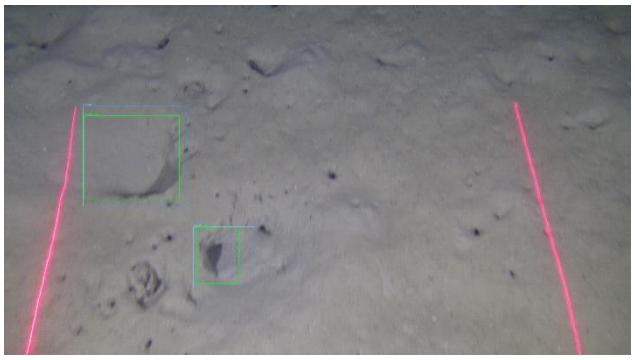
Figure. 6.8 - 6.12 shows the detections of *Nephrops* burrows using MobileNet, Inception, ResNet50, ResNet101 and YOLOv3 models with a different combination of FU 30 and FU 22 datasets. The green bounding boxes on the images shown in this section are the TP detections by the trained model. The blue bounding boxes show the correct ground annotations. The red bounding boxes are the FP detections of trained models.

Figure. 6.8 (a-e) shows the detections of all the models. These models are trained and tested using the FU 30 dataset. In this example, the MobileNet model detects one TP *Nephrops* burrows while the inception and all other models correctly detect two.

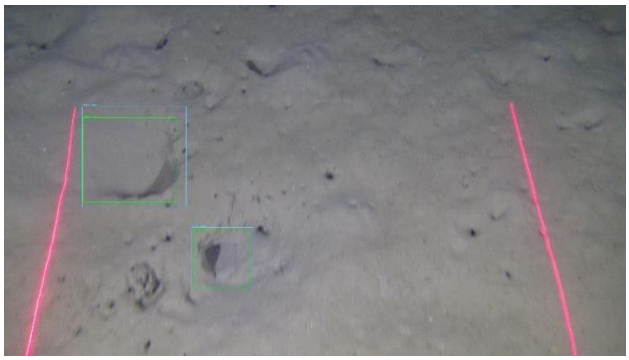
Figure. 6.9 (a-e) shows the detections on the FU 22 dataset. These models are trained and tested by the FU 22 dataset. The MobileNet missed two TP detections. The inception model also missed one detection. However, the ResNet50 and ResNet101 can detect all four TP detections. Figure 6.9(e) shows the detections from the YOLOv3 model. Figures 6.8 and 6.9 show more TP detections with the Inception, ResNet50, and ResNet101 than with the MobileNet model.



(a) MobileNet

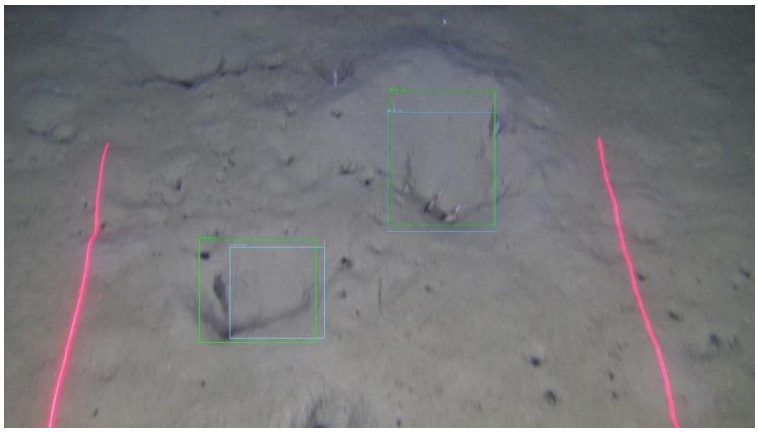


(b) Inception v2



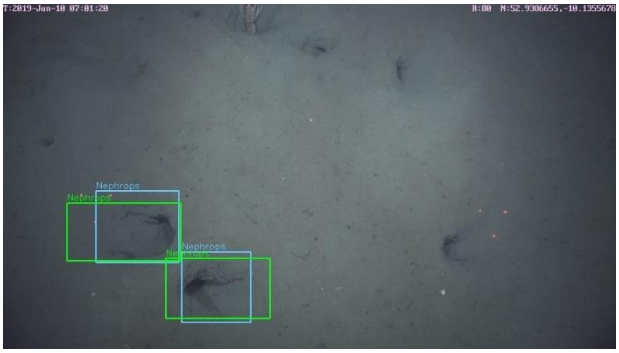
(c) ResNet50

(d) ResNet101

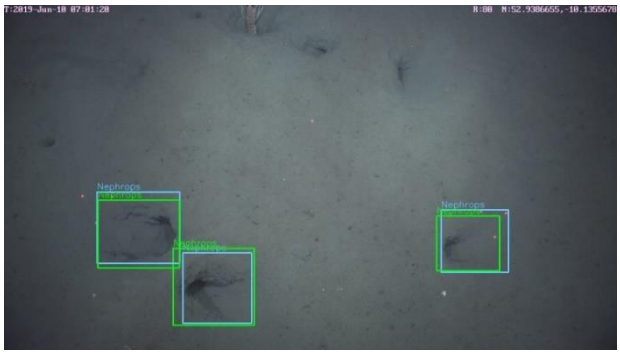


(e) YOLOv3

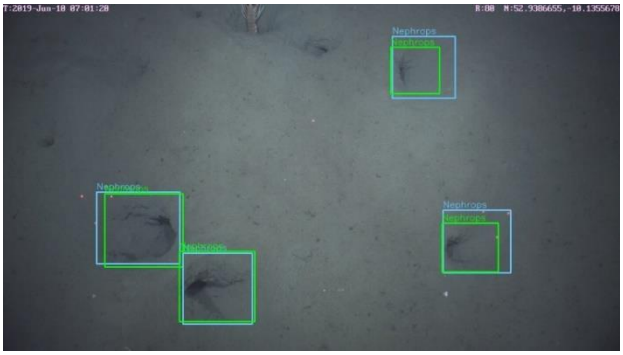
Figure 6.8: Nephrops burrows detections with FU30 dataset (a) Detections with MobileNet model, (b) Detections with Inception model, (c) Detections with ResNet50 model, (d) Detections with ResNet101 model, (e) Detections with YOLOv3 model



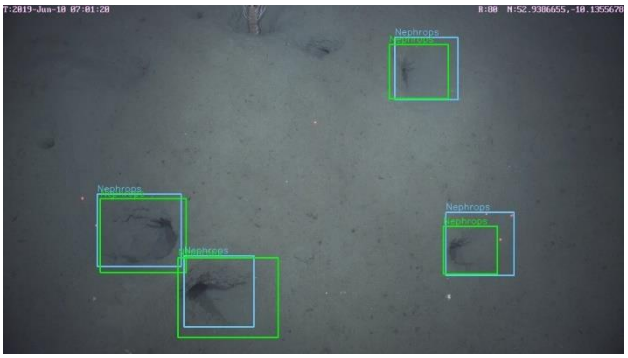
(a) MobileNet



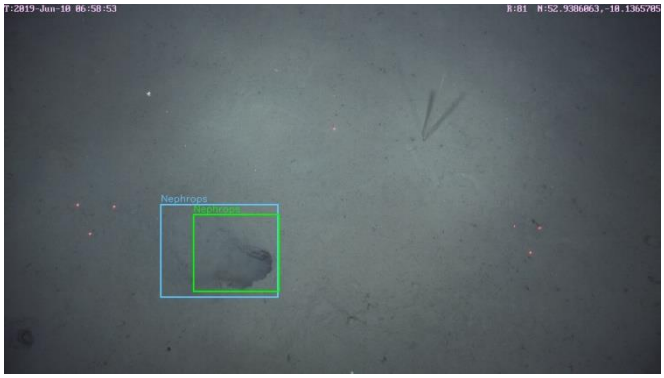
(b) Inception



(c) ResNet50



(d) ResNet101

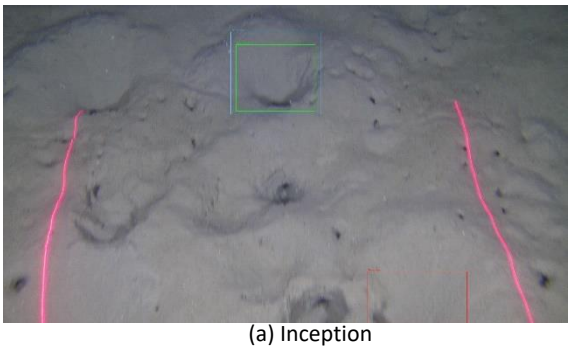


(e) YOLOv3

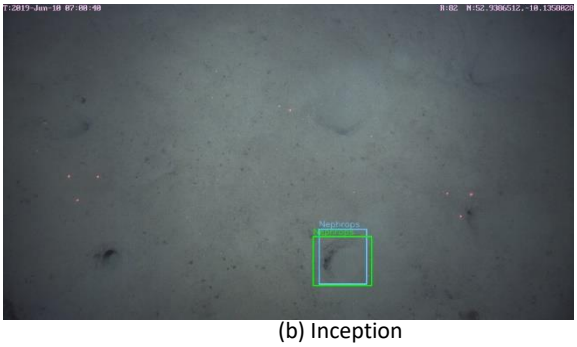
Figure 6.9: Nephrops burrows detections with FU22 dataset (a) Detections with MobileNet model, (b) Detections with Inception model, (c) Detections with ResNet50 model, (d) Detections with ResNet101 model, (e) Detections with YOLOv3 model

Figure. 6.10 shows the results obtained by MobileNet, Inception, ResNet50 and ResNet101 models when trained and tested by the hybrid dataset. Figure. 6.10 (a) and (b) shows only one TP detection of the FU 30 and FU 22 dataset with the MobileNet model. Figure 6.10 (c) and (d) show the results obtained from the Inception model: two TP and two FP detections on the FU 30 image while three TP detections of burrows on the FU 22 image. The ResNet50 results are shown in Figure 6.10 (e) and (f). The results show an improvement in FU 30 detections where there is no FP, and an extra TP is detected in FU 22 data. The ResNet101 model did not perform very well in

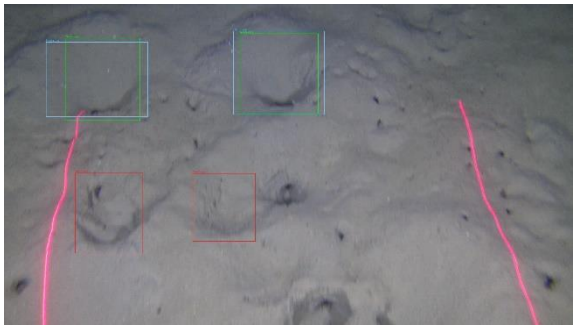
the hybrid dataset as it detected two FP and missed two TP in the FU 30 and FU 22 datasets, respectively.



(a) Inception

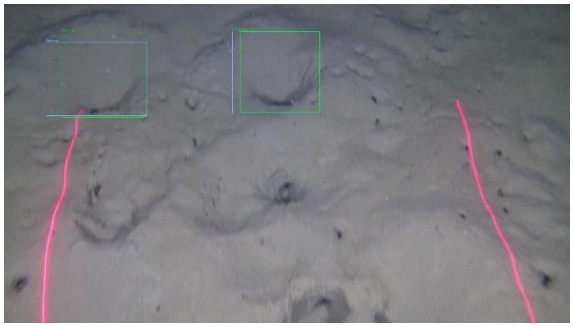


(b) Inception

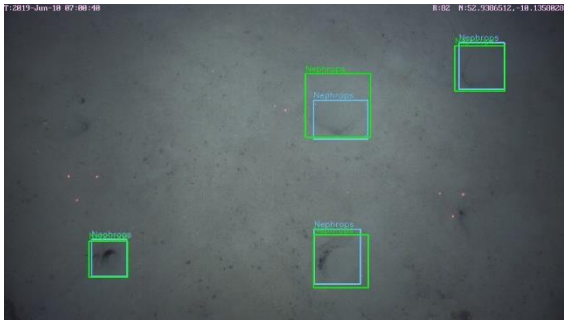


(c) Inception

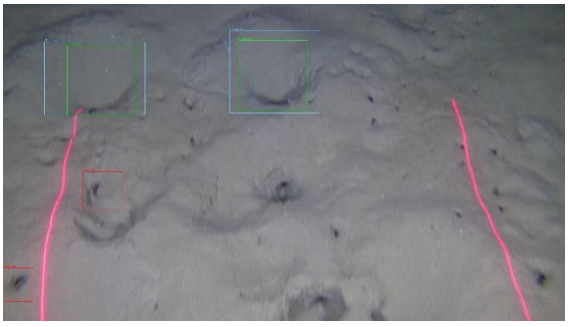
(d) Inception



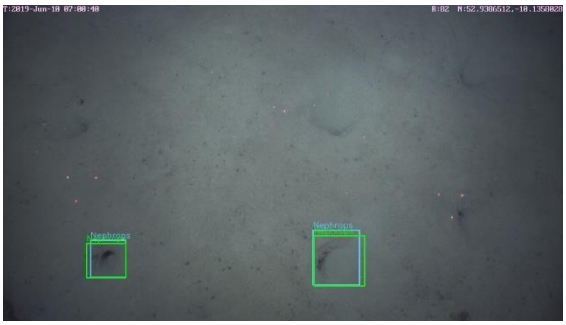
(e) ResNet50



(f) ResNet50



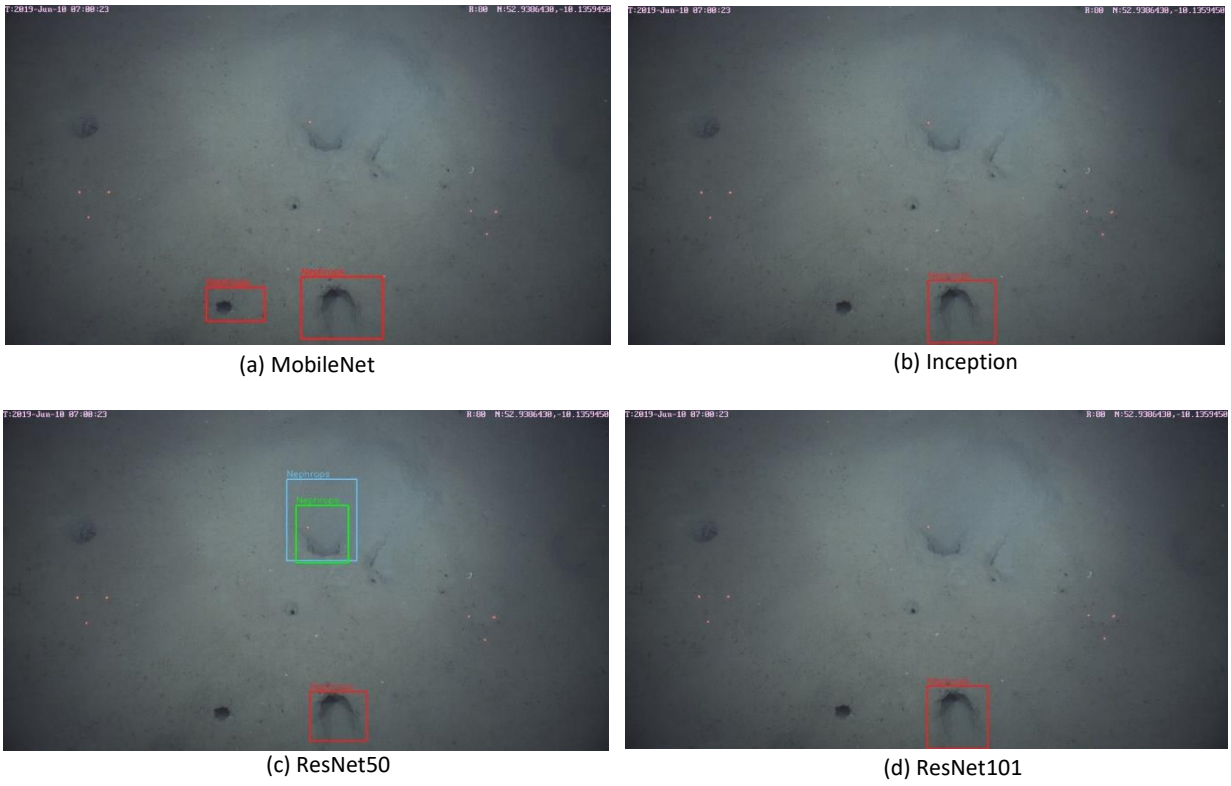
(g) ResNet101



(h) ResNet101

Figure 6.10: Nephrops burrows detections with Hybrid dataset (a) Detections of FU30 with MobileNet model, (b) Detections of FU22 with MobileNet model, (c) Detections of FU30 with Inception model, (d) Detections of FU22 with Inception model, (e) Detections of FU30 with ResNet50 model, (f) Detections of FU22 with ResNet50 model, (g) Detections of FU30 with ResNet101 model, (h) Detections of FU22 with ResNet101 model,

Figure. 6.11 shows the results obtained by MobileNet, Inception, ResNet50 and ResNet101 models when the models are trained by the FU 30 dataset and tested by the FU 22 dataset. Figure. 6.11 (a) and (b) could not detect any TP from the FU 22 dataset. However, Figure. 6.11 (c), the ResNet50 model can detect one TP with one FP. These models do not perform well as they are trained on different station datasets. This insight helps us to identify the differences between datasets and their characteristics, which usually impact the performance of the models.



(a) MobileNet (b) Inception

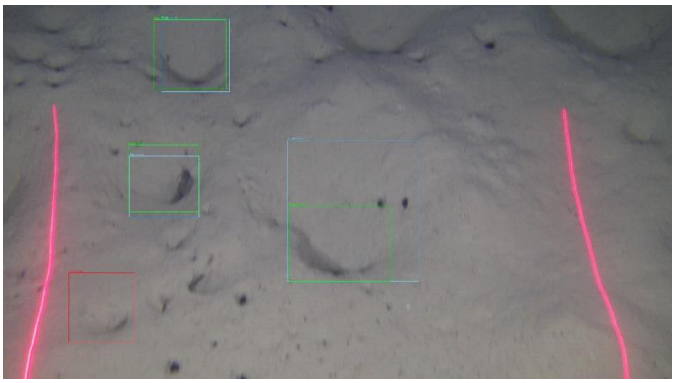
(c) ResNet50 (d) ResNet101

Figure 6.11: Nephrops burrows detections trained with FU30 dataset and Tested with FU22 (a) Detections with MobileNet model, (b) Detections with Inception model, (c) Detections with ResNet50 model, (d) Detections with ResNet101 model

Similarly, Figure. 6.12 shows the results obtained by MobileNet, Inception, ResNet50 and ResNet101 models when the models are trained by the FU 22 dataset and tested by the FU 30 dataset. These models performed well compared to the previous experiments when trained by the FU 30 dataset. The models can detect the TP in the FU 30 dataset. Figure. 6.12 (a) only detects one TP with no FP, but the Inception and ResNet50 models in Figure. 6.12 (b) and (c) can detect three TP with one FP. Figure 6.12 (d) detects only two TP with no FP.

(a) MobileNet

(b) Inception



(c) ResNet50

(d) ResNet101

Figure 6.12: Nephrops burrows detections trained with FU22 dataset and Tested with FU30 (a) Detections with MobileNet model, (b) Detections with Inception model, (c) Detections with ResNet50 model, (d) Detections with ResNet101 model

6.3. Experiments and Results of *Nephrops* Burrows Detection Refinement

This section evaluates the results of different experiments performed using the proposed detection refinement algorithm. The work applies the detection refinement on the detections from three other models, Inception, ResNet50, and ResNet101, for training with the Gulf of Cadiz dataset. Each model is trained up to 100k iterations, and a log is maintained for each 10k iteration for evaluation.

6.3.1. Quantitative Analysis

The quantitative analysis uses an annotated video with a frame rate of 25 fps to test the Inception, ResNet50, and ResNet101 models. The video is divided into five temporal segments, each of one minute. Each temporal segment has 1500 frames.

The work records several detections from each temporal segment by all three models. The detection is then processed through the proposed refinement algorithm to identify the TP, FP, and missed detections. Table 6.8 shows each model's results obtained in the 1st temporal segment and their corresponding improvement by the proposed detection refinement algorithm. The algorithm is run with W = 8, 12, and 16. In each temporal window, the algorithm is tested with $\lambda = 0.3$ and 0.4 and finds out the number of TP, FP, missed detection, and F1-score (geometric mean of precision and recall metrics) in each minute of the video. Table 6.18-6.12 shows the ground truth (GT), TP, FP, and missed (Miss) detections along with the mean values of precision, recall, and F1-score of each temporal segment. The %Before is the result obtained before applying the STF, while the %After shows the results obtained using the refinement algorithm.

Table 6.8: Detections and refinement results of 1st temporal segment

1st Temporal Segment											
	GT = 255					**Recall**		**Precision**		**F1-Score**	
	W	**l**	**TP**	**FP**	**Miss**	**%Age Before**	**%Age After**	**%Age Before**	**%Age After**	**%Age Before**	**%Age After**
Inception	8	0.3	166	9	13	65.1	70.2	94.9	95.2	77.2	80.8
	8	0.4	149	26	12	58.4	63.1	85.1	86.1	69.3	72.9
	12	0.3	165	10	15	64.7	70.6	94.3	94.7	76.7	80.9
	12	0.4	68	107	9	26.7	30.2	38.9	41.8	31.6	35.1
	16	0.3	163	12	41	63.9	80.0	93.1	94.4	75.8	86.6
	16	0.4	66	109	19	25.9	33.3	37.7	43.8	30.7	37.9
ResNet50	8	0.3	188	20	31	73.7	85.9	90.4	91.6	81.2	88.7
	8	0.4	177	31	20	69.4	77.3	85.1	86.4	76.5	81.6
	12	0.3	186	22	43	72.9	89.8	89.4	91.2	80.3	90.5
	12	0.4	110	98	19	43.1	50.6	52.9	56.8	47.5	53.5
	16	0.3	175	33	41	68.6	84.7	84.1	86.7	75.6	85.7
	16	0.4	93	115	12	36.5	41.2	44.7	47.7	40.2	44.2
ResNet101	8	0.3	217	26	24	85.1	94.5	89.3	90.3	87.1	92.3
	8	0.4	164	79	20	64.3	72.2	67.5	70.0	65.9	71.0
	12	0.3	188	55	28	73.7	84.7	77.4	79.7	75.5	82.1
	12	0.4	100	143	18	39.2	46.3	41.2	45.2	40.2	45.7
	16	0.3	181	62	21	71.0	79.2	74.5	76.5	72.7	77.8
	16	0.4	96	147	13	37.6	42.7	39.5	42.6	38.6	42.7

Table 6.9 shows each model's results obtained in the 2nd temporal segment and their corresponding improvement by the proposed detection refinement algorithm. The algorithm is run with W = 8, 12, and 16.

Table 6.9: Detections and refinement results of 2nd temporal segment

2nd Temporal Segment											
	GT = 585					**Recall**		**Precision**		**F1-Score**	
	W	**l**	**TP**	**FP**	**Miss**	**%Age Before**	**%Age After**	**%Age Before**	**%Age After**	**%Age Before**	**%Age After**
Inception	8	0.3	398	33	61	68.0	78.5	92.3	93.3	78.3	85.2
	8	0.4	324	107	46	55.4	63.2	75.2	77.6	63.8	69.7
	12	0.3	393	38	73	67.2	79.7	91.2	92.5	77.4	85.6
	12	0.4	271	160	41	46.3	53.3	62.9	66.1	53.3	59.0
	16	0.3	393	38	115	67.2	86.8	91.2	93.0	77.4	89.8
	16	0.4	269	162	68	46.0	57.6	62.4	67.5	53.0	62.2
ResNet50	8	0.3	420	45	105	71.8	89.7	90.3	92.1	80.0	90.9
	8	0.4	306	159	85	52.3	66.8	65.8	71.1	58.3	68.9
	12	0.3	404	61	114	69.1	88.5	86.9	89.5	77.0	89.0
	12	0.4	241	224	78	41.2	54.5	51.8	58.7	45.9	56.6
	16	0.3	363	102	168	62.1	90.8	78.1	83.9	69.1	87.2
	16	0.4	232	233	104	39.7	57.4	49.9	59.1	44.2	58.2
ResNet101	8	0.3	441	31	103	75.4	93.0	93.4	94.6	83.4	93.8
	8	0.4	433	139	89	74.0	89.2	75.7	79.0	74.8	83.8
	12	0.3	468	49	103	80.0	97.6	90.5	92.1	84.9	94.8
	12	0.4	309	263	68	52.8	64.4	54.0	58.9	53.4	61.6
	16	0.3	415	57	145	70.9	95.7	87.9	90.8	78.5	93.2
	16	0.4	300	272	89	51.3	66.5	52.4	58.9	51.9	62.4

Table. 6.10 shows each model's results obtained by the 3rd temporal segment and their corresponding improvement by the proposed detection refinement algorithm. The algorithm is run with W = 8, 12, and 16.

Table 6.10: Detections and refinement results of 3rd temporal segment

3rd Temporal Segment											
	GT = 480					**Recall**		**Precision**		**F1-Score**	
	W	**l**	**TP**	**FP**	**Miss**	**%Age Before**	**%Age After**	**%Age Before**	**%Age After**	**%Age Before**	**%Age After**
Inception	8	0.3	163	23	45	34.0	43.3	87.6	90.0	48.9	58.5
	8	0.4	132	54	37	27.5	35.2	71.0	75.8	39.6	48.1
	12	0.3	160	26	47	33.3	43.1	86.0	88.8	48.0	58.1
	12	0.4	106	80	30	22.1	28.3	57.0	63.0	31.8	39.1
	16	0.3	159	27	46	33.1	42.7	85.5	88.4	47.7	57.6
	16	0.4	64	122	28	13.3	19.2	34.4	43.0	19.2	26.5
ResNet50	8	0.3	291	43	87	60.6	78.8	87.1	89.8	71.5	83.9
	8	0.4	269	65	69	56.0	70.4	80.5	83.9	66.1	76.6
	12	0.3	280	54	106	58.3	80.4	83.8	87.7	68.8	83.9
	12	0.4	203	131	59	42.3	54.6	60.8	66.7	49.9	60.0
	16	0.3	274	60	114	57.1	80.8	82.0	86.6	67.3	83.6
	16	0.4	181	153	55	37.7	49.2	54.2	60.7	44.5	54.3
ResNet101	8	0.3	354	40	105	73.8	95.6	89.8	92.0	81.0	93.8
	8	0.4	335	59	88	69.8	88.1	85.0	87.8	76.7	87.9
	12	0.3	368	46	111	76.7	99.8	88.9	91.2	82.3	95.3
	12	0.4	302	92	64	62.9	76.3	76.6	79.9	69.1	78.0
	16	0.3	325	45	136	67.7	96.0	87.8	91.1	76.5	93.5
	16	0.4	268	126	79	55.8	72.3	68.0	73.4	61.3	72.8

Table. 6.11 shows each model's results obtained in the 4th temporal segment and their corresponding improvement by the proposed detection refinement algorithm. The algorithm is run with W = 8, 12, and 16.

Table 6.11: Detections and refinement results of 4th temporal segment

4th Temporal Segment											
	GT = 468					**Recall**		**Precision**		**F1-Score**	
	W	**l**	**TP**	**FP**	**Miss**	**%Age Before**	**%Age After**	**%Age Before**	**%Age After**	**%Age Before**	**%Age After**
Inception	8	0.3	304	24	64	65.0	78.6	92.7	93.9	76.4	85.6
	8	0.4	280	48	51	59.8	70.7	85.4	87.3	70.4	78.2
	12	0.3	296	32	67	63.2	77.6	90.2	91.9	74.4	84.1
	12	0.4	235	93	48	50.2	60.5	71.6	75.3	59.0	67.1
	16	0.3	293	35	72	62.6	78.0	89.3	91.3	73.6	84.1
	16	0.4	206	122	43	44.0	53.2	62.8	67.1	51.8	59.4
ResNet50	8	0.3	330	28	66	70.5	84.6	92.2	93.4	79.9	88.8
	8	0.4	284	74	50	60.7	71.4	79.3	81.9	68.8	76.3
	12	0.3	327	31	81	69.9	87.2	91.3	92.9	79.2	90.0
	12	0.4	247	111	50	52.8	63.5	69.0	72.8	59.8	67.8
	16	0.3	325	33	98	69.4	90.4	90.8	92.8	78.7	91.6
	16	0.4	232	126	49	49.6	60.0	64.8	69.0	56.2	64.2
ResNet101	8	0.3	388	42	50	82.9	93.6	90.2	91.3	86.4	92.4
	8	0.4	352	78	37	75.2	83.1	81.9	83.3	78.4	83.2
	12	0.3	387	43	57	82.7	94.9	90.0	91.2	86.2	93.0
	12	0.4	247	183	38	52.8	60.9	57.4	60.9	55.0	60.9
	16	0.3	380	50	61	81.2	94.2	88.4	89.8	84.6	92.0
	16	0.4	232	198	31	49.6	56.2	54.0	57.0	51.7	56.6

Table. 6.12 shows each model's results obtained in the 5th temporal segment and their corresponding improvement by the proposed detection refinement algorithm. The algorithm is run with W = 8, 12, and 16.

Table 6.12: Detections and refinement results of 5th temporal segment

5th Temporal Segment											
	GT = 571					**Recall**		**Precision**		**F1-Score**	
	W	**l**	**TP**	**FP**	**Miss**	**%Age Before**	**%Age After**	**%Age Before**	**%Age After**	**%Age Before**	**%Age After**
Inception	8	0.3	349	26	73	61.1	73.9	93.1	94.2	73.8	82.8
	8	0.4	265	110	58	46.4	56.6	70.7	74.6	56.0	64.3
	12	0.3	302	73	75	52.9	66.0	80.5	83.8	63.8	73.8
	12	0.4	219	156	42	38.4	45.7	58.4	62.6	46.3	52.8
	16	0.3	300	75	100	52.5	70.1	80.0	84.2	63.4	76.5
	16	0.4	199	176	51	34.9	43.8	53.1	58.7	42.1	50.2
ResNet50	8	0.3	390	27	67	68.3	80.0	93.5	94.4	78.9	86.6
	8	0.4	353	64	50	61.8	70.6	84.7	86.3	71.5	77.6
	12	0.3	360	57	56	63.0	72.9	86.3	87.9	72.9	79.7
	12	0.4	268	149	33	46.9	52.7	64.3	66.9	54.3	59.0
	16	0.3	358	59	85	62.7	77.6	85.9	88.2	72.5	82.6
	16	0.4	224	193	40	39.2	46.2	53.7	57.8	45.3	51.4
ResNet101	8	0.3	494	41	54	86.5	96.0	92.3	93.0	89.3	94.5
	8	0.4	436	99	28	76.4	81.3	81.5	82.4	78.8	81.8
	12	0.3	463	72	41	81.1	88.3	86.5	87.5	83.7	87.9
	12	0.4	309	226	21	54.1	57.8	57.8	59.4	55.9	58.6
	16	0.3	453	82	58	79.3	89.5	84.7	86.2	81.9	87.8
	16	0.4	258	277	16	45.2	48.0	48.2	49.7	46.7	48.8

Table 6.13 shows the accumulative ground truth (GT), TP, FP, and missed (Miss) detections along with the mean values of precision, recall, and F1-score of each temporal segment. The %Before is the result obtained before applying the STF, while the %After shows the results obtained using the refinement algorithm. Table 6.13 shows that ResNet101 gives the best F1-score in the five temporal segments, followed by ResNet50 and Inception. A small IoU value of 0.3 was better than 0.4 in terms of precision, recall, and F1-score values because the area surrounding burrows is sometimes not well defined for all three models. The effect of window size W shows a trend of better results for smaller values (mostly, W = 8 is better than W = 12 and W = 16).

Table 6.13: Detections of all temporal segments with refinements. Detections are refined using W = 8, 12, and 16 with λ = 0.3 and 0.4. The refined detection shows total number of TP, FP, and missed detections and F1-score.

	GT = 2359					**Recall**		**Precision**		**F1-Score**	
	W	**l**	**TP**	**FP**	**Miss**	**%Age Before**	**%Age After**	**%Age Before**	**%Age After**	**%Age Before**	**%Age After**
Inception	8	0.3	1380	115	256	58.5	69.4	92.3	93.4	71.6	79.6
	8	0.4	1150	345	204	48.7	57.4	76.9	79.7	59.7	66.7
	12	0.3	1316	179	277	55.8	67.5	88.0	89.9	68.3	77.1
	12	0.4	899	596	170	38.1	45.3	60.1	64.2	46.7	53.1
	16	0.3	1308	187	374	55.4	71.3	87.5	90.0	67.9	79.6
	16	0.4	804	691	209	34.1	42.9	53.8	59.4	41.7	49.9
ResNet50	8	0.3	1619	163	356	68.6	90.6	90.9	92.9	78.2	91.8
	8	0.4	1389	393	274	58.9	87.2	77.9	84.0	67.1	85.5
	12	0.3	1557	225	400	66.0	92.5	87.4	90.7	75.2	91.6
	12	0.4	1069	713	239	45.3	85.7	60.0	73.9	51.6	79.4
	16	0.3	1495	287	506	63.4	97.0	83.9	88.9	72.2	92.7
	16	0.4	962	820	260	40.8	86.6	54.0	71.3	46.5	78.2
ResNet101	8	0.3	1894	180	336	80.3	94.5	91.3	92.5	85.5	93.5
	8	0.4	1720	454	262	72.9	84.0	79.1	81.4	75.9	82.7
	12	0.3	1874	265	340	79.4	93.9	87.6	89.3	83.3	91.5
	12	0.4	1267	907	209	53.7	62.6	58.3	61.9	55.9	62.3
	16	0.3	1754	296	421	74.4	92.2	85.6	88.0	79.6	90.1
	16	0.4	1154	1020	228	48.9	58.6	53.1	57.5	50.9	58.1

After applying the detection refinement algorithm, the work performed experiments to determine the accuracy using mean average precision (mAP). Two different image sets are selected from the third (image set 1) and fifth (image set 2) temporal segments. Each set consists of almost 200 images. Table 6.14 shows the definition of experiments performed.

Table 6.14: Experiments definition for detection refinement.

Experiment	Model	Testing Set
Experiment 1	Inception	Image set 1
Experiment 2	ResNet50	Image set 1
Experiment 3	ResNet101	Image set 1
Experiment 4	Inception	Image set 2
Experiment 5	ResNet50	Image set 2
Experiment 6	ResNet101	Image set 2

Figures 6.13 – 6.18. shows the results of the first three experiments performed on image set 1. The graphs show the results of detections with and without applying the detection refinement algorithm. Results clearly show that the mAP increases after using the refinement algorithm. Figure 6.13 shows the detections from the Inception model and their refinement. Figure. 6.14. shows the detections from the ResNet50 model and their refinement, while Figure. 6.15 shows the detections from the ResNet101 model and their refinement.

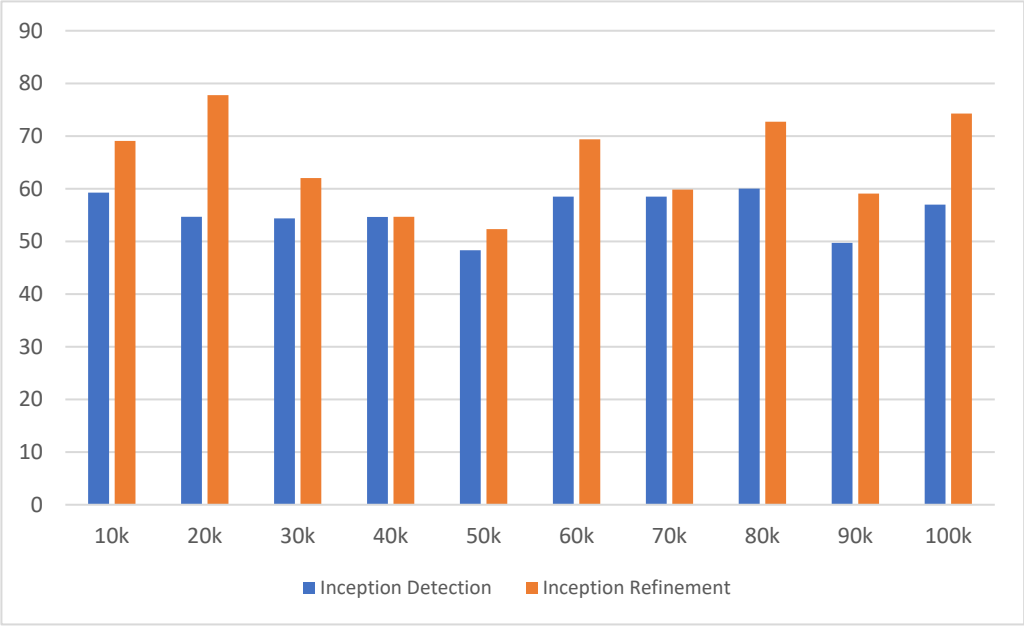


Figure 6.13: Experiment performed with image set 1 show mean average precision (mAP) of detection refinement with Inception model.

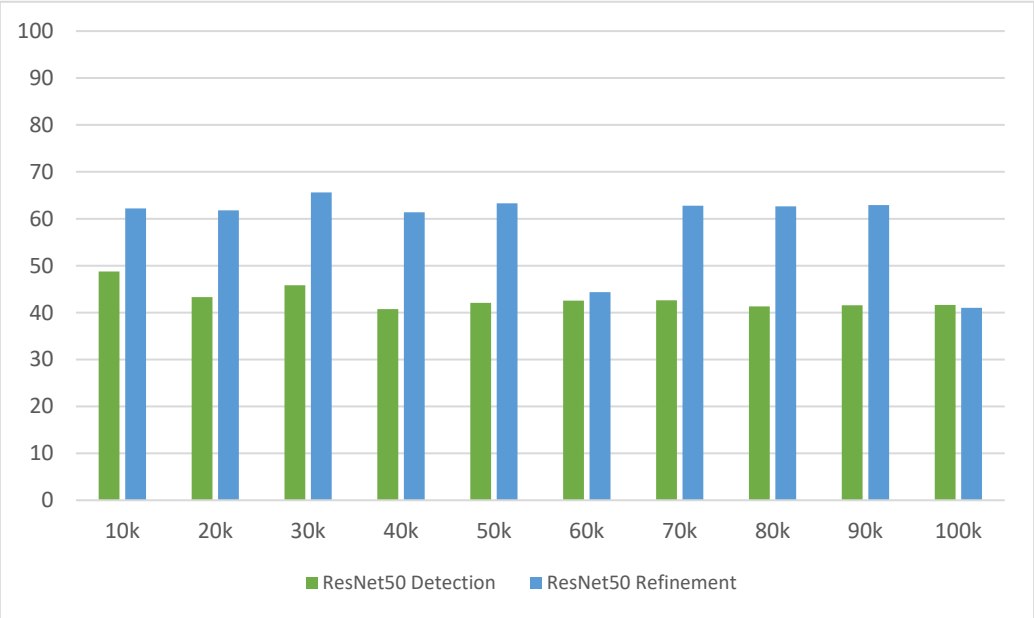


Figure 6.14: Experiment performed with image set 1 show mean average precision (mAP) of detection refinement with ResNet50 model.

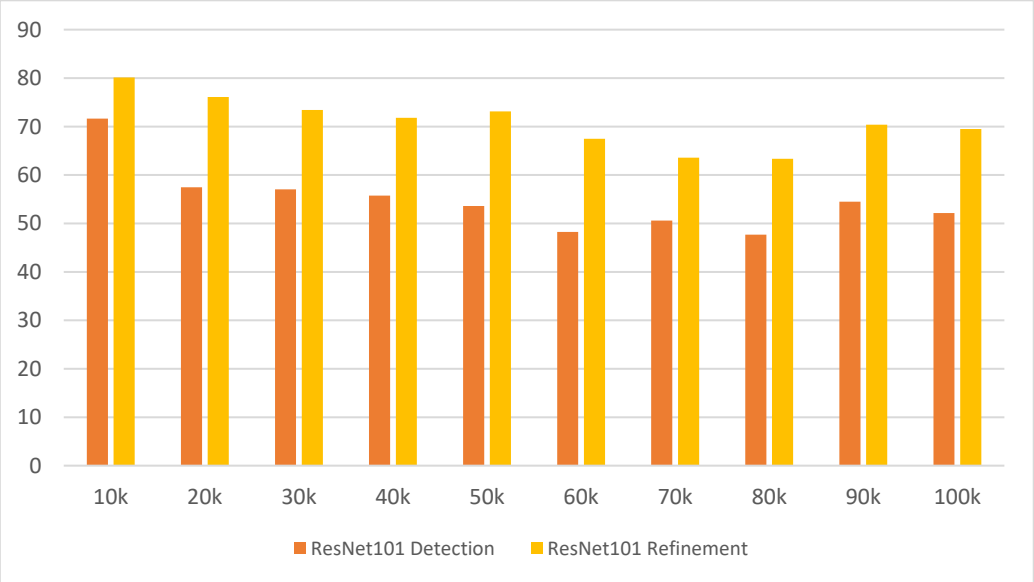


Figure 6.15: Experiment performed with image set 1 show mean average precision (mAP) of detection refinement with ResNet101 model.

Figures 6.16-6.18. are the results of experiments 4, 5 and 6 performed on image set 2. The graphs show the results of detections with and without applying the detection refinement algorithm. Results clearly show that the mAP increases after using the refinement algorithm. Figure 6.16 shows the detections from the Inception model and their refinement. Figure 6.17 shows the detections from the ResNet50 model and their refinement. In contrast, Figure 6.18 shows the detections from the ResNet101 model and their refinement.

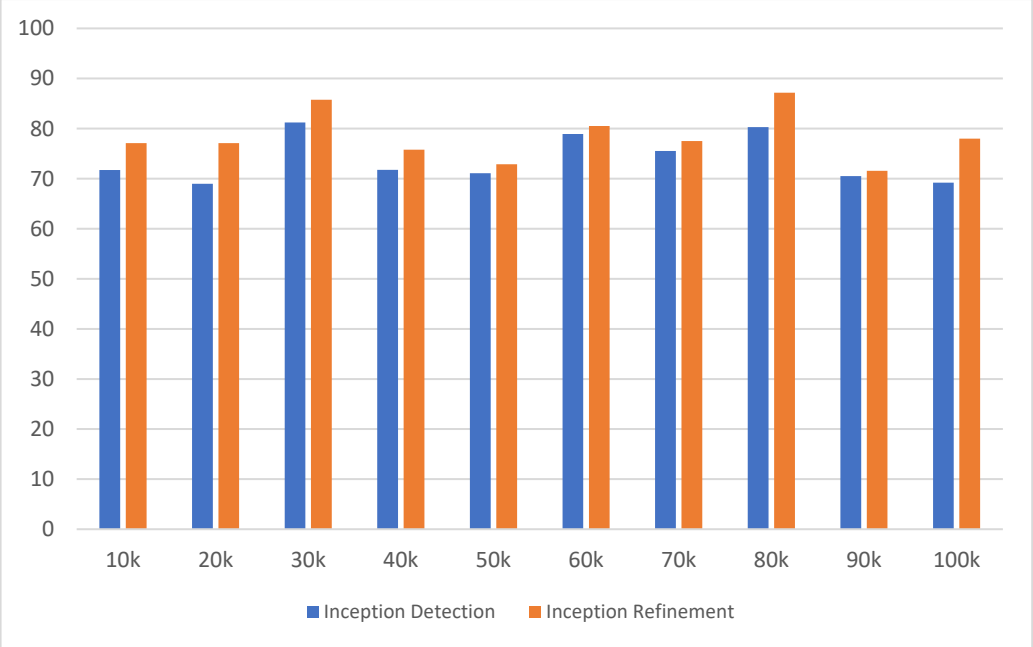


Figure 6.16: Experiment performed with image set 2 shows mean average precision (mAP) of detection refinement with Inception model.

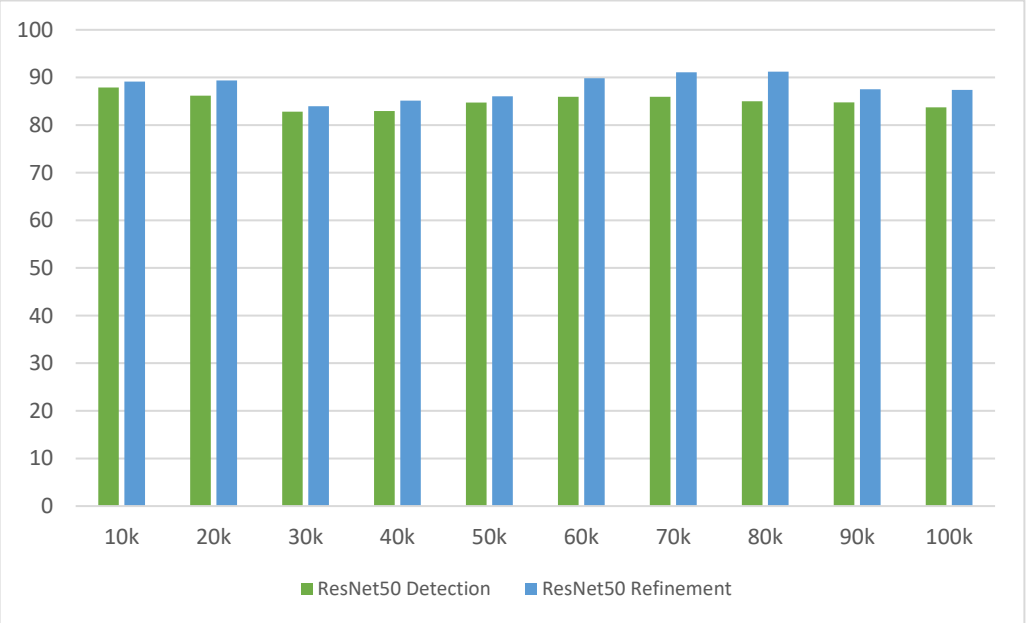


Figure 6.17: Experiment performed with image set 2 shows mean average precision (mAP) of detection refinement with ResNet50 model.

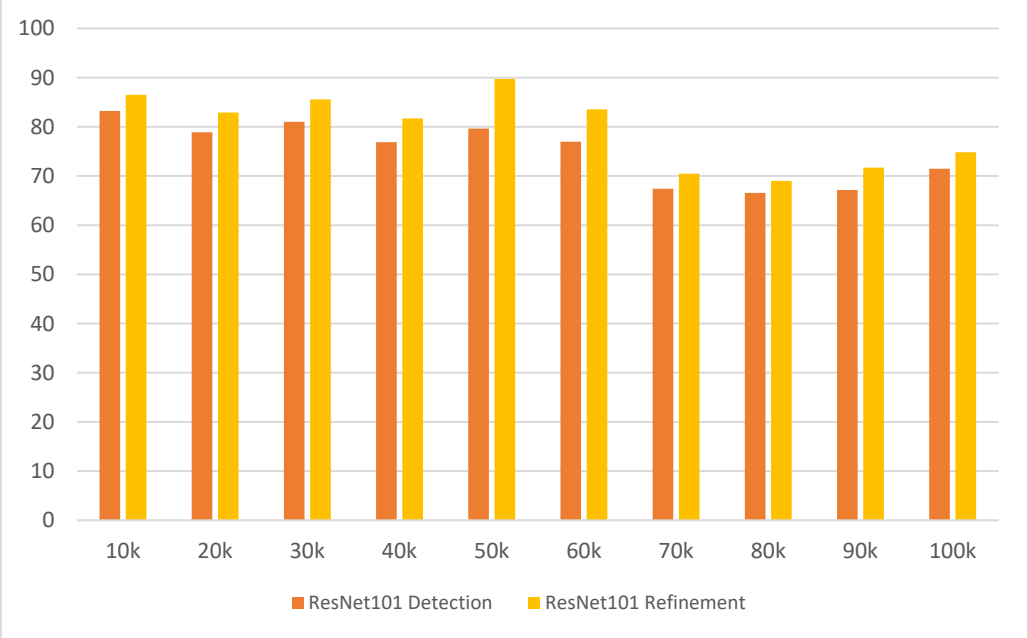


Figure 6.18: Experiment performed with image set 2 show mean average precision (mAP) of detection refinement with ResNet101 model.

Image set 1 result shows a higher improvement in mAP after applying the proposed refinement algorithm compared to Image set 2 results, where some improvement is also achieved, partly because Image set 1 obtained a lower mAP before the refinement. Image set 2 is better quality than the images in Image set 1 because of the better appearance of burrows and fewer camera movement artefacts. This suggests that mAP is quite sensitive to video quality and that the proposed refinement algorithm somewhat compensates for this.

6.3.2. Qualitative Analysis

In this section, the performance of the proposed detection refinement algorithm is analyzed qualitatively by applying it to the results obtained from the Inception, ResNet50, and ResNet101 models. The red bounding boxes on the images shown in this section are the original detections obtained from the models; green bounding boxes are the recovered missed detections after applying the refinement algorithm, and ground truth data are marked with blue bounding boxes.

Figure. 6.19 shows a typical example of suppression of FP from the detections obtained from the Inception model. Figure 6.19 (a–c) shows three frames where all burrows' entrances are detected correctly. However, some FP detections are also obtained yet are suppressed by our proposed algorithm, resulting in a correct detection, shown in Figure 6.19 (d–f).

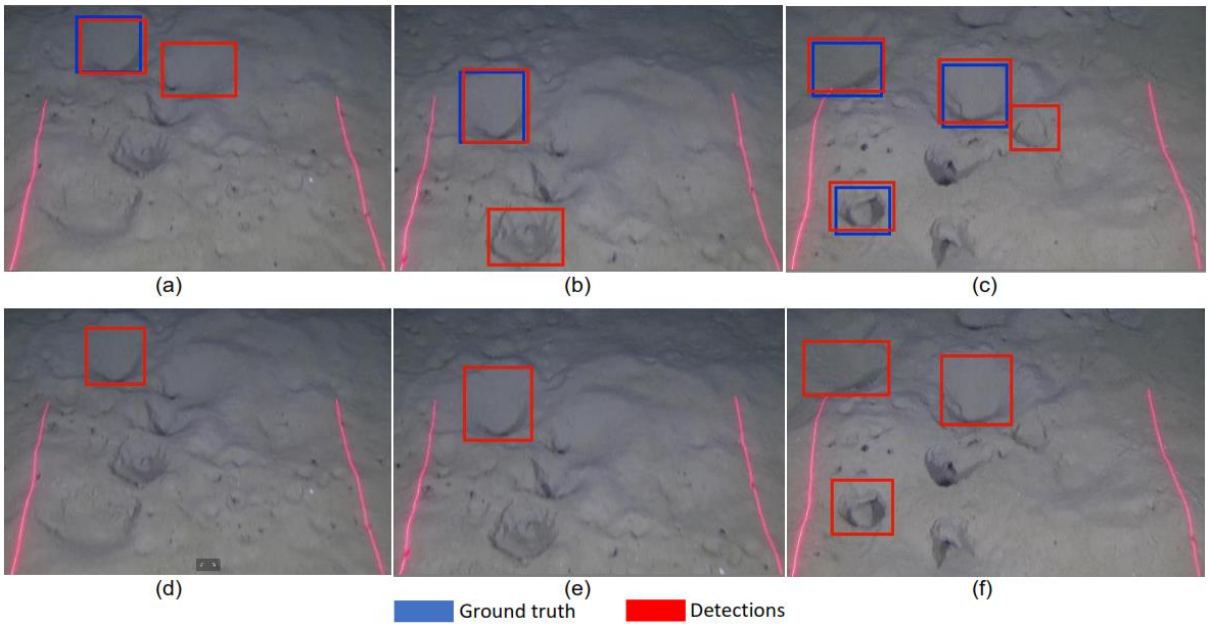


Figure 6.19: False positive suppression using detection refinement algorithm (a–c) are the ground truth (blue color bounding boxes), and original detections from the Inception model (red color bounding boxes) (d–f) are the refined detections.

A second rectification performed by the proposed detection refinement algorithm is the identification of missed detections. Figure 6.20 shows an example of six consecutive frames before (a–f) and after (g–l) the application of this algorithm. Figure 6.20 (a) shows two *Nephrops* burrows detections but missed one detection in (b), (c), (d), and (e), which is correctly rectified by the algorithm, as it is shown in the corresponding images (h), (i), (j), and (k). It can also be shown that ground truth annotations contain a third object in Figure 6.20 (d, f), which is correctly detected by the models but is not shown in Figure 6.20 (a–c, e), possibly due to the viewing angle of some frames. However, the identification of missed detections has a good impact on improving the accuracy and precision of the results. A similar approach is followed to rectify the detections from the ResNet50 and ResNet101 models.

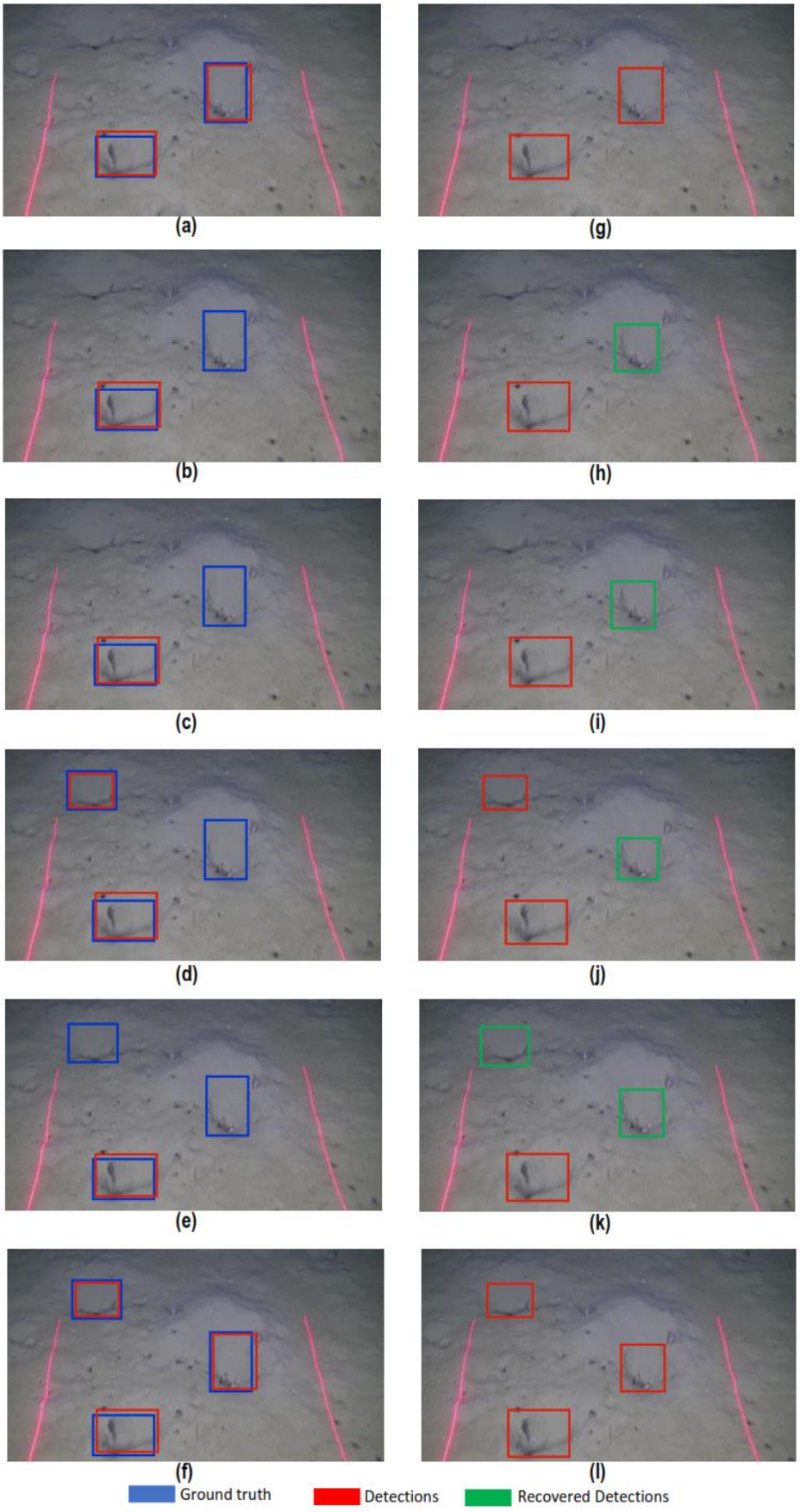


Figure 6.20: Identification of true positive missed detections. Panels (a–f) are the original detections from the Inception model, and (g–l) are the identification of missed detections in the consecutive frames.

6.4.Experiments and Results of *Nephrops* Burrows Tracking and Counting

This section evaluated the results of different experiments performed with tracking and counting *Nephrops* burrows. The burrows tracking and counting algorithm is applied to the FU30 dataset. The YOLOv3 model is used and trained with the FU 30 dataset for object detection. The model is trained up to 100k iterations, and a log is maintained for each 10k iteration for evaluation.

Before applying the proposed tracking algorithm, the work applies multiple OpenCV tracking algorithms to track and count the burrows. The OpenCV tracking algorithms are Boostig tracker, MIL, KCF, TLD, MedianFLow, MOSSE and CSRT. For experimentation. a nine-minute video from the FU 30 station is used. The video is divided into nine temporal segments of one minute each. Each minute is evaluated separately and experimented with the tracker mentioned above and later with the proposed tracking and counting algorithm. The results are presented quantitatively and qualitatively.

6.4.1. Quantitative Analysis

The quantitative analysis divides an annotated video with a frame rate of 25 fps into nine temporal segments. Each segment is used for the detection and counting of burrows. The detection and tracking algorithms run together to find the unique number of burrows in each temporal segment separately. The work recorded several detections from each temporal segment by YOLOv3. The detection is then processed through the proposed tracking and counting algorithm to identify unique burrows. The algorithm will count the number of burrows after each temporal segment. Table 6.15 shows the complete results of each temporal segment. The results show the tracking and counting with the frame number of each burrow detected. Here, the work does not consider FP detections. The aim is to test the proposed algorithm on the TP tracking and counting of *Nephrops* burrows.

Table 6.15: Details of each burrow count framewise distribution in the proposed temporal segments.

Temporal Segments	Burrow Count (Ground truth)	Proposed Tracking and Counting											Total Count
RF09_Min1	10	Frame No	161	362	464	624	676	1040	1050	1440	-	-	-	**10**
		Burrow Count	1	2	2	1	1	1	1	1	-	-	-	
RF09_Min2	16	Frame No	92	114	210	230	290	309	360	393	502	1020	1070	**16**
		Burrow Count	1	1	1	3	2	2	2	1	1	1	1	
RF09_Min3	11	Frame No	1	180	350	421	576	675	1154	1450	-	-	-	**11**
		Burrow Count	1	1	1	1	1	2	3	1	-	-		
RF09_Min4	9	Frame No	160	410	810	996	1102	1165	1349	1390	-	-	-	**9**
		Burrow Count	1	1	1	2	1	1	1	1	-	-	-	
RF09_Min5	14	Frame No	224	290	776	870	940	1130	1230	1250	1270	1315	-	**14**
		Burrow Count	1	2	2	1	1	1	2	1	1	2	-	
RF09_Min6	10	Frame No	320	510	630	665	718	730	1097	1111	1460	-	-	**10**
		Burrow Count	1	1	1	2	1	1	1	1	1	-	-	
RF09_Min7	13	Frame No	460	539	775	805	825	856	900	920	1035	1225	1400	**13**
		Burrow Count	1	1	2	1	1	1	1	2	1	1	1	
RF09_Min8	6	Frame No	76	274	320	657	885	1360	-	-	-	-	-	**6**
		Burrow Count	1	1	1	1	1	1	-	-	-	-	-	
RF09_Min9	18	Frame No	49	66	454	466	484	516	760	780	793	830	1335	**18**
		Burrow Count	1	2	1	1	3	2	1	2	3	1	1	

6.4.2. Qualitative Analysis

This section measures the performance of the proposed tracking and counting algorithm qualitatively. The blue bounding boxes on the images in this section are the original detections obtained from the models with the tracking in consecutive frames.

Figure 6.21 shows the identification, tracking and counting of a *Nephrops* burrow in some consecutive frames. These frames are extracted from the 'RF09_Min1' temporal segment. The figure shows the detection in frame one, and the proposed tracking algorithm also starts tracking the burrow. The burrow is detected in subsequent consecutive frames, and the tracking module tracks the burrow based on spatial and temporal analysis. The figure shows eight different frames from the first fifty frames to show how the tracking algorithm works and counting the burrow. The burrow counts are not increasing as the same burrow is identified and tracked in each frame. Hence, the proposed work can count the number of unique *Nephrops* burrows in a video frame.

.

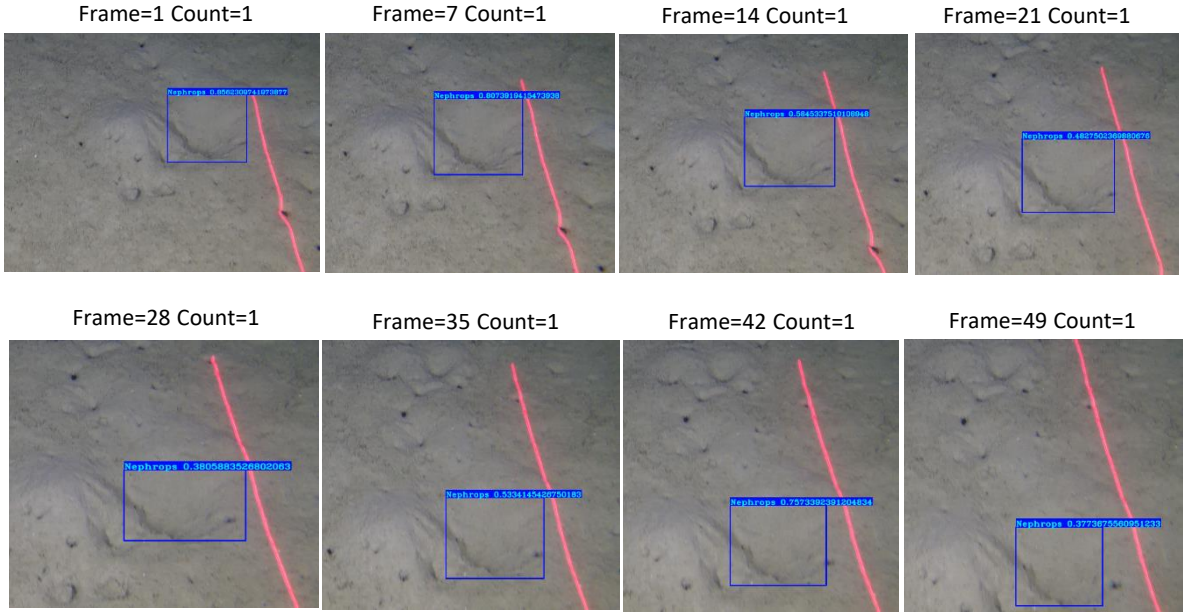


Figure 6.21: Nephrops burrows count in temporal segment 1 on the consecutive's frames.

Figure 6.22 also shows a similar result as discussed in the previous example. These frames are extracted from the 'RF09_Min3' temporal segment. The figure shows eight consecutive frames from the input video. The burrow is detected and tracked in each frame until it disappears from the visibility window. Each frame detects the burrow with a different confidence value and size. This is the main reason for the failure of a known tracking algorithm. The proposed spatial-temporal tracking algorithm tracks the burrow based on their spatial intersection values. As the figure shows, the detected burrows bounding boxes are different in size in each frame, but the algorithm can track them and count them as one burrow.

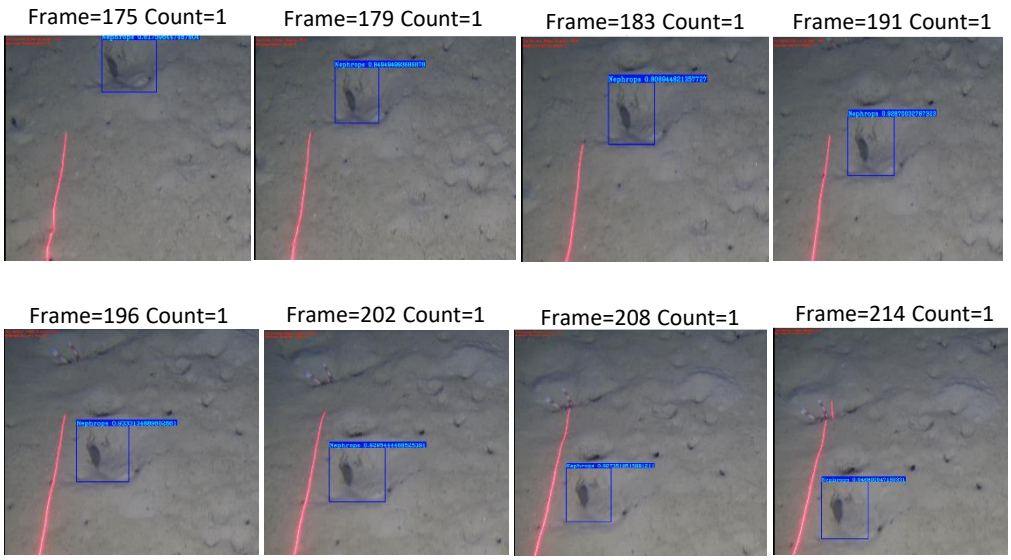


Figure 6.22: Nephrops burrows count in temporal segment 3 on the consecutive's frames.

Figure 6.23 shows the tracking results from the Boosting tracking algorithm. The results clearly show that the algorithm is tracking well in the first few frames, but as the camera starts moving, the algorithm loses the tracking and tracks some other parts of the frame. This leads to an inaccurate count of burrows.

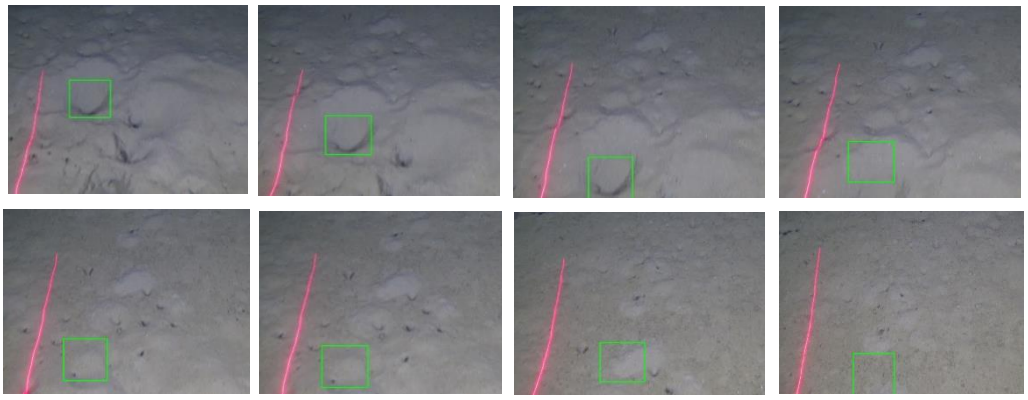


Figure 6.23: Nephrops burrows count in temporal segment 3 on the consecutive's frames.

Similarly, Figure 6.24 shows the results from the CSRT tracking algorithm. The burrows are tracking fine in the initial few frames. After that, the tracking algorithm lost its position and coordinates. The last four frames in the figure show the burrow's wrong tracking. CSRT is less effective than the boosting algorithm.

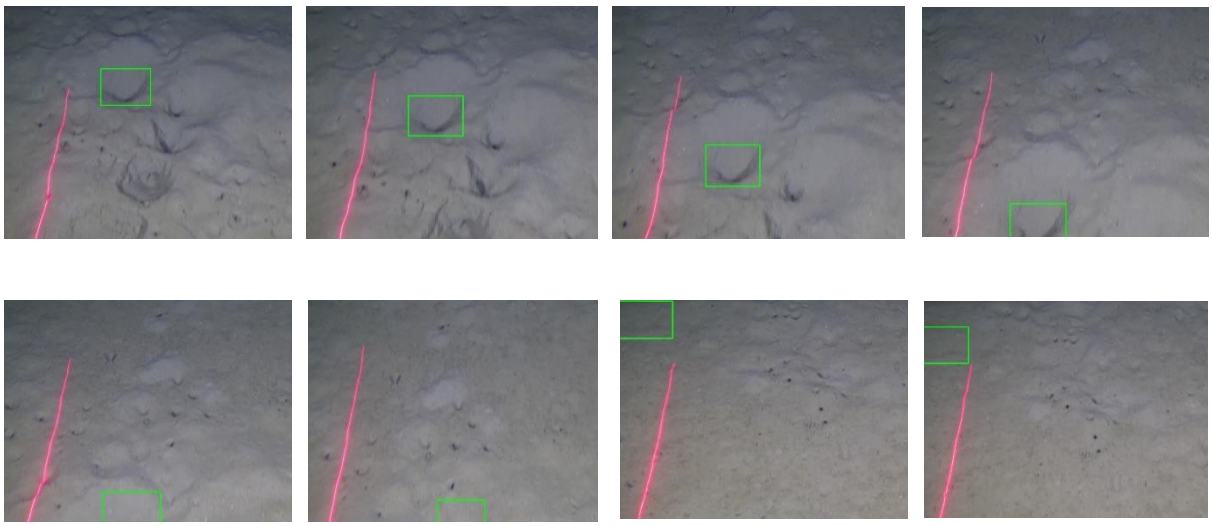


Figure 6.24: Nephrops burrows count using CSRT tracking algorithm.

Figure 6.25 shows the KCF tracking algorithm results. KCF lost the burrow information and his tracking, leading to the wrong count of burrows. In the later frames, the algorithm continuously tracked the burrow at the bottom of the window.

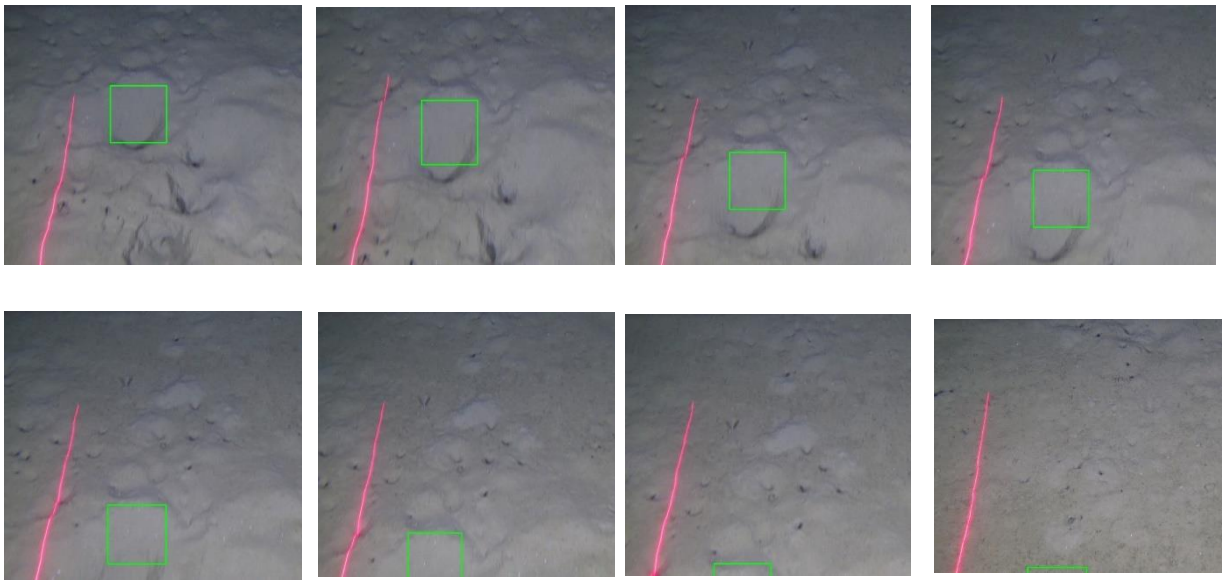


Figure 6.25: Nephrops burrows count using KCF tracking algorithm.

The median flow tracking algorithm shows promising results but loses the information in the later frames as other OpenCV tracking algorithms do. Figure 6.26 shows the tracking results obtained by the median flow tracking algorithm.

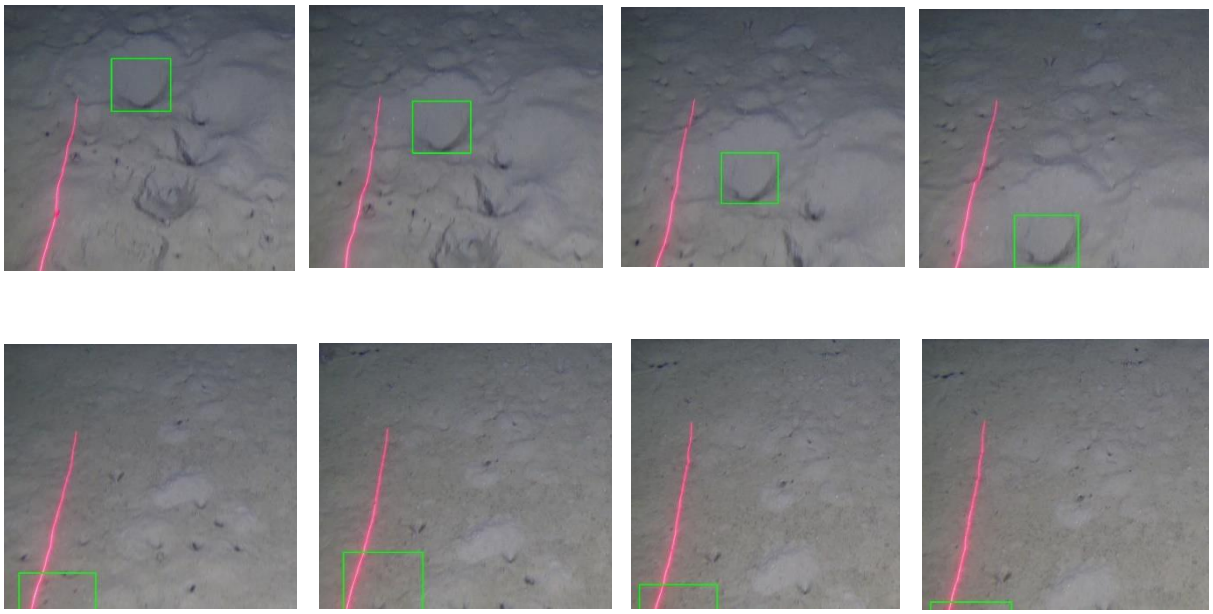


Figure 6.26: Nephrops burrows count using Median flow tracking algorithm.

The MIL tracking algorithm runs perfectly fine until the *Nephrops* burrow is visually present on the frames, but it loses the information and maps the tracking to the wrong place. Figure 6.27 shows the MIL tracking algorithm results.

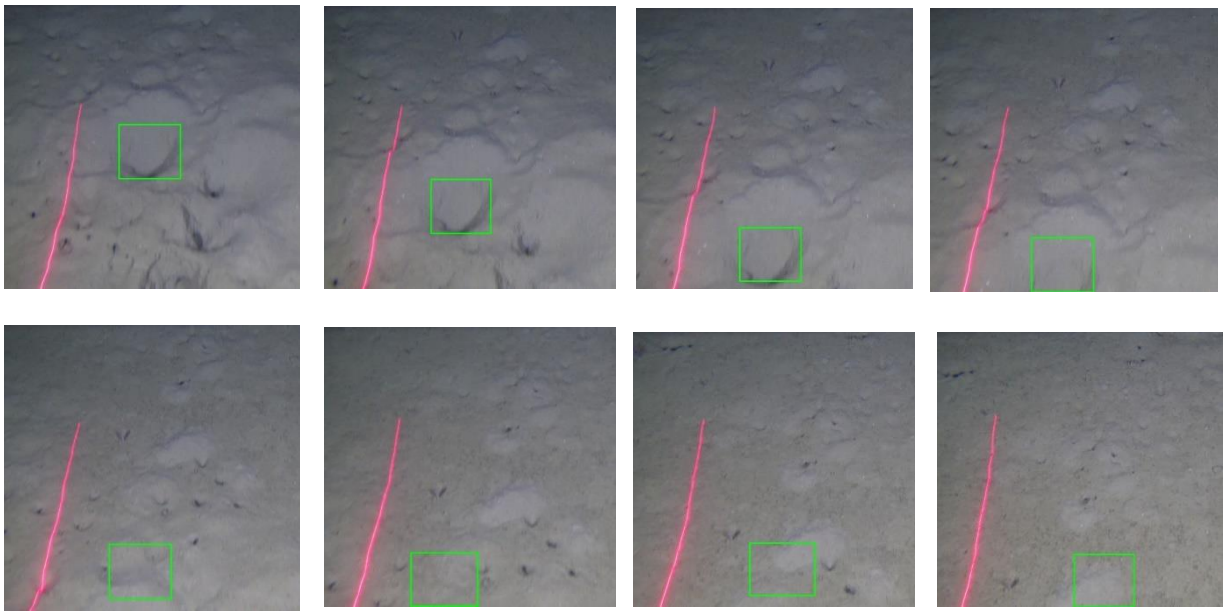


Figure 6.27: Nephrops burrows count using MIL tracking algorithm.

The Mosse tracking algorithm lost the information of the *Nephrops* burrow at the initial and lost the tracking. The figure below shows the results of the Mosse tracking algorithm.

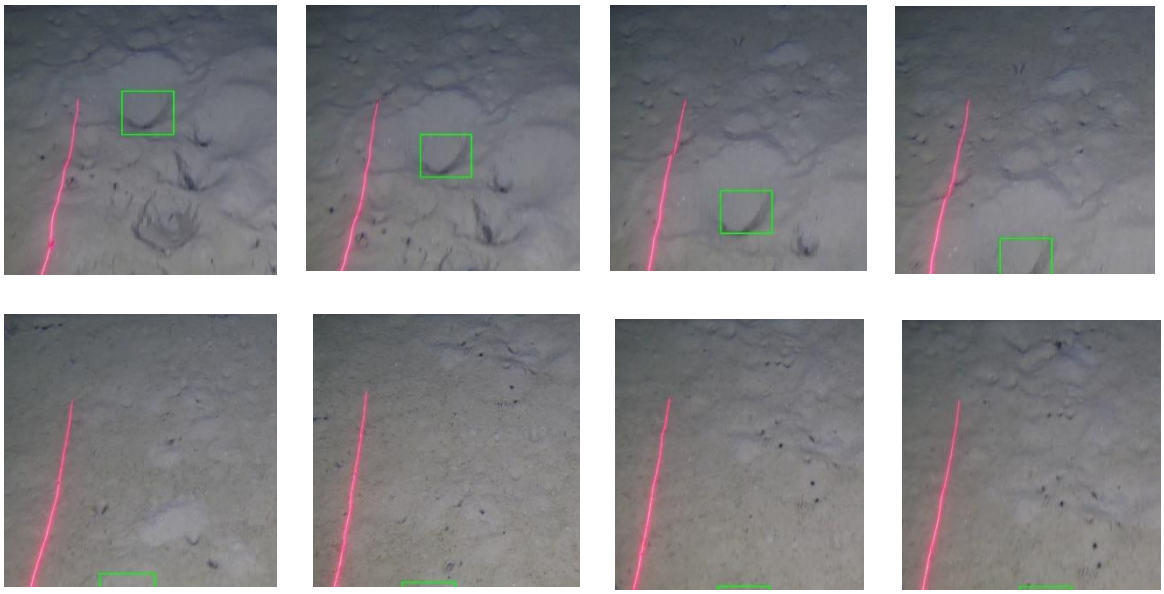


Figure 6.28: Nephrops burrows count using Mosse tracking algorithm.

Finally, the TLD tracking algorithm is applied to the same data. This algorithm cannot track the burrows properly from the initial frames, leading to an inaccurate count. Figure 6.29 shows the results of the TLD tracking algorithm

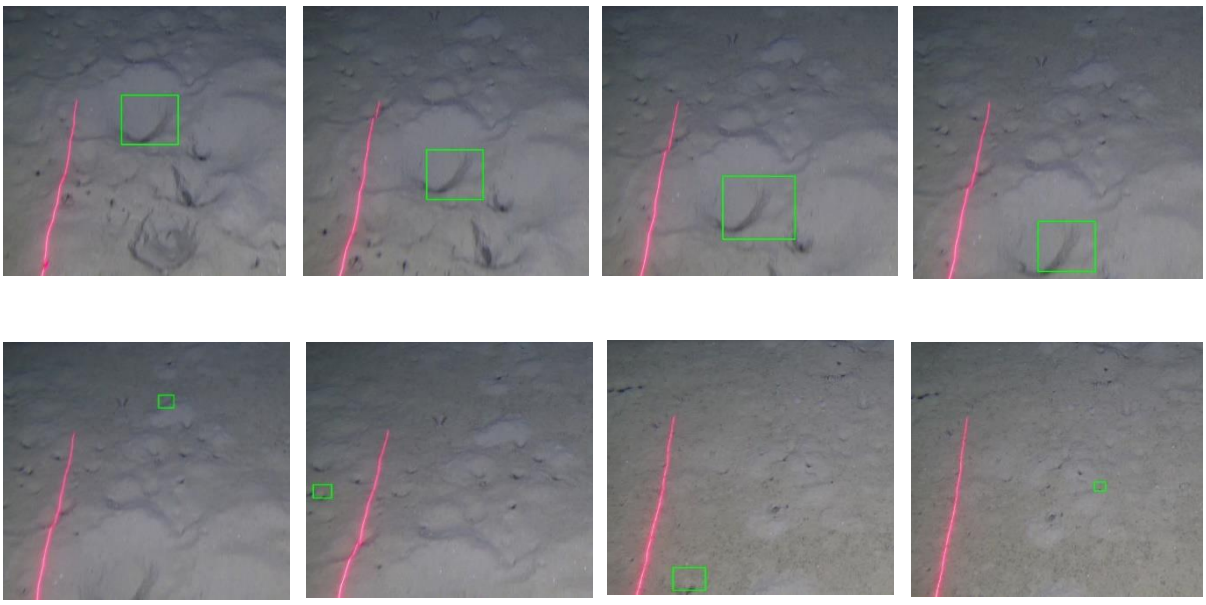


Figure 6.29: Nephrops burrows count using TLD tracking algorithm.

This page intentionally left blank.

Chapter 7: Conclusion and Future Work

'Our imagination is the only limit to what we can hope to have in the future'

Charles F. Kettering

7.1. Conclusions

Problems faced by marine scientists during the assessment of *Nephrops norvegicus* species during underwater TV surveys have been addressed in this thesis. One of the main contributions of the work has been the study of the behavior of deep learning algorithms on the complex underwater dataset of different FUs.

Currently, the *Nephrops* data are collected through the UWTV surveys and are reviewed manually by trained experts. Many of the data were difficult to process due to complex environmental conditions. Burrows systems are quantified following the protocol established by ICES. The image data (which refers to video or still data) for each station is reviewed independently by at least two experts, and the counts are recorded for each minute onto the log sheet records. Each row of the log sheet records the minute, the number of burrows system count, and the time stamp. Count data are screened to check for any unusual discrepancies using Lin's Concordance Correlation Coefficient (CCC) with a threshold of 0.5. Lin's CCC measures the ability of counters to precisely reproduce each other's counts on a scale of 0.5 to 1, where 1 is perfect concordance. Only stations with a threshold lower than 0.5 were reviewed again by the experts.

Our first contribution is to develop the dataset for the deep learning models. No such dataset exists that someone can use to validate the results. The data is collected from their yearly underwater surveys from FU 30 and FU 22 stations. After many revisions, the current work selected a few videos for annotation (the videos are selected with Marine experts based on the *Nephrops* burrows densities). The annotation process for *Nephrops* burrows is quite complex and one of the most time-consuming and sensitive jobs finished in many months. The work started with initial training about burrows from Marine experts before annotating. For annotation, the Microsoft VOTT image annotation tool is used. The Marine expert validates each annotation before

adding it to the dataset. This process took a long time as confirming every annotation is time-consuming and sensitive. After validating each annotation, a curated dataset is used for training and testing the model.

Different types of deep learning-based models have been finetuned and applied to the created dataset of FU 22 and FU 30. The work proposed five different neural networks, including MobileNet, Inception, ResNet50, ResNet101 and YOLOv3, for detecting *Nephrops* burrows. The work used transfer learning and finetuned these models for better accuracy. All the models are trained and tested with the different combinations of datasets. A complete methodology is proposed for automatically detecting *Nephrops* burrows using deep learning models. This work makes a significant advancement for all the groups working on the *Nephrops norvegicus* counting for stock assessment, where it is shown to detect and accurately count the *Nephrops* burrows automatically. Our results prove that deep learning algorithms are a valuable and effective strategy to help marine science experts assess the abundance of *Nephrops* norvegicus species when underwater video/image surveys are carried out yearly, following ICES recommendations. The automatic detection algorithms could replace the tedious and sometimes tricky manual and human review of data, which is nowadays the standard procedure, with the promise of better accuracy, coverage of more significant areas in sampling and higher consistency in the assessment.

Deep learning algorithms performed very well on the FU 22 and FU 30 datasets in identifying the burrows of *Nephrops norvegicus*. However, due to the complex nature of the underwater environment, generic CNN-based object detectors still face challenges in underwater object detection. These challenges include image blurring, texture distortion, color shift, and scale variation, which result in low precision and recall rates. To tackle this challenge, this thesis contributes by developing a Novel Detection Refinement Technique for Accurate Identification of *Nephrops norvegicus* burrows. The proposed technique is based on each detection's spatial–temporal value. The work detected the burrows in each frame in a video sequence and then obtained the spatial and temporal information across the multiple frames to refine the *Nephrops* burrows detections. The proposed algorithm helped suppress the FP burrows. It allowed us to find the missed TP detection, achieving better accuracy and tracking and counting burrows in a video sequence. When integrated with any detector, the proposed method consistently increased the performance. The performance was calculated using mAP. This mechanism helps marine science experts in the assessment of the abundance of this species.

Finally, another contribution lies in the tracking and counting of *Nephrops* burrows. Multiple OpenCV tracking algorithms are applied to track and count the burrows, but due to three significant challenges, these tracking algorithms fail to track the *Nephrops*

burrow. The first challenge is the camera's movement; our objects are not moving, but the camera is moving in the forward direction, leaving the object behind. The second challenge is the characteristics and size of burrows that are not fixed, and each new burrow can vary in size and other characteristics. The third challenge is the angle/opening of the burrow. Each burrow opening can vary in direction, and the angle of the burrow can also change due to these challenges. The traditional object-tracking mechanism is not very effective. We proposed the tracking and counting of *Nephrops* burrows using the spatial-temporal values of each burrow. The proposed spatial-temporal technique tracks each burrow based on its spatial and temporal values and counts the unique burrows. The unique burrows are counted using the intersection values of detected burrows in consecutive frames.

From an experimental point of view, our contribution lies in comparing *Nephrops* burrows detection with different models, the deep analytics and application of detection refinement algorithm by calculating the precision, recall and F1 score, and the comparison of the proposed tracking algorithm is also compared with the OpenCV tracking algorithms. All these experiments were performed for the different combinations of datasets and different levels of parameters. Results show that our approach has better results regarding *Nephrops* burrows detections, refinements, tracking and counting of *Nephrops* burrows.

7.2. Future work

Many adaptations, tests, and experiments have been left for the future due to lack of time and data (i.e. the experiments with accurate data are usually very time-consuming, and each data needs cross-validation from the Marine experts). Future work concerns deeper analysis of data and *Nephrops* systems.

In future work, the plan is to use a more extensive curated dataset from FU 22 and FU 30 areas with expert annotations to improve the training of the Deep Learning network and validate the algorithm with data from other regions, which usually shows different habitats and relation with other marine species, and in image processing point of view, also differences in image quality, video acquisition procedures, and background textures. At the same time, detection accuracy could be obtained using more dense object detection models and novel architectures. Also, I will use diverse datasets from UWTV surveys conducted in other *Nephrops* stocks in other countries.

The following areas could be explored in this study:

- Build a semi-auto annotation tool for *Nephrops* burrows that helps the Marine scientists in the UWTV survey to count and prepare the dataset for deep learning analysis.
- Explore the mechanism to identify and measure the size of individual *Nephrops* burrow entrances.
- Propose a solution to identify the *Nephrops* burrow systems from already identified burrows.
- Correlate the *Nephrops* burrow systems with each other to identify the number of complexes/systems.

Bibliography

[1] T. Rimavicius and A. Gelzinis, "A comparison of the deep learning methods for solving seafloor image classification task," Communications in Computer and Information Science, vol. 756, pp. 442–453, 2017.

[2] H. Qin, X. Li, Z. Yang and M. Shang, "When underwater imagery analysis meets deep learning: A solution at the age of big visual data," in Proc OCEANS'15 MTS/IEEE, Washington, DC, USA, pp. 1–5, 2015.

[3] M. Jiménez, I. Sobrino and F. Ramos, "Objective methods for defining mixed-species trawl fisheries in Spanish waters of the Gulf of cádiz," Fisheries Research, vol. 67, no. 2, pp. 195–206, 2004.

[4] FAO yearbook. Fishery and Aquaculture Statistics 2019/FAO annuaire. Statistiques des pêches et de l'aquaculture 2019/FAO anuario. Estadísticas de pesca y acuicultura 2019. Rome/Roma.

[5] Issifu, I., Alava, J.J., Lam, V.W., Sumaila, U.R. (2022). Impact of ocean warming, overfishing and mercury on European fisheries: A risk assessment and policy solution framework. *Front. Mar. Sci.* 8:770805. doi: 10.3389/fmars.2021.770805

[6] The State of Mediterranean and Black Sea Fisheries 2020. General Fisheries Commission for the Mediterranean. Rome. doi: 10.4060/cb2429en

[7] EU Council Regulation 2018/120 (2018, Jan 18). "Fixing for 2018 the fishing opportunities for certain fish stocks and groups of fish stocks, applicable in Union waters and, for union fishing vessels, in certain non-union waters, and amending regulation (EU) 2017/127", [Online]. Available: http://data.europa.eu/eli/reg/2018/120/oj.

[8] Fischer, W., G. Bianchi and W.B. Scott 1981 Lobsters. 5: pag.var. In FAO Species identification sheets for fishery purposes. Eastern Central Atlantic (fishing areas 34, 47; in part). Canada Funds-in-Trust. Ottawa, Department of Fisheries and Oceans Canada, by arrangement with the Food and Agriculture Organization of the United Nations. 1-7

[9] Bianchini, M.L., L.D. Stefano and S. Ragonese 1998 Siza and age at onset of sexual maturity of female Norway lobster *Nephrops norvegicus* L. (Crustacea: Nephropidae) in the Strait of Sicily (central Mediterranean Sea). Sci. Mar. 62(1-2):151-159.

[10] ICES. 2017, "Report of the Workshop on *Nephrops* Burrow Counting," WKNEPS 2016 Report, Reykjavík, Iceland. ICES CM 2016/SSGIEOM:34, International Council for the Exploration of the Sea, Copenhagen V, Denmark, pp. 9–11, November 2016.

[11] Bailey, N., Chapman, C.J., Alfonso-Dias, M., Turrell, W. (1995). The influence of hydrographic factors on *Nephrops* distribution and biology. ICES CM/Q:17, 13 pp.

[12] Tully, O., and Hillis, J.P. (1995). Causes and spatial scales of variability in population structure of *Nephrops norvegicus* (L.) in the Irish Sea. Fish. Res. 21, 329–347. doi: 10.1016/0165-7836(94)00303

[13] Maynou, F.X., and Sardà F. (1997). *Nephrops norvegicus* population and morphometrical characteristic in relation to substrate heterogeneity. Fish. Res. 30, 139–149. doi: 10.1016/S0165-7836(96)00549-8

[14] Farmer, A.S.D. (1974). Field assessments of diurnal activity in Irish Sea populations of the Norway Lobster *Nephrops norvegicus* (L.) (Decapoda: Nephropidae). Estuar. Coast. Mar. Sci. 2, 37–47. doi: 10.1016/0302-3524(74)90026-7

[15] ICES. 2017, "Report of the Workshop on *Nephrops* Burrow Counting," WGNEPS 2016 Report, Reykjavík, Iceland. ICES CM 2016/SSGIEOM:34, International Council for the Exploration of the Sea, Copenhagen V, Denmark, pp. 9–11, November 2016.

[16] A. Leocádio, A. Weetman and K. Wieland, "Using UWTV surveys to assess and advise on *Nephrops* stocks," ICES Cooperative Research Report, no. 340. pp. 49, 2018. [Online]. Available: DOI 10.17895/ices.pub.4370.

[17] ICES, 2021. [Online]. Available: https://www.ices.dk/about-ICES/Pages/default.aspx.

[18] Morello, E.B., Froglia, C., Atkinson, R.J.A. (2007). Underwater television as a fishery-independent method for stock assessment of Norway lobster (*Nephrops norvegicus*) in the central Adriatic Sea (Italy). ICES J. Mar. Sci. 64, 1116–1123. doi: 10.1093/icesjms/fsm082.

[19] Y. Vila, C. Burgos and M. Soriano, "*Nephrops* (FU 30) UWTV survey on the Gulf of Cadiz grounds," in Proc. ICES, Report of the Working Group for the Bay of Biscay and the Iberian waters Ecoregion (WGBIE), vol. 11, Copenhagen, Denmark, pp. 503, 2015.

[20] EU (2020). Council Regulation (EU) No. 123/2020 of 27 January 2020 fixing for 2020 the fishing opportunities for certain fish stocks and groups of fish stocks, applicable in Union waters and, for Union fishing vessels, in certain non-Union waters. https://eur-ex.europa.eu/eli/reg/2020/123/oj [accessed July 29, 2022]

[21] ICES 2007. Report of the Workshop on the use of UWTV surveys for determining abundance in *Nephrops* stocks throughout European waters (WKNEPHTV). ICES CM: 2007/ACFM: 14 Ref: LRC, PGCCDBS.

[22] ICES. 2008. Report of the Workshop and training course on *Nephrops* burrow identification (WKNEPHBID), 25-29 February 2008, Belfast, Northern Ireland, UK. ICES CM 2008/LRC:03. 44 pp.

[23] ICES. 2010. Report of the Study Group on *Nephrops* Surveys (SGNEPS), 9-11 November 2010, Lisbon, Portugal. ICES CM 2010/SSGESST:22. 95 pp.

[24] L. Linn, "A concordance correlation coefficient to evaluate reproducibility," Biometrics, vol. 45, no. 1, pp. 255–68, 1989.

[25] J. Doyle, C. Lordan, I. Hehir, R. Fitzgerald, O. Connor et al., "The 'smalls' *Nephrops* grounds (FU 22) 2013 UWTV survey report and catch options for 2014," Marine Institute UWTV Survey Report, Galway, Ireland, 2013.

[26] Microsoft CSE group. (2020, June 3), "Visual object tagging tool (VOTT), an electron app for building end to end object detection models from images and videos, v2.2.0. [Online]. Available: https://github.com/microsoft/VoTT.

[27] A. Ahvonen, J. Baudrier, H. Diogo, A. Dunton, A. Gordoa et al. "Working group on recreational fisheries surveys (WGRFS, outputs from 2019 meeting)," ICES Scientific Report, vol. 2, no. 1, pp. 1–86, 2020, [Online]. Available: DOI 10.17895/ices.pub.5744.

[28] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-cNN: Towards real-time object detection with region proposal networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137–1149, 2017.

[29] Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–15 December 2015; pp. 1440–1448..

[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the inception architecture for computer vision," in Proc. IEEE Conf. on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, pp. 2818–2826, 2016.

[31] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. Chen, "Mobilenetv2: inverted residuals and linear bottlenecks," in Proc. Conf. on Computer Vision and Pattern Recognition, Salt Lake City, UT, pp. 4510–4520, 2018.

[32] Understanding and Coding a ResNet in Keras. Available online: https://towardsdatascience.com/understanding-and-coding-a-resnet-in-keras-446d7ff84d33

[33] TensorFlow Core v2.8.0. Available online: https://www.tensorflow.org/api_docs/python/tf/keras/applications/resnet/ResNet101 (accessed on 20 March 2022).

[34] Redmon, Joseph and Ali Farhadi. "YOLOv3: An Incremental Improvement." *ArXiv* abs/1804.02767 (2018):

[35] https://www.ibm.com/in-en/cloud/learn/neural-networks

[36] https://www.wgu.edu/blog/neural-networks-deep-learning-explained2003.html#close

[37]https://machinelearningmastery.com/how-to-use-transfer-learning-when-developing-convolutional-neural-network-models/

[38] R. Girsshick, J. Donahue, T. Darrell and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, no. 1, pp. 142–158, 2016.

[39] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-cNN: Towards real-time object detection with region proposal networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137–1149, 2017.

[40] R. Shima, H. Yunan, O. Fukuda, H. Okumura, K. Arai et al. "Object classification with deep convolutional neural network using spatial information," in Proc. Int. Conf. on Intelligent Informatics and Biomedical Sciences (ICIIBMS), Okinawa, Japan, pp. 135–139, 2017.

[41] S. Soltan, A. Oleinikov, M. Demirci and A. Shintemirov, "Deep learning-based object classification and position estimation pipeline for potential use in robotized pick-and-place operations," Robotics, vol. 9, no. 3, 2020.

[42] S. Masubuchi, E. Watanabe, Y. Seo, S. Okazaki, K. Watanabe et al. "Deep-learning-based image segmentation integrated with optical microscopy for automatically searching for two-dimensional materials," npj 2D Mater Appl, vol. 4, no. 3, pp. 1–9, 2020.

[43] T. Lin, M. Maire, S. Belongie, J. Hays, P. Perona et al. "Microsoft COCO: common objects in context," in Proc. 13th European Conf. on Computer Vision, ECCC, vol 8693, Zurich, Switzerland, pp. 740–755, 2014.

[44] I. Sutskever, J. Martens, G. Dahl and G. Hinton, "On the importance of initialization and momentum in deep learning," in Proc. of the 30th Int. Conf. on Machine Learning (ICML-13), vol. 28, no. 3, Atlanta GA, USA, pp. 1139–1147, 2013.

[45] R. Pascanu, T. Mikolov and Y. Bengio. "On the difficulty of training recurrent neural networks," ArXiv Preprint, vol. 1211, 5063, pp. 1–12, 2012. [Online]. Available: https://arxiv.org/pdf/1211.5063.pdf.

[46] T. Tieleman and G. Hinton, "Divide the gradient by a running average of its recent magnitude," Neural Networks for Machine Learning, vol. 4, pp. 26–31, 2012.

[47] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6517-6525, doi: 10.1109/CVPR.2017.690.

[48] Bochkovskiy, Alexey, Chien-Yao Wang and Hong-Yuan Mark Liao. "YOLOv4: Optimal Speed and Accuracy of Object Detection." *ArXiv* abs/2004.10934 (2020):

[49] C. -Y. Wang, A. Bochkovskiy and H. -Y. M. Liao, "Scaled-YOLOv4: Scaling Cross Stage Partial Network," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 13024-13033, doi: 10.1109/CVPR46437.2021.01283.

[50] Huang, X.; Wang, X.; Lv, W.; Bai, X.; Long, X.; Deng, K.; Dang, Q.; Han, S.; Liu, Q.; Hu, X.; et al. PP-YOLOv2: A Practical Object Detector. *arXiv* **2021**, arXiv:2104.10419

[51] M. Zhang, C. Wang, J. Yang and K. Zheng, "Research on Engineering Vehicle Target Detection in Aerial Photography Environment based on YOLOX," 2021 14th International Symposium on Computational Intelligence and Design (ISCID), 2021, pp. 254-256, doi: 10.1109/ISCID52796.2021.00066.

[52] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen et al., "Tensorflow: large-scale machine learning on heterogeneous distributed systems," in Proc. 12th USENIX Conf. on Operating Systems Design and Implementation, Savannah, GA, USA, pp. 265–283, 2016. 5344 CMC, 2022, vol.70, no.3

[53] M. Everingham, L. Van Gool, C. Williams, J. Winn and A. Zisserman, "The pascal visual object classes (VOC) challenge," International Journal of Computer Vision, vol. 88, pp. 303–338, 2010.

[54] Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893.

[55] Dollár, P.; Appel, R.; Belongie, S.; Perona, P. Fast feature pyramids for object detection. IEEE Trans. Pattern Anal. Mach. Intell. 2014, 36, 1532–1545.

[56] Dollár, P.; Tu, Z.; Perona, P.; Belongie, S. Integral Channel Features; BMVC Press: Sussex, UK, 2009.

[57] Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D.; Ramanan, D. Object Detection with Discriminatively Trained Part-Based Models. IEEE Trans. Pattern Anal. Mach. Intell. 2010, 32, 1627–1645.

[58] Song, H.A.; Lee, S.-Y. Hierarchical representation using NMF. In International Conference on Neural Information Processing; Lee, M., Hirose, A., Hou, Z.-G., Kil, R.M., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; Volume 8226, pp. 466–473.

[59] Chan, A.B.; Morrow, M.; Vasconcelos, N. Analysis of crowded scenes using holistic properties. In Proceedings of the Performance Evaluation of Tracking and Surveillance workshop at CVPR, Miami, FL, USA, 2009; Available online: http://visal.cs.cityu.edu.hk/static/pubs/workshop/pets09-crowds.pdf (accessed on 20 March 2022).

[60] Saqib, M.; Khan, S.D.; Blumenstein, M. Texture-based feature mining for crowd density estimation: A study. In Proceedings of the 2016 International Conference on Image and Vision Computing New Zealand (IVCNZ), Palmerston North, New Zealand, 21–22 November 2016; pp. 1–6.

[61] Zhang, C.; Li, H.; Wang, X.; Yang, X. Cross-scene crowd counting via deep convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 833–841.

[62] Chan, A.B.; Vasconcelos, N. Modeling, Clustering, and Segmenting Video with Mixtures of Dynamic Textures. IEEE Trans. Pattern Anal. Mach. Intell. 2008, 30, 909–926.

[63] Girshick, R.B.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 23–28 June 2014.

[64] Saqib, M.; Khan, S.D.; Blumenstein, M. Texture-based feature mining for crowd density estimation: A study. In Proceedings of the 2016 International Conference on Image and Vision Computing New Zealand (IVCNZ), Palmerston North, New Zealand, 21–22 November 2016; pp. 1–6.

[65] Girshick, R.B.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 23–28 June 2014.

[66] Uijlings, J.R.R.; van de Sande, K.E.A.; Gevers, T.; Smeulders, A.W.M. Selective Search for Object Recognition. Int. J. Comput. Vis. 2013, 104, 154–171.

[67] Li, X.; Shang, M.; Qin, H.; Chen, L. Fast accurate fish detection and recognition of underwater images with fast R-CNN. In Proceedings of the OCEANS 2015—MTS/IEEE, Washington, DC, USA, 19–22 October 2015; pp. 1–5.

[68] Villon, S.; Chaumont, M.; Subsol, G.; Villéger, S.; Claverie, T.; Mouillot, D. Coral reef fish detection and recognition in underwater videos by supervised machine learning: Comparison between deep learning and HOG+SVM methods. In Advanced Concepts for Intelligent Vision Systems; Blanc-Talon, J., Distante, C., Philips, W., Popescu, D., Scheunders, P., Eds.; Springer: Cham, Switzerland, 2016; Volume 10016, pp. 160–171.

[69] Rathi, D.; Jain, S.; Indu, S. Underwater Fish Species Classification using Convolutional Neural Network and Deep Learning. arXiv 2018, arXiv:1805.10106.

[70] Xu, W.; Matzner, S. Underwater Fish Detection Using Deep Learning for Water Power Applications. In Proceedings of the 2018 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 13–15 December 2018; pp. 313–318.

[71] Mandal, R.; Connolly, R.M.; Schlacher, T.A.; Stantic, B. Assessing fish abundance from underwater video using deep neural networks. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–6.

[72] Gundam, M.; Charalampidis, D.; Ioup, G.; Ioup, J.; Thompson, C. Automatic fish classification in underwater video. Proc. Gulf. Caribb. Fish. Inst. 2015, 66, 276282.

[73] Jalal, A.; Salman, A.; Mian, A.; Shortis, M.; Shafait, F. Fish detection and species classification in underwater environments using deep learning with temporal information. In Ecological Informatics; Elsevier: Amsterdam, The Netherlands, 2020; Volume 57, p. 101088. ISSN 1574-9541.

[74] Sung, M.; Yu, S.C.; Girdhar, Y. Girdhar Vision based real-time fish detection using convolutional neural network. In Proceedings of the OCEANS 2017-Aberdeen, Aberdeen, UK, 19–22 June 2017; pp. 1–6.

[75] Jäger, J.; Rodner, E.; Denzler, J.; Wolff, V.; Fricke-Neuderth, K. Seaclef 2016: Object proposal classification for fish detection in underwater videos. In Proceedings of the Conference and Labs of the Evaluation Forum (CLEF), Évora, Portugal, 5–8 September 2016; Volume 1609, pp. 481–489.

[76] Zhuang, P.; Xing, L.; Liu, Y.; Guo, S.; Qiao, Y. Marine Animal Detection and Recognition with Advanced Deep Learning Models. In Proceedings of the Conference and Labs of the Evaluation Forum (CLEF), Dublin, Ireland, 11–14 September 2017.

[77] Zhao, Z.; Liu, Y.; Sun, X.; Liu, J.; Yang, X.; Zhou, C. Composited FishNet: Fish Detection and Species Recognition From Low-Quality Underwater Videos. IEEE Trans. Image Process. 2021, 30, 4719–4734.

[78] Labao, A.B.; Naval, P.C., Jr. Cascaded deep network systems with linked ensemble components for underwater fish detection in the wild. Ecol. Inform. 2019, 52, 103–112.

[79] Salman, A.; Siddiqui, S.A.; Shafait, F.; Mian, A.; Shortis, M.R.; Khurshid, K.; Ulges, A.; Schwanecke, U. Automatic fish detection in underwater videos by a deep neural network-based hybrid motion learning system. ICES J. Mar. Sci. 2019, 77, 1295–1307.

[80] Dieleman, S. Classifying Planktons with Deep Neural Networks. Available online: http://benanne.github.io/.2015/03/17/plankton.html (accessed on 22 March 2022).

[81] Lee, H.; Park, M.; Kim, J. Plankton classification on imbalanced large scale database via convolutional neural networks with transfer learning. In Proceedings of the IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, 25–28 September 2016; pp. 3713–3717.

[82] Shiela, M.M.A.; Soriano, M.; Saloma, C. Classification of coral reef images from underwater video using neural networks. Opt. Express 2005, 13, 8766–8771.

[83] Elawady, M. Sparsem: Coral Classification Using Deep Convolutional Neural Networks. Master's Thesis, Hariot-Watt University, Edinburgh, UK, 2014.

[84] L. Boominathan, S. S. Kruthiventi, and R. V. Babu, "Crowdnet: A deep convolutional network for dense crowd counting," in Proceedings of the 2016 ACM on Multimedia Conference. ACM, 2016, pp. 640–644.

[85] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards realtime object detection with region proposal networks," in Advances in neural information processing systems, 2015, pp. 91–99.

[86] M. Modasshir, S. Rahman, O. Youngquist and I. Rekleitis, "Coral Identification and Counting with an Autonomous Underwater Vehicle," 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), 2018, pp. 524-529, doi: 10.1109/ROBIO.2018.8664785.

[87] Wageeh, Y., Mohamed, H.ED., Fadl, A. et al. YOLO fish detection with Euclidean tracking in fish farms. J Ambient Intell Human Comput 12, 5–12 (2021).

[88] R. Tanaka, T. Nakano and T. Ogawa, "Sequential Fish Catch Counter Using Vision-based Fish Detection and Tracking," OCEANS 2022 - Chennai, 2022, pp. 1-5, doi: 10.1109/OCEANSChennai45887.2022.9775327.

[89] G. S. Gaude and S. Borkar, "Fish Detection And Tracking For Turbid Underwater Video," 2019 International Conference on Intelligent Computing and Control Systems (ICCS), 2019, pp. 326-331, doi: 10.1109/ICCS45141.2019.9065425.

[90] Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893.

[91] Dollár, P.; Appel, R.; Belongie, S.; Perona, P. Fast feature pyramids for object detection. IEEE Trans. Pattern Anal. Mach. Intell. 2014, 36, 1532–1545.

[92] Dollár, P.; Tu, Z.; Perona, P.; Belongie, S. Integral Channel Features; BMVC Press: Sussex, UK, 2009.

[93] Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D.; Ramanan, D. Object Detection with Discriminatively Trained Part-Based Models. IEEE Trans. Pattern Anal. Mach. Intell. 2010, 32, 1627–1645.

[94] Song, H.A.; Lee, S.-Y. Hierarchical representation using NMF. In International Conference on Neural Information Processing; Lee, M., Hirose, A., Hou, Z.-G., Kil, R.M., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; Volume 8226, pp. 466–473.

[95] L. Boominathan, S. S. Kruthiventi, and R. V. Babu, "Crowdnet: A deep convolutional network for dense crowd counting," in Proceedings of the 2016 ACM on Multimedia Conference. ACM, 2016, pp. 640–644.

[96] V. Lempitsky and A. Zisserman, "Learning to count objects in images," in Advances in neural information processing systems, 2010, pp. 1324– 1332.

[97] V.-Q. Pham, T. Kozakaya, O. Yamaguchi, and R. Okada, "Count forest: Co-voting uncertain number of targets using random forest for crowd density estimation," in Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3253–3261.

[98] B. Xu and G. Qiu, "Crowd density estimation based on rich features and random projection forest," in Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on. IEEE, 2016, pp. 1–8.

[99] D. B. Sam, S. Surya, and R. V. Babu, "Switching convolutional neural network for crowd counting," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 1, no. 3, 2017, p. 6.

[100] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 589–597.

[101] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards realtime object detection with region proposal networks," in Advances in neural information processing systems, 2015, pp. 91–99.

[102] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 779– 788.

[103] C. Gao, P. Li, Y. Zhang, J. Liu, and L. Wang, "People counting based on head detection combining adaboost and cnn in crowded surveillance environment," Neurocomputing, vol. 208, pp. 108–116, 2016.

[104] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in Computer Vision and Pattern Recognition, 2014.

[105] Bochkovskiy, A.; Wang, C.Y.; Liao, H.Y.M. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv 2020, arXiv:2004.10934.

[106] Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. Ssd: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.

[107] He, K., Zhang, X., Ren, S., Sun, J. (2014). Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds) Computer Vision – ECCV 2014. ECCV 2014.

[108] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and ´ S. Belongie. Feature pyramid networks for object detection. In CVPR, 2017.

[109] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," arXiv preprint arXiv:1312.6229, 2013.

[110] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in European conference on computer vision. Springer, 2016, pp. 21–37

[111] T.-Y. L. P. G. Ross and G. K. H. P. Dollar, "Focal loss for dense object ´ detection."

[112] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In International Conference on Machine Learning, pages 6105–6114. PMLR, 2019

[113] Bochkovskiy, Alexey, Chien-Yao Wang and Hong-Yuan Mark Liao. "YOLOv4: Optimal Speed and Accuracy of Object Detection." *ArXiv* abs/2004.10934 (2020):

[114] C. -Y. Wang, A. Bochkovskiy and H. -Y. M. Liao, "Scaled-YOLOv4: Scaling Cross Stage Partial Network," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 13024-13033, doi: 10.1109/CVPR46437.2021.01283.

[115] Huang, X.; Wang, X.; Lv, W.; Bai, X.; Long, X.; Deng, K.; Dang, Q.; Han, S.; Liu, Q.; Hu, X.; et al. PP-YOLOv2: A Practical Object Detector. arXiv 2021, arXiv:2104.10419

[116] M. Zhang, C. Wang, J. Yang and K. Zheng, "Research on Engineering Vehicle Target Detection in Aerial Photography Environment based on YOLOX," 2021 14th International Symposium on Computational Intelligence and Design (ISCID), 2021, pp. 254-256, doi: 10.1109/ISCID52796.2021.00066.

[117] Li, D, Miao, Z, Peng, F, et al. Automatic counting methods in aquaculture: A review. J World Aquac Soc. 2020; 1– 15. https://doi.org/10.1111/jwas.12745

[118] Solahudin, M., Slamet, W., & Dwi, A. S. (2018). Vaname (Litopenaeus vannamei) shrimp fry counting based on image processing method. Earth and Environmental Science, 147(1), 2014.

[119] Labuguen, R. T., Volante, E. J. P., Causo, A., Bayot, R., Peren, G., Macaraig, R. M., & Tangonan, G. L. (2012). Automated fish fry counting and schooling behavior analysis using computer vision. In 2012 IEEE 8th International Colloquium on Signal Processing and its Applications (pp. 255–260). IEEE.

[120] Jing, D., Han, J., Wang, X., Wang, G., Tong, J., Shen, W., & Zhang, J. (2017). A method to estimate the abundance of fish based on dual-frequency identification sonar (DIDSON) imaging. Fisheries Science, 83(5), 685–697.

[121] Newbury, P. F., Culverhouse, P. F., & Pilgrim, D. A. (1995). Automatic fish population counting by artificial neural network. Aquaculture, 133(1), 45–55.

[122] Fan, L., & Liu, Y. (2013). Automate fry counting using computer vision and multi-class least squares support vector machine. Aquaculture, 380,91–98.

[123] Lau, P. Y., Correia, P. L., Fonseca, P., & Campos, A. (2012). Estimating Norway lobster abundance from deep-water videos: An automatic approach. IET Image Processing, 6(1), 22–30.

[124] Sharif, M. H., Galip, F., Guler, A., & Uyaver, S. (2015). A simple approach to count and track underwater fishes from videos. In 2015 18th International Conference on Computer and Information Technology (ICCIT) (pp. 347–352). IEEE.

[125] Chuang, M. C., Hwang, J. N., Williams, K., & Towler, R. (2015). Tracking live fish from low-contrast and low-frame-rate stereo videos. IEEE Transactions on Circuits and Systems for Video Technology. IEEE Transactions on Circuits and Systems for Video Technology, 25(1).s

[126] Huang, T. W., Hwang, J. N., Romain, S., & Wallace, F. (2019). Fish tracking and segmentation from stereo videos on the wild sea surface for electronic monitoring of rail fishing. IEEE Transactions on Circuits and Systems for Video Technology, 29 (10), 3146-3158.

[127] Spampinato, C, Jessica ChenBurger, Gaya Nadarajan, & Bob Fisher. (2008). Detecting, tracking and counting fish in low quality unconstrained underwater videos. Proceedings of the International Conference on Computer Vision Theory & Applications, s2, 514–519.

[128] M. Modasshir, S. Rahman, O. Youngquist and I. Rekleitis, "Coral Identification and Counting with an Autonomous Underwater Vehicle," 2018 IEEE International Conference on Robotics and Biomimetics (ROBIO), 2018, pp. 524-529, doi: 10.1109/ROBIO.2018.8664785.

[129] https://ehsangazar.com/object-tracking-with-opencv-fd18ccdd7369

[130] Hussam El-Din Mohamed, Ali Fadl, Omar Anas, Youssef Wageeh, Noha ElMasry, Ayman Nabil, Ayman Atia, MSR-YOLO: Method to Enhance Fish Detection and Tracking in Fish Farms, Procedia Computer Science, Volume 170, 2020, Pages 539-546, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.03.123.

[131] Wageeh, Y., Mohamed, H.ED., Fadl, A. et al. YOLO fish detection with Euclidean tracking in fish farms. J Ambient Intell Human Comput 12, 5–12 (2021).

[132] X. Li, Z. Wei, L. Huang, J. Nie, W. Zhang and L. Wang, "Real-Time Underwater Fish Tracking Based on Adaptive Multi-Appearance Model," 2018 25th IEEE International Conference on Image Processing (ICIP), 2018, pp. 2710-2714, doi: 10.1109/ICIP.2018.8451469.

[133] R. Tanaka, T. Nakano and T. Ogawa, "Sequential Fish Catch Counter Using Vision-based Fish Detection and Tracking," OCEANS 2022 - Chennai, 2022, pp. 1-5, doi: 10.1109/OCEANSChennai45887.2022.9775327.

[134] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, “Simple online and real-time tracking.” Proc. 2016 IEEE International Conference on Image Processing (ICIP2016), pp.3464–3468, Sept. 2016.

[135] G. S. Gaude and S. Borkar, "Fish Detection And Tracking For Turbid Underwater Video," 2019 International Conference on Intelligent Computing and Control Systems (ICCS), 2019, pp. 326-331, doi: 10.1109/ICCS45141.2019.9065425.

[136] Sokolova, M., Thompson, F., Mariani, P., & Krag, L. A. (2021). Towards sustainable demersal fisheries: NepCon image acquisition system for automatic Nephrops norvegicus detection. PLOS ONE, 16(6), e0252824. https://doi.org/10.1371/journal.pone.0252824.

[137] Avsar, E., Feekings, J. P., & Krag, L. A. (2023). Estimating catch rates in real time: Development of a deep learning based Nephrops (Nephrops norvegicus) counter for demersal trawl fisheries. Frontiers in Marine Science, 10, 1129852. https://doi.org/10.3389/fmars.2023.1129852.

Appendix A: *Curriculum Vitae*

Atif Naseer

Experience

Faculty member, Science and Technology Unit, Umm Al Qura University, Makkah Al Mukarammah, Saudi Arabia	(2012–date)
Faculty member, Department of Computing, Riphah International University, Islamabad, Pakistan	(2006–2012)

Education

PhD in Telecommunication Engineering (Dissertation submitted) University of Malaga, Málaga, Spain	(2017–2023)
Master in Software Engineering National University of Sciences and Technology, Islamabad, Pakistan	(2007–2010)
Bachelor in Software Engineering University of Engineering and Technology, Taxila, Pakistan	(2001–2005)

Training / certifications

Vision Understanding and Machine Intelligence (VISUM)	2019
NI Software Radio Peripherals (USRP) Software Defined Radio, Saudi Arabia	2019
NI LabVIEW Communications, Saudi Arabia	2019
Developing Applications with the Hortonworks Data Platform using Java	2014

Publications

1. Atif Naseer, Enrique Nava Baro, Sultan Daud Khan, and Yolanda Vila. 2022. "A Novel Detection Refinement Technique for Accurate Identification of *Nephrops norvegicus* Burrows in Underwater Imagery". *Sensors* 22, no. 12: 4441. https://doi.org/10.3390/s22124441.
2. Atif Naseer, Baro, E. N., Khan, S. D., Vila, Y., Doyle, J. (2022). "Automatic Detection of *Nephrops norvegicus* Burrows from Underwater Imagery Using Deep Learning". CMC-*Computers, Materials & Continua*, 70(3), 5321–5344.
3. Aguzzi, J., Aristegui-Ezquibela, M., Burgos, C., Doyle, J., Atif Naseer., O'Connor, J., Pereira, B., Silva, C., Sköld, M., Vacherot, J-P., Vila, Y., Weetman, A., & Wieland, K. (2022). Working Group on *Nephrops* Surveys (WGNEPS; outputs from 2021). International Council for the Exploration of the Sea (ICES). ICES Scientific Report Vol. 4 No. 29 . https://doi.org/10.17895/ices.pub.19438472
4. Working Group on *Nephrops* Surveys (WGNEPS outputs from 2020): 1-114 (2021) ICES Scientific Reports 3(36): 1-114 (2021).
5. Aristegui-Ezquibela, M., Aguzzi, J., Burgos, C., Doyle, J., Fallon, N., Fifas, S., Jónasson, J., Jonsson, P., Lundy, M., Martinelli, M., Masmitja, I., McAllister, G., Medvešek, D., Atif Naseer, Reeve, C., Silva, C., Simon, J., Vacherot, J-P., Vigo-Fernandez, M., ... Wieland, K. (2021). Working Group on *Nephrops* Surveys (WGNEPS ; outputs from 2020). International Council for the Exploration of the Sea (ICES). ICES Scientific Report Vol. 3 No. 36 https://doi.org/10.17895/ices.pub.8041
6. Aristegui-Ezquibela, M. [et al.]. "ICES. 2020. Working Group on *Nephrops* Surveys (WGNEPS; outputs from 2019). ICES Scientific Reports". 2020
7. Atif. Naseer, E. N. Baro, S. D. Khan and Y. V. Gordillo, "Automatic Detection of *Nephrops norvegicus* Burrows in Underwater Images Using Deep Learning," 2020 Global Conference on Wireless and Optical Technologies (GCWOT), 2020

Patent

1. Emad Felemban, Sultan Daud Khan, Atif Naseer, Faizan Ur Rehman, Saleh Basalamah, “Deep learning framework for congestion detection and prediction in human crowds”, US20220254162A1, 11 August 2022.