IDENTIFICATION OF HIGH-FREQUENCY PERIODIC ACOUSTIC TAGS WITH DEEP LEARNING

by

Santosh Kumar Medisetty

Submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
June 2021

This thesis is dedicated to my family who have been the great support to me for this long journey.

Table of Contents

List of Tables

List of Figures

Abstract

Marine life researchers use the concept of fish tracking to determine the activity and behaviour of fish. It allows the researchers to recognize the valuable biological and physical support systems required by fish species at various life stages. There have been several advancements in fish tracking using different approaches that include electronic tags and acoustic tags. In acoustic fish tracking, the fish are equipped with acoustic tags that are either externally attached or surgically implanted into the fish. These tags can be tracked by a receiver placed within a range underwater. The tags emit high-frequency signals omnidirectionally at regular intervals, and these signals are recorded as pings in the receiver. The tags are identified by the delay between the pulses, which is typically between 1 second and 10 seconds. The tags can also be configured to emit two closely spaced pulses providing a new way to identify the tags.

The pings recorded by the receiver are represented as acoustic time series data that is analyzed to determine which tag the ping originated from. This analysis of identifying the pings (marking) from high-frequency tags which use a double pulse encoding scheme is currently done at Innovasea with the help of a visual analytics system called 'MarkTags', where the marking is done automatically within the software, or the user manually tunes the data to a particular tag period and marks the pings which consume much manual effort. We proposed a machine learning solution for identifying the tags using deep learning.

Our work discusses a novel approach to finding the pings by segmenting the images created from the pings. The pings are created as images using a 2D-histogram approach in which the pings are represented as pixels in an image. These images are segmented for the pings from the tag using a neural network called UNet. We developed a model that is trained on data recorded in different conditions. When the trained model was asked to identify the tags not seen during training, it could do so with an accuracy of over 95% and is close to human-level annotations. Our experimental results show that the machine learning approach matches the human-level performance, which can replace human intervention to a great extent.

Acknowledgements

I am extremely thankful to DeepSense, Innovasea, and Mitacs for funding my master's program and for providing me an opportunity to work on this project. This work was supported by Mitacs through the Mitacs Accelerate program. In addition to Mitacs, this research was enabled in part by support provided by Innovasea Inc.(www.innovasea.com) and DeepSense (www.deepsense.ca) in the form of the aforementioned Mitacs Accelerate program. Computations were performed on the DeepSense (deepsense.ca) high-performance computing platform. DeepSense is funded by ACOA, the Province of Nova Scotia, the Centre for Ocean Ventures and Entrepreneurship (COVE), IBM Canada Ltd. and the Ocean Frontier Institute (OFI).

I consider myself lucky to have worked under the supervision of Dr. Stan Matwin. I wish to express my sincere thanks and gratitude to my co-supervisor Dr. Oliver Kirsebom for constantly supporting me throughout my thesis and providing useful insights in doing my research. I am grateful to him for guiding and supporting me for my entire project.

I wish to thank Dr. Christopher Whidden for always providing ideas and useful suggestions for my project. Special thanks to Jennifer LaPlante, Dr. Lu Yang, Dr. Geetika Bhatia, and Dr. Jason Newport from DeepSense for constantly supporting my project.

I would like to thank Jean Quirion and Frank Smith of Innovasea Inc. for offering me an opportunity to work on this project.

Chapter 1

Introduction

The main objective of this thesis is to determine if deep learning can be used to identify the high-frequency periodic acoustic fish tags and optimize the methods to identify the tags.

1.1 Fish Tracking

Aquatic organisms like fish exhibit a variety of behaviours and reactions to factors present in the marine environment[1]. Our ability to predict organism responses to these alterations will rely on knowledge of animal movements. The tracking of fish movements allows scientists to recognize valuable biological and physical support systems required by fish species at various life stages[2] and also to assess fish responses to environmental changes. Though fish tracking gives information on fish migrations due to environmental changes, human disturbances like motorboat noise can also impact fish movements[3]. This might cause the fish to cease moving or hide in shelters[4]. Apart from the ecological insights, fish tracking can also provide information about the abiotic data from the regions that are otherwise impossible to collect[5]. Fish tracking also has several other uses. E.g., to estimate the natural mortality and exploitation rates [6], study behavioural and migration patterns. The process of tracking these movements of fish in any aquatic environment using different types of sensors like electronic tags or acoustic tags is called fish tracking.

The scientific study of fish has a long history. Initially, the tracking was only limited to larger organisms like marine mammals[7]. With the development of low-cost tiny tags, fixed receivers, and coded signals, it is now possible to track the movements of small fish across vast distances in freshwater[8]. Over the years, there have been several advancements in fish tracking using different approaches. Most of them include tracking using electronic tags and acoustic tags[8]. Moreover, in the

last 20 years, the advancements in technology made the researchers shift to long-term passive telemetry as opposed to active telemetry [9]. In passive telemetry that involves fish tags, an array of hydrophones is deployed in water and used to track the tagged fish. The receivers have internal memory and can record the data over a long period. A single hydrophone is used in active telemetry, and it is lowered into the water from the vessel. Whenever a transmitter is detected, the position of the animal can be determined with high precision [10]. Both active and passive telemetry have advantages in their own way. Even, there was a significant rise in the number of publications related to fish tracking.

Innovasea, a leading ocean tech company located in Halifax, Nova Scotia, works on fish tracking technology that employs different types of tags with different coding schemes. In this thesis, we discuss one of the approaches Innovasea currently uses to identify fish tags, their limitations, and our solution to their problem.

1.2 Limitations of Innovasea's Approach and Our Solution

In acoustic fish tracking, the fishes are equipped with acoustic tags either externally attached to the fish or surgically implanted into the fish. These tags emit the data omnidirectionally in the form of pulses. Innovasea uses different types of tags with different coding schemes. A new type of tag called HTI acoustic tag(developed by Hydroacoustic Technology Inc.) is used with an encoding scheme called pulse rate encoding scheme, also known as pulse interval encoding scheme. In this encoding scheme, the time interval between the consecutive pulses is controlled with the help of a microcontroller present in the tag and this time interval between the pulses makes the tags unique. There are several other encoding schemes like PPM and BPSK(Innovasea's HR system) used in other types of tags at Innovasea, where the information of the tags is encoded in the time intervals between successive pulses(in a series of pulses) in PPM and in the phase of the transmitted signal in BPSK. These tags can be detected by a receiver placed within a range underwater. The signals received at the receiver contain the direct signals from the fish or by the reflections of signals from surrounding structures or water surface and contain noise due to external factors, which will be explained in detail in the preliminaries chapter. The signals transmitted by the tags are recorded as pings in the receiver.

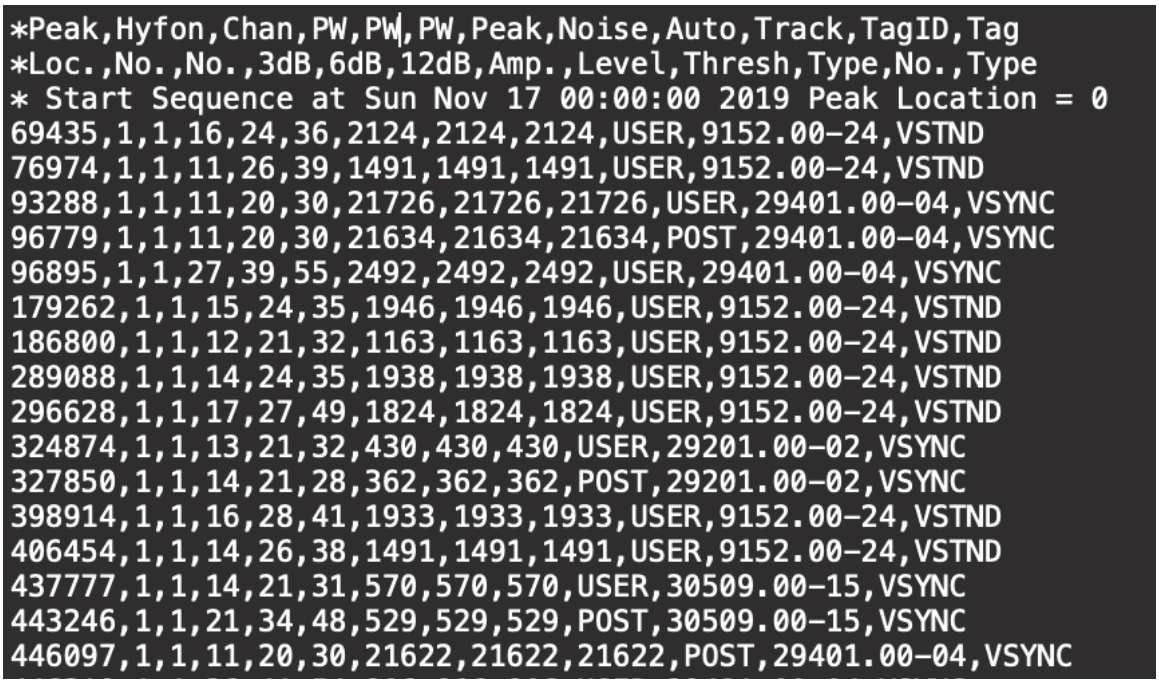
The recorded pings are to be analyzed and identified to know from which tag the ping originated. The analysis of identifying the pings from high-frequency tags with a pulse rate encoding scheme is currently done at Innovasea with the help of a visual analytics system called 'MarkTags.' In the MarkTags, Innovasea developed an image representation in which the received pings are plotted according to their time of arrival(x) and their displacement(y) with respect to a chosen clock rate. In this representation, pings originating from a (stationary) tag with a ping rate that matches the clock rate will describe a horizontal track, allowing them to be identified and "marked" by a human analyst through visual inspection. This is called the marking process. There exists an auto-marking technique in MarkTags in which the marking is done automatically by the software. However, the results obtained by the auto-marking technique are poor, and its performance is less than 50% of the manual performance. The results of the auto-marking technique are discussed in the chapter-5. While most of the marking is automatic, a manual annotator verifies and identifies the pings missed during auto marking. To manually mark the data generated in 1 day, it takes approximately 24 hours to identify the pings from 40 tags in 10 hydrophones. Though manual marking is accurate, it consumes much manual effort to do the task. The main motivation behind the implementation of machine learning in marking the data is to reduce human involvement and improve the poor performance of the existing auto-marking solution.

The manual marking process of identifying the tags is replaced by a machine learning approach to auto-mark the tag data in this project. In this approach, the pings are converted to images with the help of 2D histograms. The pings are represented as pixels in these images. These images are segmented using a machine learning model UNet[11] in which a label is assigned to each pixel. From these segmented images, the pixels are inversely mapped to pings. The process of creating images and segmenting them is explained in chapter 4.

1.3 Data and Description

When a tagged fish moves in water, its position is captured by the hydrophones or receivers within the range, and the data is extracted from the hydrophones. The information captured in the receivers contains the time at which the ping occurs and

the strength of the signal from fish tags and other information discussed below. The time information is used to retrieve which tag is responsible for the ping. This data is not typical of acoustic telemetry datasets. The data we use has two types of files. One is the Raw data files, and the other is Marked data files. The raw data files have the extension *.rat, and the marked data files have the extension *.tat. The raw data files and the marked data files have almost the same information, but the marked data files have some extra information about the tag and marking type. The information about various features in the dataset is given below.



```
*Peak,Hyfon,Chan,PW,PW,PW,Peak,Noise,Auto,Track,TagID,Tag
*Loc.,No.,No.,3dB,6dB,12dB,Amp.,Level,Thresh,Type,No.,Type
* Start Sequence at Sun Nov 17 00:00:00 2019 Peak Location = 0
69435,1,1,16,24,36,2124,2124,2124,USER,9152.00-24,VSTND
76974,1,1,11,26,39,1491,1491,1491,USER,9152.00-24,VSTND
93288,1,1,11,20,30,21726,21726,21726,USER,29401.00-04,VSYNC
96779,1,1,11,20,30,21634,21634,21634,POST,29401.00-04,VSYNC
96895,1,1,27,39,55,2492,2492,2492,USER,29401.00-04,VSYNC
179262,1,1,15,24,35,1946,1946,1946,USER,9152.00-24,VSTND
186800,1,1,12,21,32,1163,1163,1163,USER,9152.00-24,VSTND
289088,1,1,14,24,35,1938,1938,1938,USER,9152.00-24,VSTND
296628,1,1,17,27,49,1824,1824,1824,USER,9152.00-24,VSTND
324874,1,1,13,21,32,430,430,430,USER,29201.00-02,VSYNC
327850,1,1,14,21,28,362,362,362,POST,29201.00-02,VSYNC
398914,1,1,16,28,41,1933,1933,1933,USER,9152.00-24,VSTND
406454,1,1,14,26,38,1491,1491,1491,USER,9152.00-24,VSTND
437777,1,1,14,21,31,570,570,570,USER,30509.00-15,VSYNC
443246,1,1,21,34,48,529,529,529,POST,30509.00-15,VSYNC
446097,1,1,11,20,30,21622,21622,21622,POST,29401.00-04,VSYNC
```

Figure 1.1: A screenshot displaying the header and records information from a TAT file

- Peak Loc.: The primary element of the dataset to consider is the 'Peak Locations.' 'Peak Locations' are the timestamps at which the pings are received from the fish tags. In other words, peak locations are integers that represent the time at which ping occurred. The pings are signals sent from acoustic tags surgically implanted in fish. Each ping represents a signal of the clock frequency of 12000 kHz. A peak location divided by 12000 gives the number of seconds after which the ping occurred after the receiver's auto-reset. The receiver auto-resets every day at a specific time to start the clock from '0' every day implying the peak

locations to start from '0' after the receiver is reset.

- Hyfon No.: The hydrophone number or the receiver number.
- Chan No.: The channel number of the hydrophone.
- PW 3 dB, 6 dB, 12 dB: The next three columns describes the pulse width at -3 dB, -6 dB and -12 dB.
- Noise Level: Average Noise Level in samples over 1 sec. before receiving the signal.
- Auto Thresh: Average of an adaptive threshold in samples over one second before receiving the signal.
- Track Type: Type of marking process. 'POST' indicates the marking done using MarkTags software. 'USER' indicates the manual marking done. 'GPS_SYNC' indicates the signals generated by the receiver to synchronize with UTC.
- TagID No. (Period No.): The period of the fish tag. The format of the period number is PPPP.PP-SS, where P indicates period and S indicates subcode.
- Tag Type: Indicates the type of tag. VSYNC indicates the time-synchronous tag, and VSTND indicates the fish tag.

The information contained in the RAT files is the same from headers' Peak Location' to 'Auto Threshold.' The other two columns,' Track Type' and 'Period No.', are added in TAT files after the data is marked.

1.4 Questions and Contribution

There have been several advancements in fish tracking over the years. These fish tracking methods include tracking using video data, image data and acoustic data, etc. On the other hand, most of the fish tracking methods which involved machine learning were only done using video data or image data. This thesis proposes a novel idea of using an image segmentation technique in tracking the fish with acoustic time-series data generated from high-frequency tags, which use the pulse rate encoding scheme.

This work tries to answer the following questions:

1. How are the pings from high-frequency tags viewed as an image?
2. How are the images classified for the presence of a tag?
3. What are the augmentation techniques implemented in this thesis?
4. How is the image segmentation technique useful in identifying the pings from tags?
5. What is the performance of our proposed machine learning method to auto-mark data compared to the manual marking and MarkTags' auto-marking process?
6. How can the performance of our proposed method improve?

The research summary and contributions of this work can be listed as follows:

- We propose a new method of auto marking the fish tracking data obtained from high-frequency acoustic tags by building a visual representation of the acoustic time-series data and training a deep learning model at identifying patterns of interest.
- We demonstrate the use of two neural networks, one to check if the given tag is present in a given image and the other neural network to find which pixels represent the identified tags.
- The models are made robust and made to work on data recorded in different conditions like different water bodies and different background noise that the model has not seen during training.

The rest of this thesis is organized as follows: Chapter 2 discusses the previous related work done. Chapter 3 includes the preliminaries to understand this document. Chapter 4 and Chapter 5 explain the proposed method, experimentation, and results. The conclusion and future work are included in Chapter 6.

Chapter 2

Literature Review

Fish tracking involves identifying and tagging the fish to gather data on survival, reproduction, activity, behavior, and physiology [12]. Many researchers have studied fish using different tagging and marking techniques. This chapter briefly discusses some of the related research works and technical advancements that happened over the years.

2.1 Fish Tracking Methods

For marine life researchers, the concept of tracking and monitoring fish is common. Fish tracking has many uses, for example, it informs fisheries operations and marine conservation efforts. Fishery scientists need to understand the movement patterns at a broader level. Acoustic telemetry can provide useful insights on fish activity and survival, given that each tag has its signal, and the receivers capture its frequency, time of arrival, and strength. Over the years, there have been several developments in tracking fish using different approaches that involve different electronic tags like radio tags and acoustic tags.

2.1.1 Tracking Using Electronic Tags

An electronic tag is attached to a fish, and the information like position, movements and physiological parameters can be recorded wirelessly using a mobile receiver or stationary loggers [13]. These receivers can be of different types like Data Storage Tags (DST), and Pop-up Satellite Archival Tags (PSAT)[13]. DSTs record and store information of environmental and/or physiological parameters in the tag, and hence these are needed to retrieve the information.The PSAT captures data on fish environments such as light, pressure, and water temperature. After a predefined period, the tag detaches from fish and floats on the surface[13]. The stored data is sent to satellites when the tags loosen from the fish and pop up to the surface.

The recent advancements in electronic tracking technology have provided several tools to study animal behavior in water and land environments, which was not possible decades ago. The use of electronic tags has proven to be a powerful and effective technology for studying movements, migrations, and habitat use of individual free-swimming fish and other aquatic animals in freshwater, estuaries, near coastal areas, and the oceans [14]. These methods can be used to monitor fish behaviour in any aquaculture environment, ponds, tanks, etc. Electronic tags provide repeated information from the same individuals, and most methods do not require the fish to be recaptured to retrieve information. An electronic tag is attached to a fish, and its information on the position, measurements of environmental and physiological parameters are transmitted wirelessly from the tag to a receiver. However, some tags record and store the information and need to be retrieved for downloading data[13].

2.1.2 Radio Telemetry

In this type of tracking, a radio transmitter is attached to a fish, and this tag emits radio signals to a radio receiver placed at a distance in the range of several tens of meters to a few kilometers away. Radio signals propagate omnidirectionally and in both water and air. Hence, the receivers can be placed either in the water or in the air. The operational frequency for the radio telemetry is in the range of 30-300 MHz. Highly suitable for depths less than 10 meters.

Each radio transmitter transmits signals with a unique combination of frequency and pulse rate or coded signals [15]. A digitally coded signal is made up of a unique sequence of pulses in time that the receiver recognizes. It is more efficient to utilize coded transmitters to track a large number of fish than to use various frequency transmitters.[15].

2.1.3 Acoustic Telemetry

In acoustic telemetry, acoustic transmitters are used. Acoustic transmitters are similar to radio transmitters attached to fish and transmit signals to a receiver at a distance of several meters to few kilometers [16]. Unlike radio transmitters' radio signals, acoustic transmitters transmit pressure waves that propagate omnidirectionally only through the water and not air. Hence, the receivers are placed only in water.

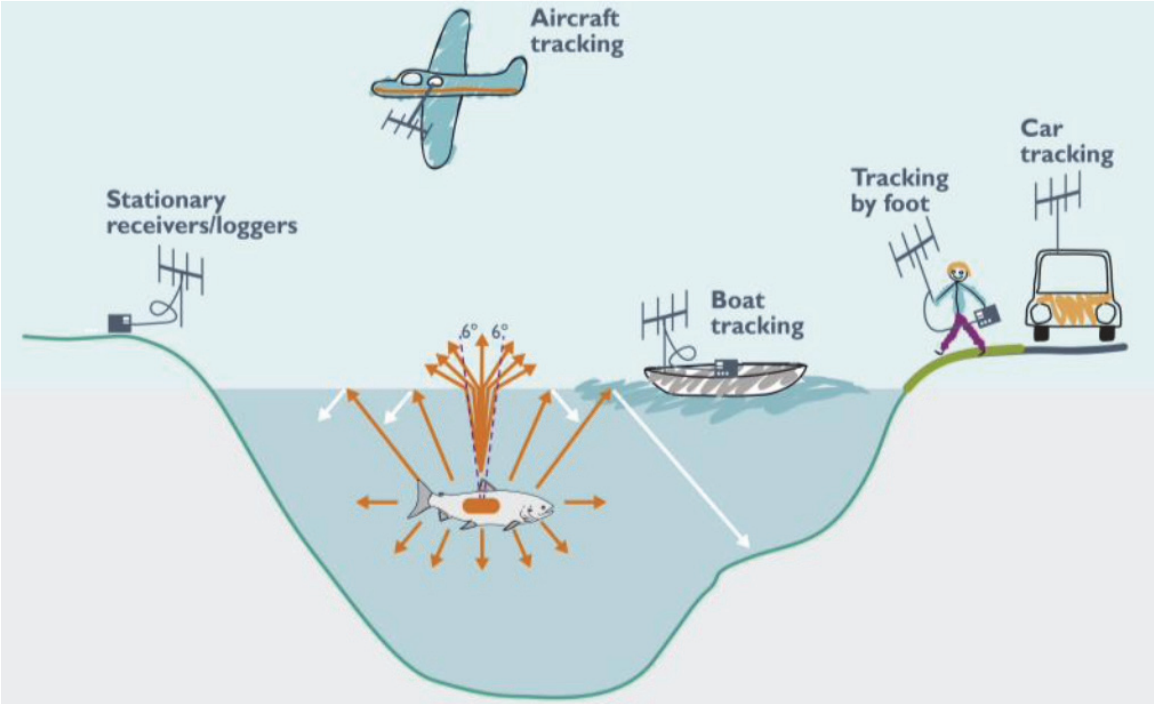


Figure 2.1: Radio telemetry illustrating the use of radio transmitters and the fish can be tracked using boat, aircraft, car, foot, and stationary receivers on the shore. Image taken from [13]

The operational frequency of transmitters is in the range of 30-300 kHz [6]. The transmitters are usually transducers that convert electrical energy into acoustic energy detected by an underwater receiver. These receivers can be stationary receivers submerged in water or receivers mounted to a boat. These receivers are termed hydrophones. This method is highly effective in depths of more than 20 meters as the depths less than 20 meters can lead to many multi-path pings, and it will be challenging to identify the actual signals. Though the data analysis is challenging in shallow environments in acoustic telemetry, it is still the preferred method because of the good detection of signals and higher resolution of position reconstruction than the radio telemetry.

Using multiple hydrophones can provide precise two or three-dimensional tracks of animals. However, post-processing the data from multiple hydrophones is challenging and requires high analytical efforts. These hydrophones can be susceptible to noise from boats or sounds from other sources.

The capabilities of acoustic tags and receivers have improved with microelectronics

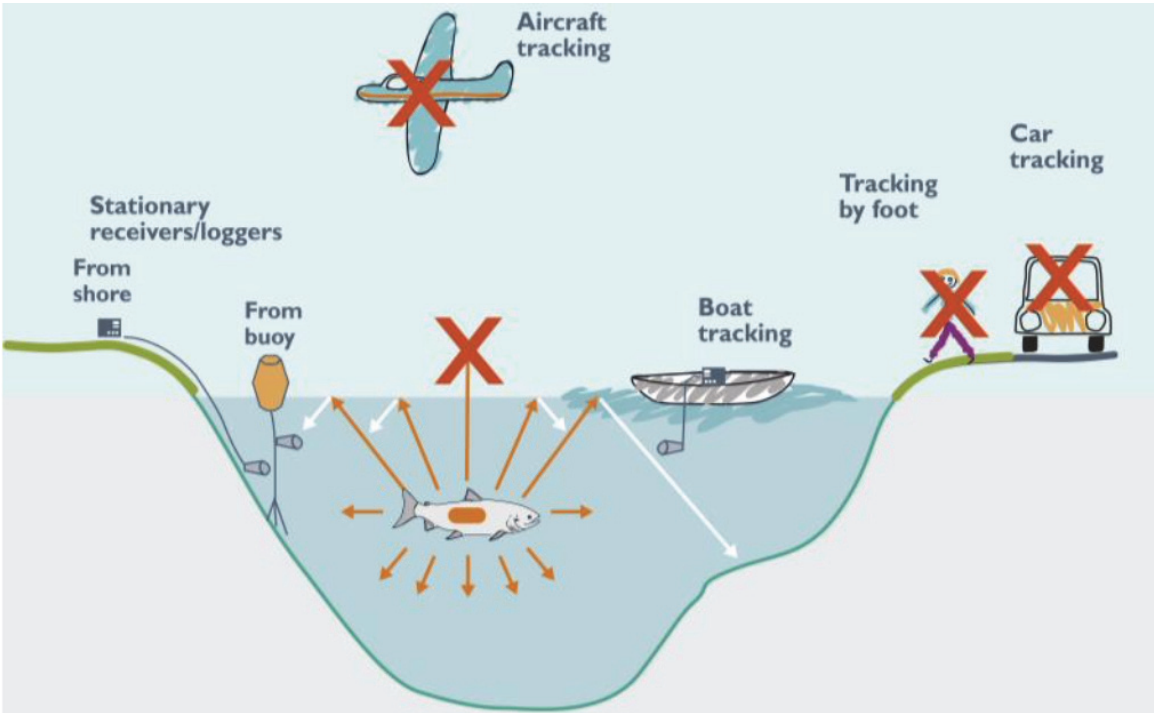


Figure 2.2: Acoustic telemetry illustrating the use of acoustic transmitters and the fish can be tracked by hydrophones or receivers mounted to boats submerged underwater. Image taken from [13]

advancements [17]. The miniaturization of acoustic transmitters was a significant development in telemetry technologies. As tags become smaller, we can learn juvenile fish's behavior and examine evolution changes in behavior [6].

Telemetry allows marine researchers to obtain high-resolution information on the movement and behavior of an individual fish. It also provides solutions to fishery science problems that have been difficult to solve using other methods, including the estimation of natural mortality and exploitation rates [6].

Although acoustic and radio telemetry provides several advantages, they have some limitations too. Telemetry experiments usually generate a large amount of data, making storage and manipulation difficult [18]. The transmitters and receivers used in the telemetry studies are expensive, and these constraints lead to smaller-sized tags. Furthermore, another type of constraint includes the transmitters of one company that may not be compatible with the receivers made by another company, limiting the collection of data from tagged fish [19].

2.2 Research Works on Acoustic Telemetry

The research works explained in this section are not directly related to our work, but these works give an idea of the areas where tagging and tracking of fish are done.

Several research works have been conducted on acoustic telemetry in recent years. Leander J et al. [20] discussed two different acoustic telemetry systems, a commonly used analog pulse-position-modulation-based system (VEMCO PPM) and a newly developed high-residency digital binary phase shift key-based system (VEMCO HR2)(BPSK) to track downstream migrating Atlantic salmon and European eel around hydropower facilities. The coding scheme used in our project is the HTI coding scheme. We shall compare the HTI coding scheme with PPM and BPSK in section 2.3.

Krause et al. [21] estimated the survival rates of weakfish Cynoscion regalis, an economically important species using acoustic telemetry. As a part of the experiment, telemetered fish were released into five estuaries from New Jersey to North Carolina. They released around 342 tagged fish in total during 2006-2012, with the estuary's sample size differing. Telemetry-tagged fish emigrated from estuaries and did not return in subsequent years, indicating low survival. Telemetry also provided information about the mortality timing between the emigration and spring spawning food during over-winter periods. In another similar research conducted by Block et al. [22], the natural mortality of Atlantic bluefin tuna fish is estimated. Bluefin tuna fish tagged in the Gulf of St.Lawrence exhibited high detection rates post-release. 91% were detected one year post-release, 61% were detected 2 years after release. A Bayesian mark-recapture model was applied to the detection data to estimate the rate of instantaneous annual mortality rate. Their results demonstrate that the acoustic tags can provide the life history estimates that are important for developing stock assessment models.

In another research, Førea et al. [23] conducted experiments to study the effect of delousing and crowding on Atlantic salmon using acoustic telemetry. They administered 21 fish equipped with novel transmitter tag type that use pressure and accelerometer sensors to compute swimming activity and swimming depth. In the study conducted over four months, the tagged fish were subjected to three thermal delousing effects. The swimming activity recorded was highly significant during the

delousing effects as compared to low activity before and after the effect. The conclusion was that the ability to collect information from fish responses using acoustic telemetry during operations may be important for developing fish farming methods.

Klinard et al. [24] conducted research to identify the predators of stocked fish in Lake Ontario using acoustic telemetry predation tags with the help of machine learning. Random Forest[25], a supervised machine learning algorithm, was used in their analysis. 48 bloater fishes with predation tags were used to track them on an array of 105 acoustic receivers to quantify the post-stocking survival and predation of prey fish in Lake Ontario. A total of 25 bloater fishes were consumed by predators after release. Post-predation detections provided sufficient information to classify movement patterns. Tagged fish Salvelinus namaycush provided the most reliable classification from behavioural predictor variables with an 89% success rate and was identified as the main consumer of bloater fish. Their research showed that supervised learning methods like random forest would provide information about the fate of stocked fish and predator-prey interactions, and it can be used to guide future stocking and management efforts.

2.3 Coding Schemes in Acoustic Telemetry

We shall review the general coding schemes like PPM, BPSK used in acoustic telemetry and compare them with the HTI coding scheme used in our project.

2.3.1 Pulse Position Modulation(PPM)

In PPM, the information is a series of pulses that represent a unique ID. This unique ID is encoded in the time intervals between successive pulses (i.e., the relative positions of pulses in a series of pulses). Each ID transmitting pulse train takes between 3 to 5 seconds (depending on the tag code space and the ID of the tag), and all pulses must be heard in order for detection to occur. The pulse train is accompanied by a pause to allow other tags to transmit. The configurations of the tag decides whether the delay is fixed or pseudorandom. PPM tag delays are usually in the 30 seconds to 3-minute range. There is a high chance of signal collision since the delay is high [20] if multiple tags are present in the vicinity of the receiver. As a result of the signal

collisions, receivers within detecting range hear unfamiliar patterns that they cannot decode. PPM is useful as it provides a higher detection range than BPSK.

2.3.2 Binary Phase Shift Key

In BPSK, the information is encoded into the signal by modulating the tone wave phase being transmitted. Transmission using BPSK is done in a fraction of seconds compared to the few seconds taken in PPM. Hence there is less chance of signal collision [20]. BPSK is power efficient algorithm than PPM as it uses less power to transmit the signals.

2.3.3 HTI coding scheme

Innovasea uses an HTI coding scheme, which is also called pulse interval coding scheme, on some of the high-frequency tags. It is a scheme in which the information is encoded in the timing between two consecutive pulses[26]. This time between the pulses is controlled by the microcontroller present in the tags and is called tag pulse rate or period that uniquely identifies the specific tag. This coding scheme is explained in detail in the preliminaries chapter of this document. Compared to PPM and BPSK, the HTI encoding scheme is useful in noisy environments and the advantage of this scheme is that all the energy in the transmitted signal is used for detection as well as to identify the tag.

PPM-based equipment is typically used for tracking larger species, over larger geographic areas such as oceans, lakes and rivers whereas BPSK and HTI based equipment are more typically used in smaller species, and most often in rivers and around hydro-electric facilities(fish passage applications). A PPM tag typically takes a few seconds to transmit an ID and optional data, whereas BPSK and HTI tags are on the order of milliseconds. Because BPSK and HTI transmissions are so much shorter than PPM, BPSK and HTI systems can support much larger numbers of tags in the water as PPM. PPM is currently much more common than the others due to its simplicity and there is no need for the type of post-processing required by the HTI coding scheme.

2.4 Related Image Segmentation Works

While convolutional neural networks (CNN) have existed for a long time, their applications were limited to classification tasks. In general, in the classification task, a single object is present in the image, requiring the CNN to identify the object in the image. However, in real-world scenarios, we see images with multiple overlapping objects that require us to classify the objects in the images and identify the boundary of the objects. This is called image segmentation. For performing image segmentation, various neural networks like R-CNN(Regional CNN)[27], Fast R-CNN[28], Faster R-CNN[29], Mask R-CNN[30], etc., were developed.

The dataset we have worked on is an acoustic time series data in which we create a visual representation of the pings. These visual representations or images are trained on a machine learning model to segment the pings in the image. To segment the images, we use a neural network called UNet[11]. The procedure to create a visual representation of pings and train them on a machine learning model is discussed in the later chapters. We now briefly discuss some of the research papers in which machine learning models are used on segmentation data similar to our project.

Unlike our project, which uses the segmentation of images from acoustic time series data, Burguera et al. [31] worked on the segmentation of Side-scan Sonar acoustic images. They proposed a method to perform on-line multi-class segmentation of side-scan sonar acoustic images to build a semantic map of the sea bottom. They used Convolutional Neural Network that follows encoder-decoder architecture, a similar Neural Network UNet used by us except that our Neural Network has skip-connections joining layers. The output images here are called ground truth images, in which each bin in the image is labeled as a specific class.

In research conducted by Yegireddi S. et al. [32], images generated from the Subbottom acoustic profiler were segmented. The images are the acoustic images of the upper sediment layer of the seabed. Here segmentation is necessary to delineate the subbottom structure from noisy acoustic data, which is not possible from conventional image processing techniques. For this purpose, an SOM (Self Organizing Maps) unsupervised neural network was used, whereas the UNet used in our project is a supervised neural network. SOM has the capability to classify the data into different clusters.

Ilin S et al. [33] used a similar segmentation technique for segmenting the biomedical acoustic images. This technique efficiently classifies a group of similar pixels and separates them into particular characteristic regions. The classification is carried out by learning vector quantization neural networks, which separate the image's primary classes.

Chapter 3

Preliminaries

In this thesis, we have used different terms and methods which readers may not be aware of. This chapter explains the terms and methods used in detail.

3.1 Fish Tag

A fish tag[10] is an electronic device that is surgically implanted into a fish. This device may contain different sensors like pressure sensors, accelerometer sensors, temperature sensors, gyroscope sensors, and other sensors[7]. These tags transmit information in the form of pulses. These tags are either surgically implanted or externally attached to a fish of interest so that once the tagged fish is released in water, it can be tracked by a receiver within a range. This range varies from a few meters to more than a kilometer[6].

3.2 HTI Acoustic Tag

Innovasea uses the high-frequency tags called HTI acoustic tags[26] besides using other tags for fish tracking. The HTI acoustic tag operates at a high frequency of 307 kHz. As compared to other types of acoustic tags, these HTI tags use "pulse-rate encoding," which increases detection range, enhances signal-to-noise ratio and pulse-arrival resolution, and decreases position variability [17]. The interval between each transmission is used through pulse-rate encoding to detect and identify the tag (Figure 3.1). To detect and monitor individual tagged fish's actions moving inside the array of receivers, each tag is programmed with a special pulse-rate encoding.

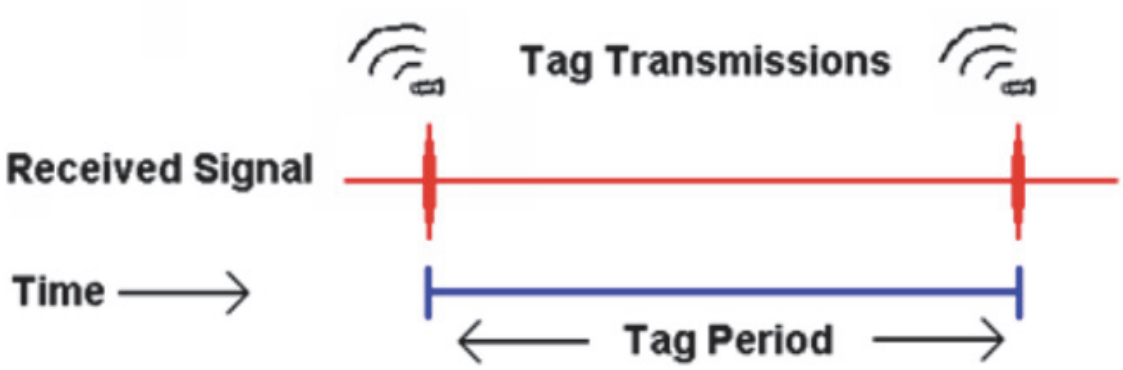


Figure 3.1: Pulse-rate interval, also referred to as the "tag period" or "ping" rate, describes the amount of elapsed time between each primary tag transmission. Image taken from [26]

The pulse rate is measured by calculating the distance between the leading edges of two consecutive pulses. By using slightly different pulse rates, tags can be identified uniquely. The timing of the start of each tag can be controlled by a microprocessor embedded within the tag. This tag can be programmed to have its own tag period to identify each tag uniquely.

The HTI tag double-pulse mode or "subcode" option can be used in addition to the tag time to increase the number of unique tag ID codes available. Each tag is programmed with a defined primary tag time and a defined secondary transmit signal, known as the subcode, using this tag coding option. This subcode specifies the exact amount of time between the transmissions of the primary and secondary signals (Figure 3.2).

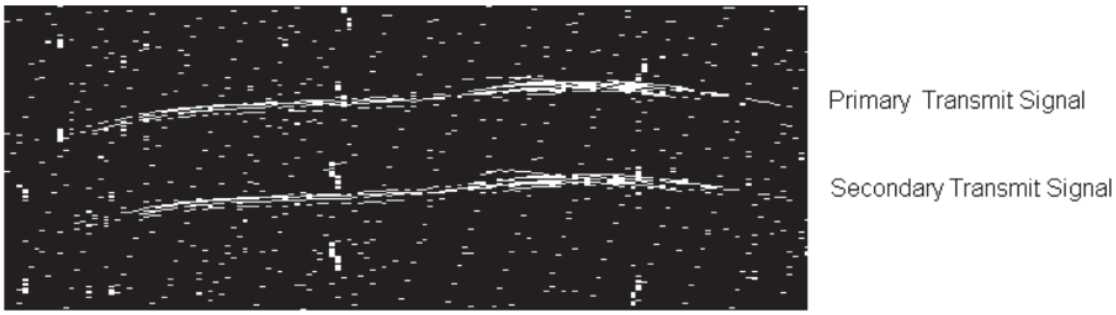


Figure 3.2: Example graph showing the primary (tag period) and secondary (subcode) transmit signal illustrating double pulse scheme. Please refer to section 1.2 for detailed information. Image taken from [26]

3.3 Multipath Signals

The fish tags emit information omnidirectionally. Multipath data are signals received from the same tag but different sources due to reflections from the water surface or surrounding structures[17]. Multipath Signals always arrive at the hydrophone after the direct-path transmission and are usually (but not always) weaker in signal strength compared to direct-path signals. Since these pings appear at different pixels in the image, it provides an ambiguous situation to the annotator where an educated guess is made to mark one of the multipath pings.

3.4 Doppler Effect on Received Pings

The Doppler effect[34] is a change in the observed sound pitch that results from relative motion. Whenever a fish moves away or towards the receiver, the pings emitted by the tag reach at a different time than the exact time period due to the doppler shift. When moving towards the receiver, the pings will have less distance to travel, so signals arrive faster. And, when moving away from the receiver, the pings have a greater distance to travel, and so signals arrive with a longer period. For example, a fish moving toward and passing a receiver generates a roughly semi-circular pattern of pings in an image when observed at a coarse resolution due to the doppler effect. Experts manually mark such patterns as a signal using the visual analytics system. This is challenging for machine learning as the neural network cannot distinguish between the slant tracks generated due to the doppler effect and the slant tracks due to the nearby tag period. This effect leads us to implement an augmentation technique(explained in section 4.4.2) to solve the problem of the machine learning approach.

Now, we shall go through the machine learning models implemented in this project.

3.5 Deep Learning

Deep learning[35] is a subset of machine learning mainly concerned with algorithms inspired by the human brain's structure and functioning called artificial neural networks, which are deep and have many nodes. An Artificial Neural network is a collection of nodes called artificial neurons. These neurons have connections that can be activated and transmit values to other neurons based on the received values. Each neuron performs a linear operation on the input data, which is essentially matrix multiplication, and the coefficients of this linear operation are called weights and biases. Weights are learned during the training process for the connections between these neurons. Typically, neurons are aggregated into layers. Different layers perform different operations on their inputs. Deep Neural networks have more layers, typically more than three layers, compared to shallow artificial neural networks, which have less than three layers to learn more features from the data.

Backpropagation[36] is employed in the learning process. Backpropagation is a gradient descent technique for choosing the optimum weight values depending on input label. Nodes with greater error rates are given lower weights, and vice versa, in order to calculate how much of the overall loss is attributable to them. The weights are then updated in such a way that the total loss is minimized. For further reading, a more comprehensive introduction to deep learning may be found in [37]

3.5.1 Overview of Neural Networks used in this work

In this work, we have used two different neural networks, the Xception neural network[38] and the UNet neural network[11]. Xception neural network is used to identify the presence of a tag in an image. It classifies the image as either '1' or '0' based on the presence of a tag. And, the UNet neural network is used identify the pings occurred from the tag in an image. This is done by image segmentation. A more detailed information of these neural networks is given in the following subsections.

3.5.2 Convolution

A convolution[39] is the application of a filter(set of weights) to an input, resulting in activation. Activation is the result of a linear operation that involves the multiplication of a set of weights with the input. When the same filter is applied to the input repeatedly, it produces a map of activations called a feature map, displaying the locations and intensities of a detected feature in the input, such as an image. These are the basic building blocks used in a neural network.

3.5.3 Convolutional Neural Network (CNN)

Convolutional Neural Networks, which are also called ConvNets, are composed of multiple layers of artificial neurons[39] in which the convolutional layers are the basic building blocks. In CNN, the layers are organized in 3 dimensions width, height, and depth. The input is an array with shape (number of inputs) x (input height) x (input width) x (input channels). Here, each input is an image, and the input channels can be RGB channels of an image. After passing through a convolutional layer, the features of the image get extracted within a convolution layer. The image becomes abstracted to a feature map, also called an activation map, with shape: (number of inputs) x (feature map height) x (feature map width) x (feature map channels)[39] . Hyperparameters of a neural network are the variables that determine the network structure and variables which determine how the network is trained. A convolutional layer within CNN generally has the following attributes:

- Convolutional filters/kernels defined by a width and height (hyper-parameters).
- The number of input channels and output channels (hyper-parameters).
- Additional hyperparameters that control the output volume of the convolution layer, such as padding, stride, and dilation.

A more detailed explanation of the CNN can be found at [40]

3.5.4 Xception Neural Network

We generate the images using the pings from acoustic tags(the method to create images is discussed in chapter 4) and assign labels as either 0 or 1, indicating the

presence or absence of a chosen tag in a specified interval. To label the images either positive or negative, we check the image for the number of pings from the tag, and if the number of pings is greater than the predefined tunable threshold, the image is labeled as a positive label or '1' or else if the number of pings is less than the predefined threshold, the image is assigned a negative label or '0'. We train a neural network with the images and the binary labels to classify the image for the presence of a tag. The neural network we used for this classification task is an Xception neural network. We shall discuss the architecture of the Xception neural network[38] briefly.

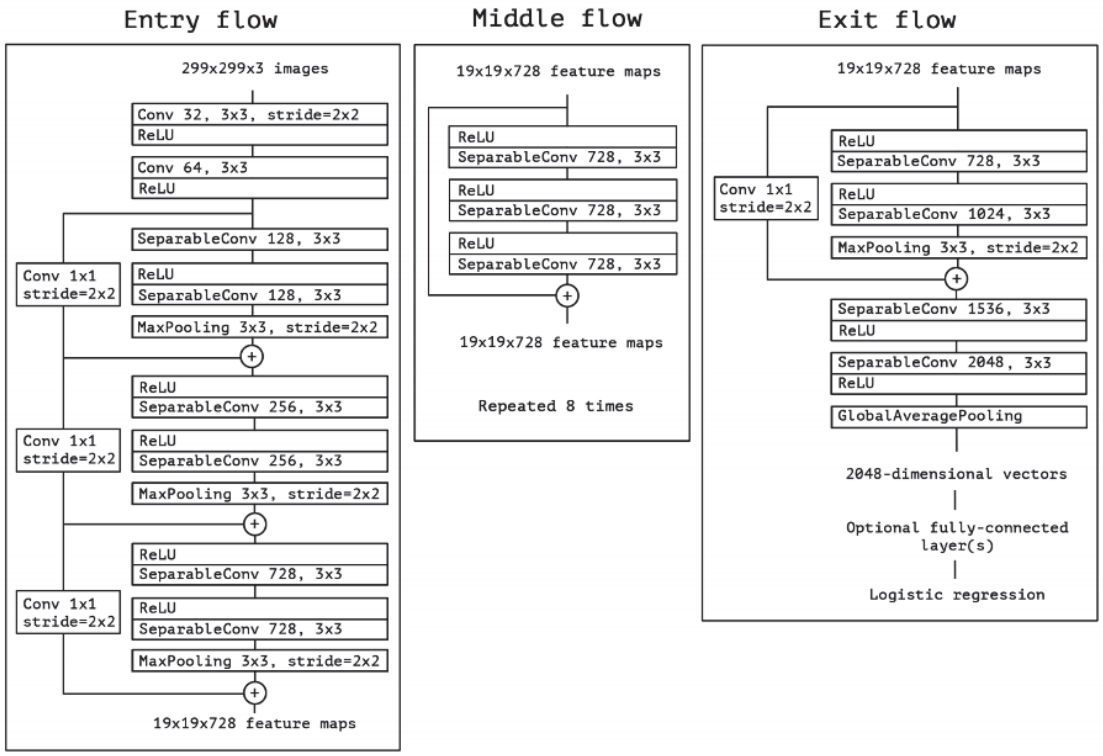


Figure 3.3: The Xception architecture in which the data goes first through entry flow, then through middle flow which is repeated 8 times, and then through exit flow. Image taken from [38]

Xception neural network is an Extreme version of Inception neural network. Xception neural network relies mainly on two parts. One is depth-wise separable convolution, and the other is the shortcuts between the convolution blocks. Xception offers architecture that is made of depth-wise separable convolution + Max Pooling, all linked with shortcut connections as shown in figure 3.3. All the convolution and separable convolution layers are followed by batch normalization. These are explained

briefly in the next subsection.

3.5.5 Depth Wise Separable Convolution

Depth wise separable convolutions are alternatives to regular convolutions that are much more efficient in computation time [38]. Depthwise convolution is the process of applying convolution of size d x d x 1 instead of the convolution of size d x d x C, where C is the number of channels and d x d is the size of the convolution filter. This creates the first volume with size K x K x C and not K x K x N, where K is the resulting dimension after convolution and N is the number of kernels[38] . We only made convolution for one kernel/ filter of the convolution, not for N of them. The next step is pointwise convolution[38] . Pointwise convolution operates classical convolution, with size 1 x 1 x N over the K x K x C volume. This allows creating a volume of shape K x K x N. Therefore, the number of operations is reduced by a factor proportional to 1/N. The Depthwise separable convolutions have become popular as they have fewer parameters than regular convolution layers and are less prone to overfitting. Since they have fewer parameters, they also require fewer operations to compute and thus are cheaper and faster. The main point to note in the Xception neural networks is that the depth-wise separable convolution follows the pointwise convolutions in an Xception neural network.

3.5.6 ResNet-50 Neural Network

ResNet-50[41] is a variant of the ResNet model, which has 48 convolutional layers with 1 Max Pooling layer and 1 Average Pooling layer. The ResNets were initially applied to image recognition tasks, but the framework can also be applied to non computer vision tasks to achieve better accuracy. Deep Neural Networks are known for identifying small, high-level features from the images, and stacking more layers can generally improve accuracy[41] . Besides, they are also susceptible to the problem of vanishing gradients. During backpropagation, the error is calculated, and gradient values are determined. These gradients are sent back to hidden layers, and weights are updated accordingly. These gradients are sent back until it reaches input layer. The vanishing gradient is a gradient that gets smaller and smaller as it approaches the input layer. In other words, the input layer does not learn effectively[41] . Hence,

deep networks will not converge, and accuracy will start to degrade or saturate at a particular value. This problem is handled by introducing shortcut connections between the layers in the deep neural network to perform identity mappings. A more detailed explanation of ResNet-50 can be found at [41]

3.5.7 Semantic Image Segmentation

Image segmentation is used to know the location of an object like a car or a cat in an image. To do this, the image is segmented, which means each pixel is assigned a label, thereby generating a pixel-wise mask of the image. Semantic image segmentation involves training a neural network to generate a pixel-wise mask of the image[42]. In general, a mask is a two-dimensional array in which each pixel has a value in-between '0' and '1'. We also used the term 'binary mask' in our thesis, which means that the value of each element in the two-dimensional array is either a '0' or '1'. This aids in the comprehension of the image at the pixel level. The output here is a high-resolution image in which each pixel is classified into a particular class. It involves training a neural network, which can be trained to generate the pixel-wise mask of the image. The neural network's output will be a high-resolution image whose size is equal to the input image. Medical imaging [43], self-driving vehicles [44], and satellite photography[45] are just a few applications for image segmentation. In medical imaging, the image segmentation technique detects any types of tumors or cancers present in the human body. In self-driving vehicles, semantic segmentation provides information about free space on the roads and detects lane markings and traffic signs. Land cover classification may be thought of as a multi-class semantic segmentation problem in satellite imaging, recognizing the kind of land cover (e.g., urban, agricultural, water, etc.) for each pixel on a satellite picture. In this thesis, we used a neural network called UNet for performing image segmentation of the images to find the pings from the fish tags. As opposed to other alternatives, we chose UNet for image segmentation because the UNet has an extra feature called skip connections joining the output of upsampling layers and the input layers which help in retrieving the precise location of features in a image. As a part of future work, we also have a plan to test other neural networks which are known for image segmentation.

3.5.8 UNet Neural Network

The UNet was developed by Ronneberger for biomedical image segmentation[11]. The main blocks of a UNet are convolution and max-pooling. Convolution and max-pooling[46] are the two most important concepts to consider in order to understand the UNet architecture [11]. Convolution is discussed in section 3.5.2. Pooling's general purpose is to reduce the size of the feature map obtained after convolution, resulting in fewer parameters in the network. The concept behind max pooling is to hold only the essential features (max valued pixels) from each area and discard the irrelevant information thereby spatially reducing the size of the image by 1/4 after max-pooling. Convolution and max-pooling also minimize the size of the images; for example, a 4 x 4 image before pooling will become a 2 x 2 image after pooling. This is known as downsampling. For example, the information present in the 4 x 4 image before pooling, almost the same information will be present in the 2 x 2 image. When we repeat the convolution step, the next layer's filters will see more context; that is, as we go further through the network, the size of the image shrinks, and the receptive field increases. The model can precisely understand the information present in the picture by downsampling[11] but loses the precise location of information present in the image. Upsampling is needed to retrieve the location detail, which requires converting the low-resolution image to a high-resolution image. Transposed convolution[47] is the most common method used for image upsampling. Transposed convolution is the inverse process of regular convolution. After transposed convolution, the output image will be of higher resolution than the input image.

The UNet architecture is split into two parts. The encoding part captures the context of the image, and the decoding part allows precise localization using transposed convolutions. In other words, the model understands what information is present in the image in the encoding part, but it loses where the information is present. This lost location information is retrieved in the decoding phase after transpose convolution. By concatenating the output of the transposed convolution layers with the feature maps from the Encoder at the same stage, we can obtain precise locations of features at each level of the decoder. This results in symmetric architecture in the form of a U, hence the term UNet.

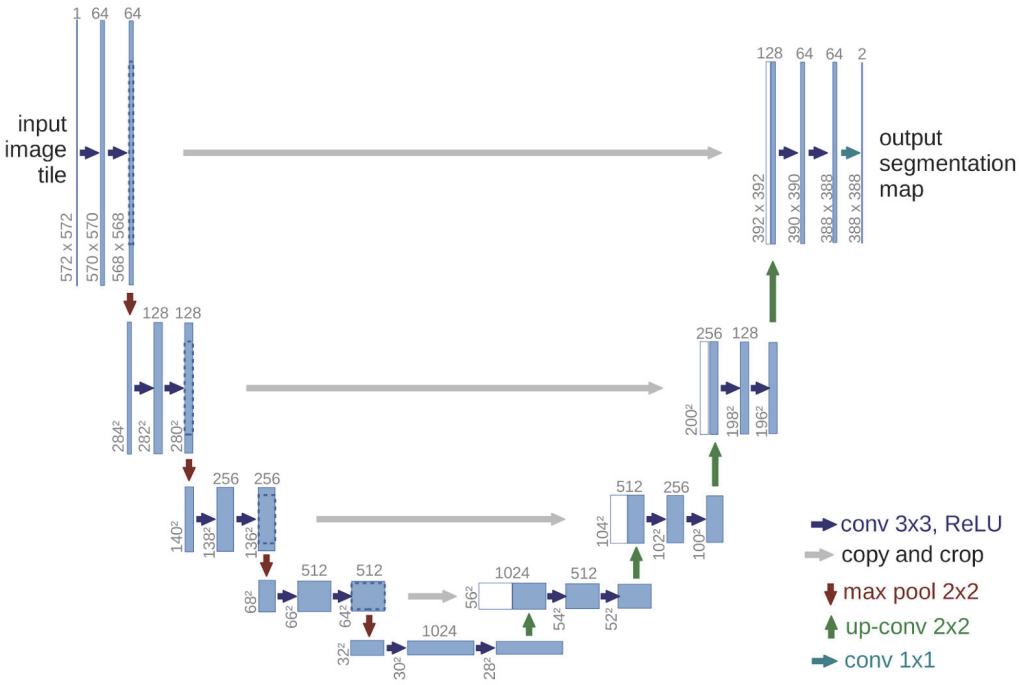


Figure 3.4: U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower-left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations. The image is taken from [11]

3.6 Post-Processing Filtering Algorithms

The application of post-processing filtering algorithms is to get the final list of marked pings from the tag. These algorithms are mainly used to filter out the false negatives identified after the inverse mapping of pixels to pings. The pings which are unmarked after the segmentation can be marked after applying the post-processing filters. These filters can be tuned using few parameters like the threshold and tolerance. There are two types of filtering algorithms implemented by my co-supervisor and co-worker in the project Dr. Oliver Kirsebom. The algorithms are the 'subcode filter' and the 'trajectory filter.'

The subcode identifies the ping pairs with a distance equal to the subcode distance of the chosen tag into account. There are two modes, 'EARLIEST_TIME' and 'LARGEST_AMPLITUDE,' in which the earlier mode marks the pixels with the earliest time arrival, i.e., out of the multiple ping pairs, the pings which arrive first to

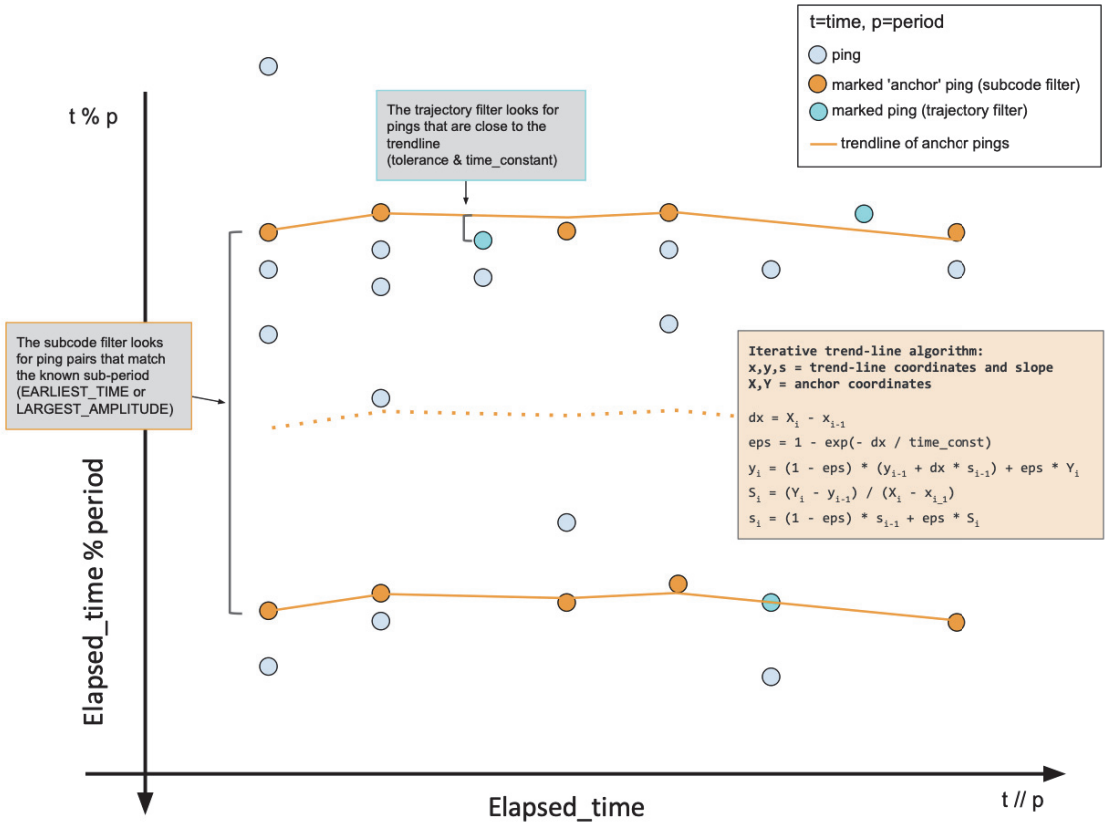


Figure 3.5: Image illustrating the post-processing filtering algorithms. The time elapsed is indicated in the horizontal axis and the displacement with respect to chosen clock rate(represented as time elapsed % chosen tag period) is indicated in vertical axis.

the receiver are marked, whereas the latter mode will mark the pixels that have the largest combinational amplitude, i.e., the pings whose combined sum of peak amplitude is maximum are marked. As shown in figure 3.5, the pings in orange are the pings from the tag, and based on the mode selected, and the ping pair will be marked. Once the ping pairs are identified, a trend line is drawn, joining the pings identified by the subcode filter, as shown in the figure. Based on the tolerance parameter set in the trajectory filter, the pings close to this trend line will be marked as the pings from the tag.

3.7 Test Metrics

The performance of the models is evaluated by comparing the UNet predicted output to the original output both at the pixel level and the ping level. The performance of the models is computed by calculating few metrics described below.

3.7.1 True Positives

These are the correctly predicted positive values which mean that the value of the actual class is yes, and the value of the predicted class is also yes[48]. In our case, the class is either a pixel or a ping associated with the tag.

3.7.2 True Negatives

These are the correctly predicted negative values which mean that the value of the actual class is no and value of the predicted class is also no.[48]

False Positives and False Negatives are the values that occur when the actual class contradicts the predicted class.

3.7.3 False Positives

When the actual class is no, and the predicted class is yes.[48]

3.7.4 False Negatives

When the actual class is yes, but the predicted class is no.[48]

3.7.5 Precision

Precision (P)[49] identifies the total number of correctly predicted samples out of the total predicted samples. It is defined as the number of true positives (T_p) over the number of true positives plus the number of false positives (F_p).

$$Precision = \frac{T_p}{T_p + F_p}$$

3.7.6 Recall

Recall (R)[49] is identified as the total number of predicted samples over the total number of samples present. It is defined as the number of true positives (T_p) over the number of true positives plus the number of false negatives (F_n).

$$Recall = \frac{T_p}{T_p + F_n}$$

3.7.7 Jaccard Index

Jaccard Index (J)[50] gives the total number of correctly predicted samples out of a total number of samples present. It is a similarity measure index that compares the members of two sets to see which members are similar and which are distinct. The higher the Jaccard index, the higher the similarity between the original and predicted samples. This metric can be used to know how similar the predicted, and original samples are.

$$JaccardIndex = \frac{T_p}{T_p + F_n + F_p}$$

3.7.8 F1 score

F1 score[51] is a function of precision and recall. F1 score conveys the balance between precision and recall. When there is an imbalanced class distribution, the F1 score is the best metric to evaluate the model.[52]

$$F1score = \frac{2 * precision * recall}{precision + recall}$$

Chapter 4

Methodologies

Our work requires analyzing the data from fish tags, recorded as pings in the receiver, to know which tag the ping originated from. This chapter explains the step-by-step process in detail, from preprocessing the data to create images, classifying the image for the presence of the tag, and segmenting the pixels from the tag in an image. Our approach contains below main steps:

1. Preprocess data to create images from raw data (RAT files).
2. Create the labels for the images using TAT data.
3. Train and test the Xception neural network with the image data.
4. Train and test the UNet neural network with the image data.
5. Inverse mapping of the pixels to pings from the UNet predicted mask.
6. Apply post-processing filtering algorithms.

4.1 Preprocess and Create Images from Raw Data

We create an image representation with the time of arrival of pings on the horizontal axis and the displacement on the vertical axis to facilitate the identification of tags based on ping rate. The main information required is the peak locations and known tag period to create images from the raw data. The initial step is to read the data from all the RAT and TAT files. Each file has information about the hydrophones, the time the file generated, peak location, channel number, and gain. The description of all these columns is discussed in section 1.3. Reading the data from files involves parsing the RAT and TAT files one at a time, matching the entries in the TAT file with the entries in the RAT file.

The peak locations, which represent the time at which pings reach the receiver, are signals of clock frequency 12000 kHz. Each peak location divided by 12000 gives the number of seconds after which the signal is transmitted after the receiver is reset, which happens every day at a specific time, implying the peak locations to start from '0' after the receiver is reset. We grouped the peak locations into 30-minute intervals to create images. These are grouped such that there is a 20-minute overlap between any two consecutive 30-minute intervals. We used the overlap to make the best use of available data. The peak locations in each of the 30-minute intervals are used to create images. An illustration of how the intervals are split is given in figure 4.1.

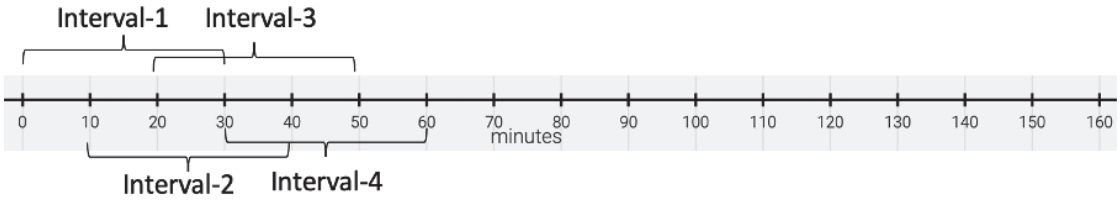


Figure 4.1: Image illustrates how we split the data into 30-minute intervals to create images. The image shows only the first four intervals split. The process continues for the entire dataset.

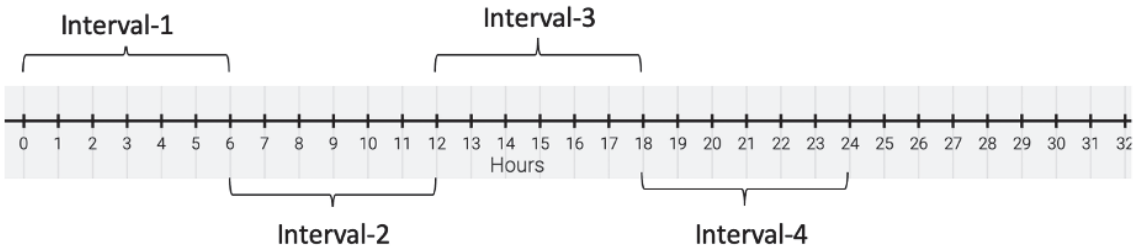


Figure 4.2: Image illustrating how the total data in 30-minute intervals is grouped into 6-hour intervals. The image shows only the first four intervals grouped. The process continues for the entire dataset.

The 30-minute intervals are now grouped such that each interval has data of 6-hours duration, as shown in figure 4.2. This is done to have separate data in each 6-hour interval, i.e., no two 6-hour intervals have pings in common. These 6-hour intervals are randomly chosen and used as training, validation, and test sets such that 60% of the total intervals form the training data, 20% of the total intervals into validation, and the rest 20% to test data as shown in figure 4.3.

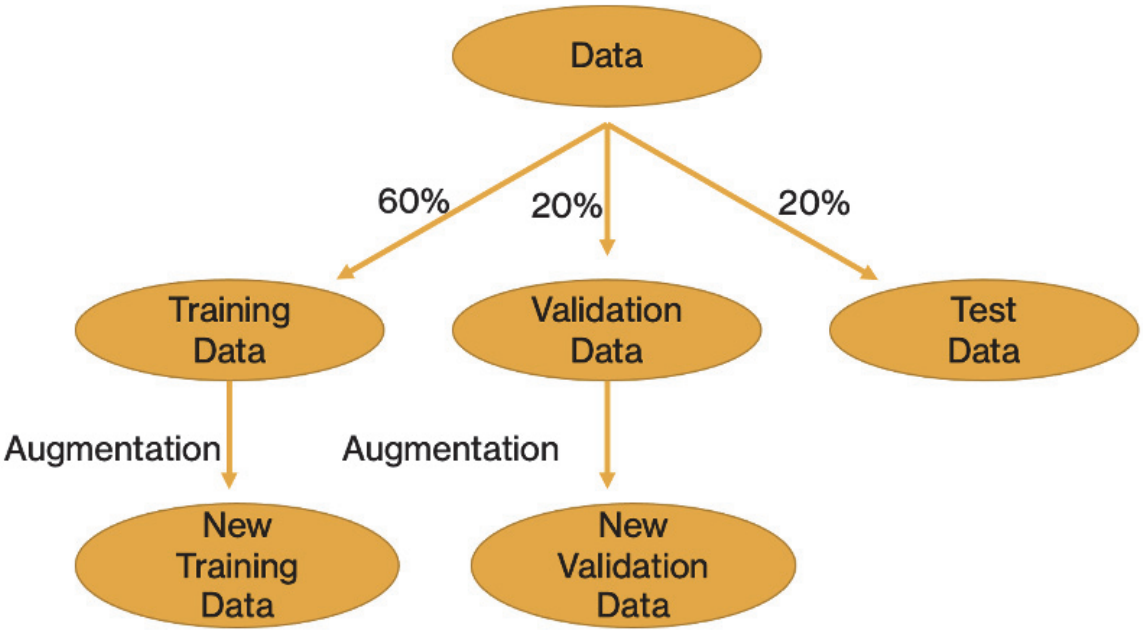


Figure 4.3: Illustration of how the data is split into training, validation and test sets

We convert the data in these intervals to images. For each peak location in the 30-minute intervals, remainders are obtained by dividing each peak location with the chosen tag period. The remainders are the displacements with respect to chosen clock rate. In this representation, pings originating from a stationary tag with a ping rate that matches the clock rate will display a horizontal track. The process of how we represent the pings as an image is given in the below equation.
X =time elapsed (in minutes)
Y = time elapsed % tag period (in seconds)
Histogram = matplotlib.pyplot.hist2d(X, Y, bins)

A 2D histogram is plotted with the array of peak locations(X) and the remainders(Y) by choosing a suitable binning of the histogram. A sample image of a 2D histogram is shown in figure 4.4.

In this 2D histogram, the horizontal axis and the vertical axis both represent a time where the peak locations are shown in the horizontal axes, and the remainders(displacements) are shown in the vertical axes. The third axis represents the number of pings present in each pixel indicated by a colour bar. Time progresses

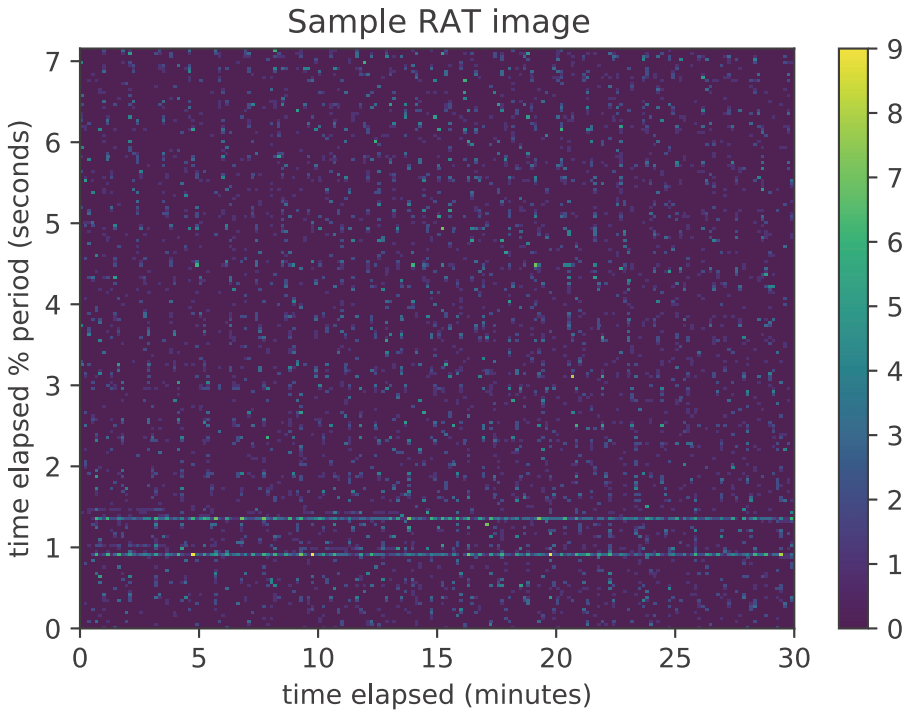


Figure 4.4: Sample 2D histogram image generated from RAT data; colour bar indicates the number of pings present in each pixel

from bottom to top in each 2D histogram. Each side of the histogram is divided into bins, and there are 192 bins on each side of the histogram. The reason for choosing 192 as the bin size is explained in section 4.6. To generalize the image, the range of the horizontal axis is taken as 0 to 30 minutes. The range of the vertical axis is 0 to the tag period (in seconds). Figure 4.4 is an example representation of the image used to train the machine learning model.

Below are the standard parameters we used to generate the images.

- Range of horizontal axis is 0 to 30-minutes.
- Range of vertical axis is 0 to chosen tag period expressed in seconds.
- The dimensions of each image are 192 x 192.

4.2 Generating Binary Labels for Image Classification

The Xception neural network used in this project classifies the image for the presence of a tag. For this, the number of pings in an image from the chosen tag is considered, and the images are assigned positive or negative labels that are either '0' or '1' based on the predefined threshold. If the number of pings in the 30-minute interval image is greater than the predefined threshold(0.2 in our case, the reason for choosing 0.2 is explained in chapter 5), then the image is assigned with a positive label of '1'. Similarly, If the number of pings in the 30-minute interval is less than the threshold, the image is assigned with a negative label of '0'. This array of images and their respective binary labels are used to train the neural network to filter the images for the presence of a tag.

4.3 Creating Binary Mask Images from TAT Data

The binary masks are images created from the marked data files (TAT files). These images are used as labeled output images during the training of the UNet model. In a 30-minute interval, pings from the RAT file are taken, and each ping is assigned either a '1' or '0' based on the condition that the ping has come from the chosen tag(checked in the TAT file), thus creating a weighted array of 0's and 1's. The length of the weighted array is equal to the length of the pings list from a 30-minute interval in the RAT file. This weighted array is used in creating the binary mask. The equation of how the binary mask is plotted is given below.
X =time elapsed (in minutes)
Y = time elapsed % tag period (in seconds)
W=Array of 0's and 1's.(length equal to length of X)
Histogram = matplotlib.pyplot.hist2d(X, Y, bins, weights=W)

The binary TAT mask contains only the pings from the chosen tag, whereas the image generated from the RAT file contains all the pings in the RAT file.

The generalized parameters like horizontal and vertical axes range and the image dimensions for binary TAT masks are the same as the RAT images. The initial classification of images for the presence of tags is tested on two different neural networks.

1. Xception Neural Network[38]

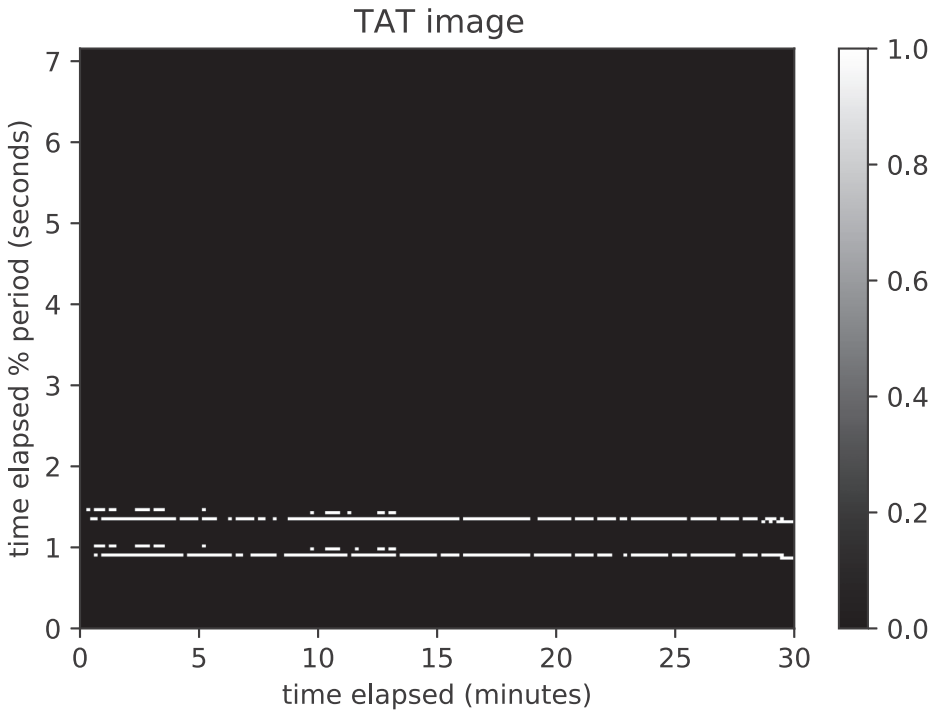


Figure 4.5: Sample binary TAT mask generated from TAT data; colour bar indicates the number of pings present in each pixel

2. ResNet-50 [41]

The two neural network models were trained on a subset of data that was recorded using stationery tags. To train the models, a subset of data recorded from a single tag in a single hydrophone is used. The subset of data is divided into 6024 training samples and 2006 validation samples. The distribution of positive and negative samples for training, validation, and test sets is given in the tables. The relative proportions of the number of positive and negative samples can vary substantially from dataset to dataset. However, this depends on how much time the fish is present within the detection range of hydrophone. We haven't tested our models on datasets with different proportion of positive and negative samples, but we anticipate that the recall and the false positive probability would be similar. The information about the training data is given in table 4.1.

Threshold=10 pings	Positive Samples	Negative samples
Training data	1620	4404
Validation data	395	1611
Test data	343	1663

Table 4.1: Distribution of positive and negative samples among datasets

When the same test data was used to test both the Xception neural network and ResNet-50, the Xception neural network was 99.5% accurate in predicting the labels correctly, whereas the ResNet-50 was 90.7% accurate in predicting the labels. So, we focused our analysis with the Xception neural network.

The classification of an image for the presence of a tag was done on the other datasets recorded in different water bodies. In classifying the images, we encountered few scenarios in which the model failed to predict the labels correctly. These failures showed us the need to augment the data to improve the performance of the model. The augmentation methods and the scenarios at which these augmentation methods were developed are discussed in section 4.4.

4.4 Augmentation Methods

There are two augmentation methods introduced in this project when working on filtering out the images for the tag, i.e. when performing analysis with the Xception Neural network. These augmentation techniques are also implemented in UNet analysis.

The two augmentation methods are:

1. Adding a random offset to shift the horizontal lines vertically.

2. Tweaking the tag period by 10ms to create negatively labeled images.

Let us discuss the two augmentation methods briefly.

4.4.1 Augmentation-1: Adding Random Offset

In the analysis of classifying the images for the presence of the chosen tag period, the model failed to detect some pings in the image in which the horizontal lines were at

the bottom of the image, as shown in figure 4.6. The model also failed in some cases where the lines were at the top of the image and also in the images in which one track was at the top of the image and the other track at the bottom. So, a random offset is added to the remainders so that each pixel in the image shifts upwards in the image. This random offset is constant for all the pixels in a 30-minute interval. An example is shown in figure 4.7, in which the low-lying horizontal lines in the image shift upwards after adding a random offset.

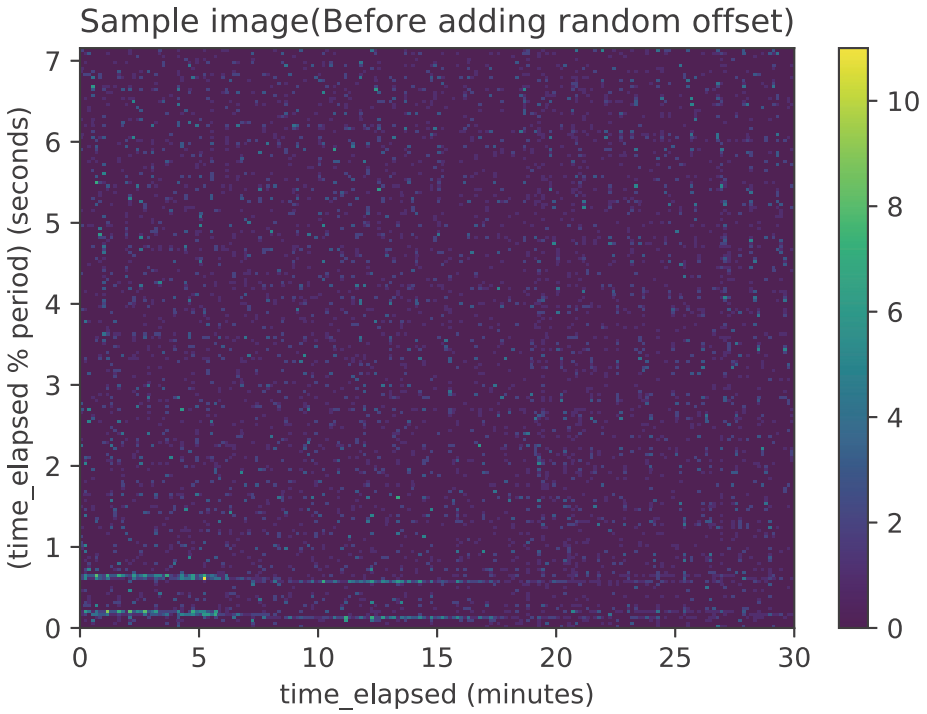


Figure 4.6: Sample image in which the horizontal lines are at the bottom of the image

When the images similar to figure 4.6 are passed through the Xception neural network, the model fails in a few cases to detect some of the pixels in the image. Furthermore, when the images with low-lying horizontal lines are replaced with random offset added images similar to figure 4.7, the neural network was able to detect most of the pixels in the image.

In an analysis done on 782 test images, 14 images were wrongly predicted by the Xception neural network. When these 14 images were replaced with the random offset

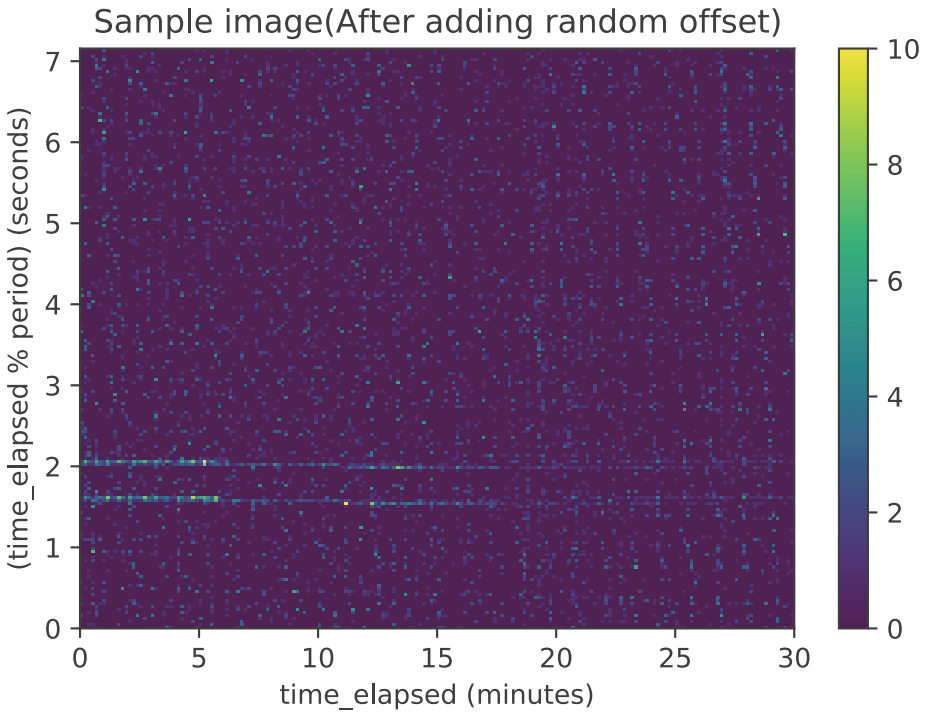


Figure 4.7: Sample image in which the low-lying horizontal lines shift upwards after random offset is added.

added images, the overall performance was improved significantly. The overall accuracy was improved from 95.2% to 96.9%. In addition to accuracy, the precision and recall were also computed before and after replacing the images with a random offset. The precision and recall before adding offset were 100% and 93.7%, respectively. When the misclassified images were replaced with the random offset added images in test data, the precision and recall were increased to 100% and 95.4%, respectively. Since there was an improvement in the performance when the 14 misclassified images were replaced, the training data is replaced with the original images, and the same images with the random offset are added to create the double-sized dataset to train the neural network. When training the model with the updated training data, the model's performance on the test data was further improved, i.e., the model was able to predict the labels with 99.6% accuracy.

When the model which is trained using augmentation method-1 is tested on the different data not used for training and validation, there were several false positives though there were no pings from the test tag. Looking at the false-positive images,

it was found that there were pings from other tags with a minor difference in the tag period causing the model to treat the lines as horizontal lines though the lines had a non-zero slope. The reason for the lines with a non-zero slope is explained in the next section.

4.4.2 Augmentation-2: Tweaking the Tag Period

A fish moving towards or away from the receiver generates pings with delay less than or greater than the actual period, respectively. This effect is also referred to as the doppler effect (explained in section 3.4). Due to the acoustic Doppler effect, the pings obtained from a moving fish appear as tracks with a non-zero slope when viewed as an image. This is because of the difference in the arrival time of pings. The pings which arrive with a minor change in the usual time delay than the actual tag period can also be considered as the pings from the tag.

Now comes the question of an expected possible delay of pings. For example, a fish with a tag period of 3000 ms moving towards the receiver at a speed of 1m/s will appear to have its period reduced by approximately (3000)*(1/1500)=2 ms where the speed of sound in water is 1500m/s. Considering the fish with a tag period of 3000 ms moving at an average speed can have its period reduced by not more than 2-3ms. So, the pings which occur with a delay of + or – 3 ms can be treated as actual tag period. Most of the fish tags used in this project have a tag period in the range of 1000 ms to 10000 ms. So, a cut-off delay of 10 ms is used so that a tag period with t ms may expect pings with delay in the range (t-10) ms and (t+10) ms. This resulted in implementing another augmentation method of adding negative labeled images with minor tag period differences.

This augmentation is used to handle the model's misclassifications in case the tag's period close to the chosen tag period is present in the same image. If there is any tag with a minute difference in the period in the order of tens of milliseconds, the model will not treat the tag as a different tag than the chosen one because of a slight slope in the horizontal lines. Hence, there will be many false positives. Several negatively labeled images are created with tag periods with a difference of 10ms before and after the chosen tag period to handle this problem of false positives. If the chosen tag

period is t milliseconds, the augmented periods will be (t – 10), (t + 10). So, there will be two sets of additional tag period images for each set of tag periods but with negative labels. In other words, out of all the images used for training, only one-third will be original images, and the other two-thirds will be duplicate and negative labeled images.

With the two augmentation techniques implemented, each 30-minute interval will have three sets of images with tag period t, (t-10), and (t+10) milliseconds. Along with these three images, there will be additional three images with a random offset added. So, for each 30-minute interval, there will be six images in total for a single tag. These two augmentation methods were also used in training the UNet model for image segmentation.

Another main point to be noted here is that these augmentation methods are used only for training and validation data. The augmentation is not applied to the test data.

4.5 CNN for Image Classification

The idea of using the Xception neural network to classify the image for the presence of tag is to check if the deep neural networks are compatible with the images generated from acoustic time series data. The goal here is to show that the neural networks can identify the tags. Let us now see the classification of images for the presence of a tag is done.

After creating images from the raw data, the next step is to train CNN to classify the images for the presence of the chosen tag. We have trained and tested a CNN called Xception Neural Network.

In the training process, the necessary preprocessing is done to prepare the data for training. The Xception neural network accepts the input image with four dimensions. So, the array of input images is reshaped to form a four-dimensional input. Since the Xception neural network has the output dimension as two-dimensions, the labels are one-hot encoded. One-hot encoding transforms the categorical data into a binary representation that could be provided to ML algorithms.

The array of images and their binary labels are fed to the neural network, and the model is trained for a significant number of epochs(20 epochs in our case) until

the validation loss gets saturated, i.e., if the validation loss decreases over the course of training and it stabilizes, that means the model has achieved a good fit. Standard values were used for hyperparameters like learning rate throughout the training. Call back is set to save the best-performing model with the highest validation accuracy throughout the training process. This best-performing model is used to test the test data. The result of this analysis is given in chapter 5.

Our analysis showed that the deep neural networks are compatible in providing good results on the images created from acoustic time series data. Since the deep neural networks showed better performance on the project data, we used UNet (a deep neural network) for image segmentation.

4.6 CNN for Image Segmentation

After the images are classified for the presence of a tag, our next goal was to design a model that identifies the pings from the tag in an image by performing image segmentation. For the image segmentation, we have used a neural network called UNet[11]. The detailed architecture of the UNet is explained in section 3.5.8.

The array of images generated from the RAT data and their binary masks generated from TAT data are used to train the UNet model. One main point to note when training the UNet model is that the image dimensions should be a multiple of 16. In our case, the image dimensions are 192 x 192, which is a multiple of 16. This is because, at each level of UNet, the size of the image reduces to half. Since there are four levels in UNet architecture, the size of the image gets divided by two for four times, i.e., the size of the image becomes (1/16) times the initial dimensions. Hence the initial dimensions of the image should be a multiple of 16.

Similar to Xception Neural Network, a standard UNet model also accepts the input only with four dimensions. The array of images is converted to a four-dimensional input to feed to the UNet neural network, and it is trained for a significant number of epochs(10 epochs in our case) until the validation accuracy and loss curves reach saturation point, i.e., the loss and accuracy values of validation data reach stability. Standard values are used for UNet hyperparameters throughout the training[11]. The segmentation outputs an image in which each pixel value ranges from 0 to 1. By carefully choosing the threshold(0.2 in our case; analysis showing why we chose 0.2 as

threshold is given in section 5.3), we can convert the image into a binary 2-dimensional array by assigning each pixel a '0' or '1' with the help of threshold. The best-performing model with the highest validation accuracy is saved and used to test the test data. When an image is given to the trained UNet model, the model should segment the pings from the tag. A detailed analysis of the experiment is given in chapter 5.

4.7 Inverse Mapping of Pixels to Pings

The next step after the image is segmented is to find the pings from the segmented pixels. Because the UNet model finds the pixels in an image associated with the pings, hence, by inverse mapping, the pings which occurred from the tag can be known. Inverse mapping of pixels to pings gives the list of marked pings in the image. The segmented image is divided into bins by computing the remainders for each of the pings in the 30-minute interval. In other words, the image is divided into x and y pairs where x is ping times, and y is the residual obtained by dividing the ping times by tag period. The equations of how the x,y values are computed are given below.
X =time elapsed (in minutes)
Y = time elapsed % tag period (in seconds)

From the x,y values, the bin numbers of each x,y pair are calculated in the image. The mask value of each bin is returned as an array. This is an array of 0's and 1's. The length of this array is equal to the length of the ping times. The locations at which there are 1's in the returned array give the locations of marked pings in the ping times array.

4.8 Implementing Post-Processing Filtering Algorithm

The final step is to apply the post-processing filtering algorithm. Application of these filtering algorithms will mark the previously unmarked pixels by the UNet. The filtering algorithms are rule-based processes whose performance depends on a few hand-tuned parameters like threshold, tolerance, and mode. The two algorithms applied are the 'subcode filter' and the 'trajectory filter' algorithms. A detailed

explanation of the algorithms is discussed in chapter 3.

Applying the subcode filter algorithm will mark the pixels by looking for the ping pairs that match the sub-period. The two modes used by the subcode filter are 'EARLIEST_TIME' or 'LARGEST_AMPLITUDE. When the 'EARLIEST_TIME' mode is selected, the ping pairs with the earliest arrival time are selected, and when the 'LARGEST_AMPLITUDE' mode is selected, the ping pairs with the largest combined signal amplitude are selected.

The next filtering algorithm applied after the subcode filter is the trajectory filter algorithm. In this algorithm, the pixels are joined by the trend line going through the ping pairs identified in the subcode filter algorithm, and the pings close to this trend line are also marked as pings from the tag.

4.9 Inference phase

Once the model is developed and trained, the sequence of steps in the project's inference or testing phase is given in the flowchart (Figure 4.8).

As shown in figure 4.8, in the testing phase or inference phase, the marking of the pings from the images is an easy process. The images are created from pings by choosing a suitable time window; for example, images are generated for each 30-minute duration. The number of pings in the specified window is extracted from raw data files by choosing a suitable time window(30-minute duration). A 2D-histogram image is created with the extracted pings and the test tag period. The generated image is passed through the UNet to segment the pixels from the tag. Once the image has been segmented for the pixels, the next step is to map the segmented pixels to pings. The final step is to apply the subcode filter, and the trajectory filter in the post-processing filter step to extract the final list of marked pings from the raw data. Finally, we obtain the list of marked pings from the tag in an image. Thus, the machine learning approach provides a useful solution to extract the pings from a chosen tag from the data containing pings from multiple tags.

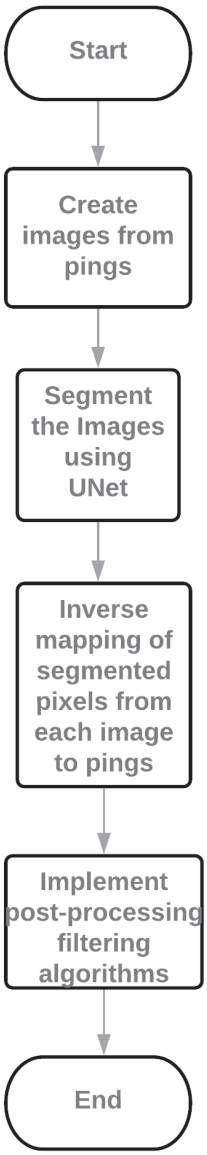


Figure 4.8: Flowchart illustrating the sequence of steps in the inference phase of the project

Chapter 5

Experiments and Results

In this chapter, we present our analysis of the information we extracted from the dataset, classification of the images for the presence of the tag, and segmenting the images for the pings from the tag. The experimental results discuss the performance of our proposed method.

5.1 Dataset

We have conducted our analysis on multiple datasets and tested the performance of the model on these datasets. In this chapter, we would like to present our detailed analysis of one of the datasets which we used the most. Later we compare the performance of the models on another dataset which is recorded in a different environment.

The name of the dataset we used for our analysis is recorded in a Shallow River Environment(SRE), so we refer to it as the 'SRE dataset.' The dataset contains 426 raw data files (RAT) and 426 marked data files (TAT). A single file in this dataset contains data of one day from a single hydrophone. In the TAT files, the tag period is in the format as explained below.

Format: PPPP.PP-SS where PPPP.PP indicates period as a fixed-point number of milliseconds, and SS indicates subcode index (01, 02, ..., 31)

Subcodes: { 225, 248, 270, 291, 311, 330, 348, 365, 381, 396, 410, 423, 435, 446, 456, 512, 523, 535, 548, 562, 577, 593, 610, 628, 647, 667, 688, 710, 733, 757, 782}

Example: The example tag period is 9152.00-24. The period here is 9152.00 in milliseconds, and the subcode is 628 milliseconds (the 24th interval in the above list). Subcode defines the separation between the primary pulse and the secondary pulse

in milliseconds.

The contents and the meaning of each column in the file are explained in section 1.3. Below is some of the information about the dataset.

- The dataset is recorded for 43 days.
- The total number of hydrophones used is 10.
- Pings from 35 different tags are recorded. Out of 35 tags, 25 are fish tags, and the other 10 are time-synchronous tags used to synchronize the receivers with the GPS clock.
- Total number of pings in the dataset is 21.5 Million.

Figures 5.1 and 5.2 show how the pings are distributed over the hydrophones and the tags. From figure 5.1, we can see which hydrophone captured the most number of pings. From figure 5.2, we can get the information of which tags were present more within the range of receivers

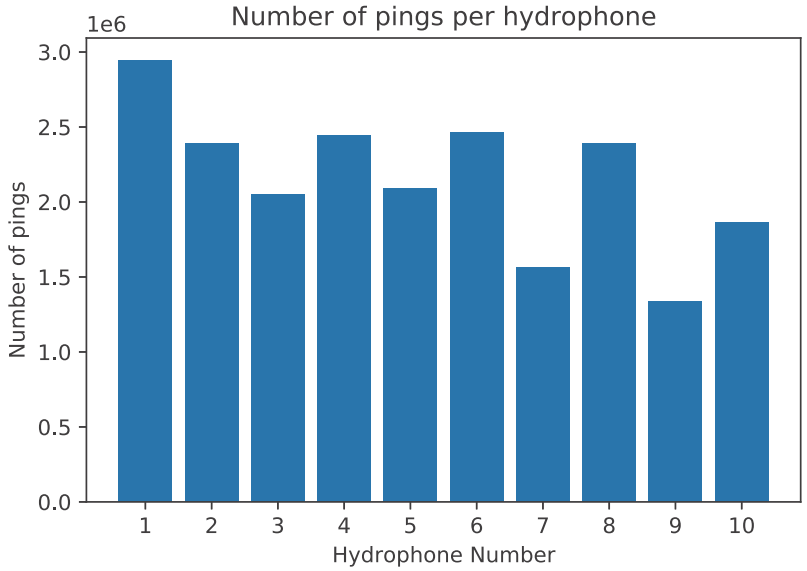


Figure 5.1: Bar plot showing the distribution of pings over the hydrophones

Tags with a period in the range of 9-10 seconds are fish tags, and the other tags with a period greater than 10 seconds are time-synchronous.

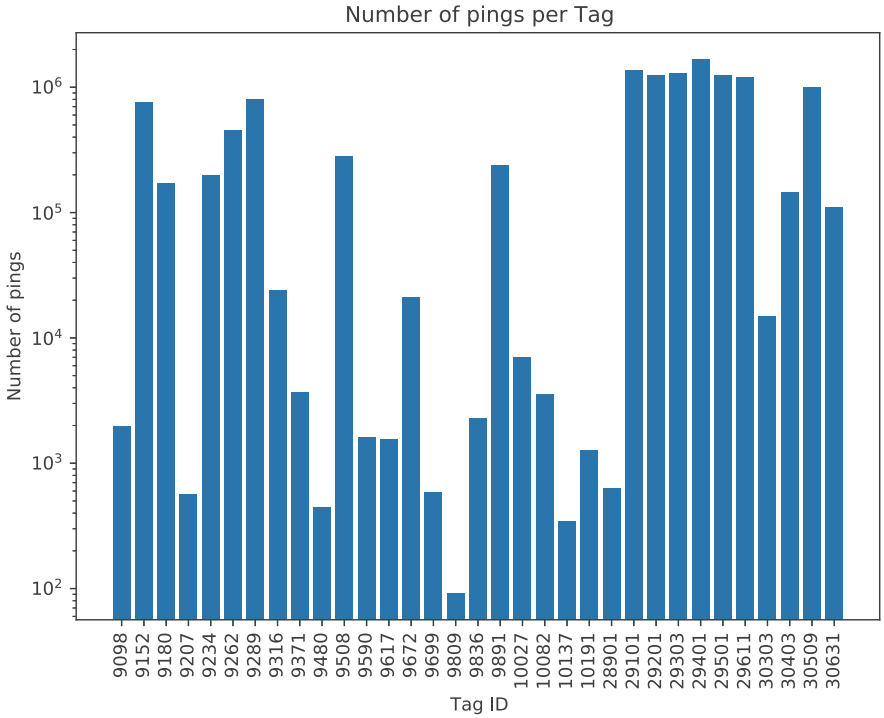


Figure 5.2: Bar plot showing the distribution of pings over the tags(tag period is given in milliseconds)

Figure 5.1 show that there are some fluctuations, but the hydrophones have the similar number of pings. Variations are only of the order of a factor of two or less. This could be taken to indicate a certain level of uniformity or similarity across the hydrophones. One might expect the UNet to generalise well across the hydrophones. However, this distribution can be different for different datasets i.e., other datasets can exhibit greater variation between hydrophones. Figure 5.2 shows that there is a greater variation between the number of pings from each tag in the order of atleast a factor of 10. For UNet to generalise well across the tags, we have chosen tags with different appearance rates for training and testing (discussed in section 5.3).

Out of all tags, we chose three tags with the greatest number of pings and only a single hydrophone, hydrophone:1, for our initial analysis. The total duration of the tags present at each of the hydrophones is given as the cumulative presence of the tag. The cumulative presence of the three chosen tags in all the hydrophones is given in figures 5.3, 5.4, and 5.5 in the form of bar plots.

The distribution of pings in bar plots 5.3, 5.4, and 5.5 show that the pings have

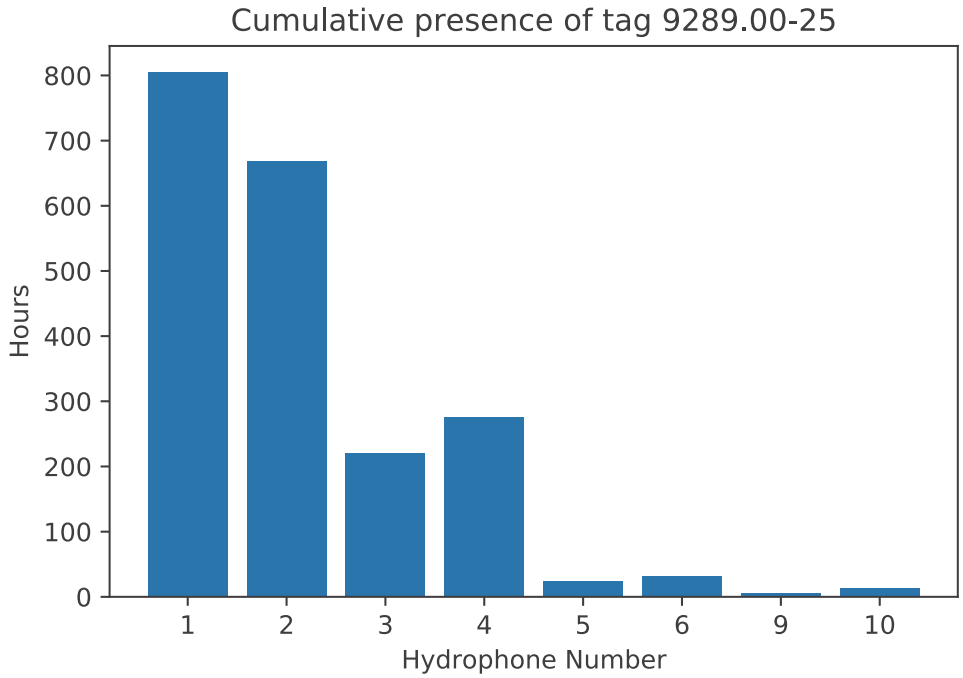


Figure 5.3: Cumulative presence of tag 9289.00-25

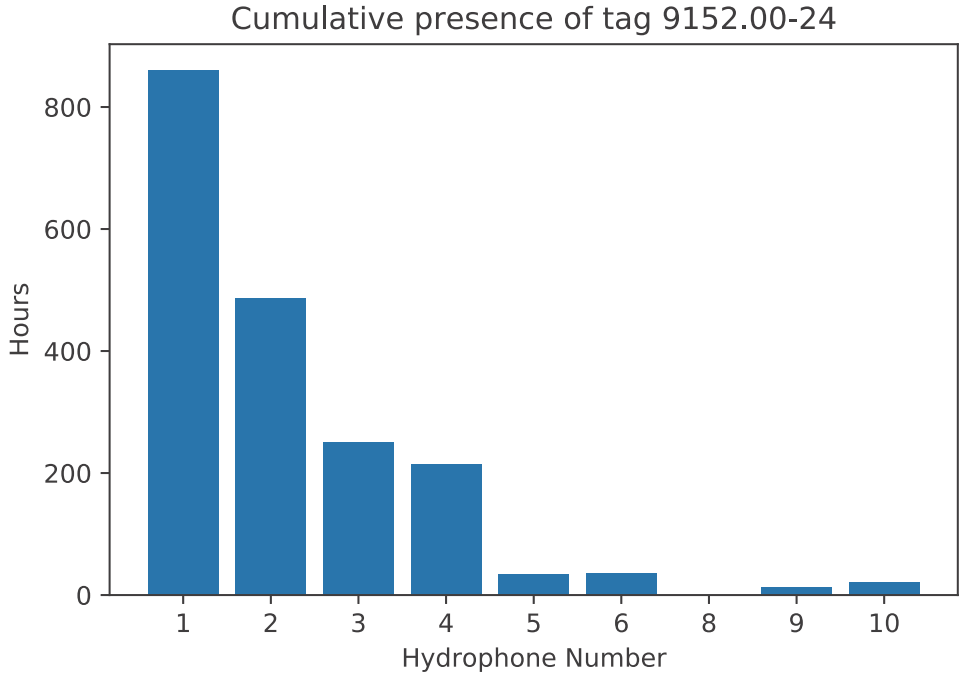


Figure 5.4: Cumulative presence of tag 9152.00-24

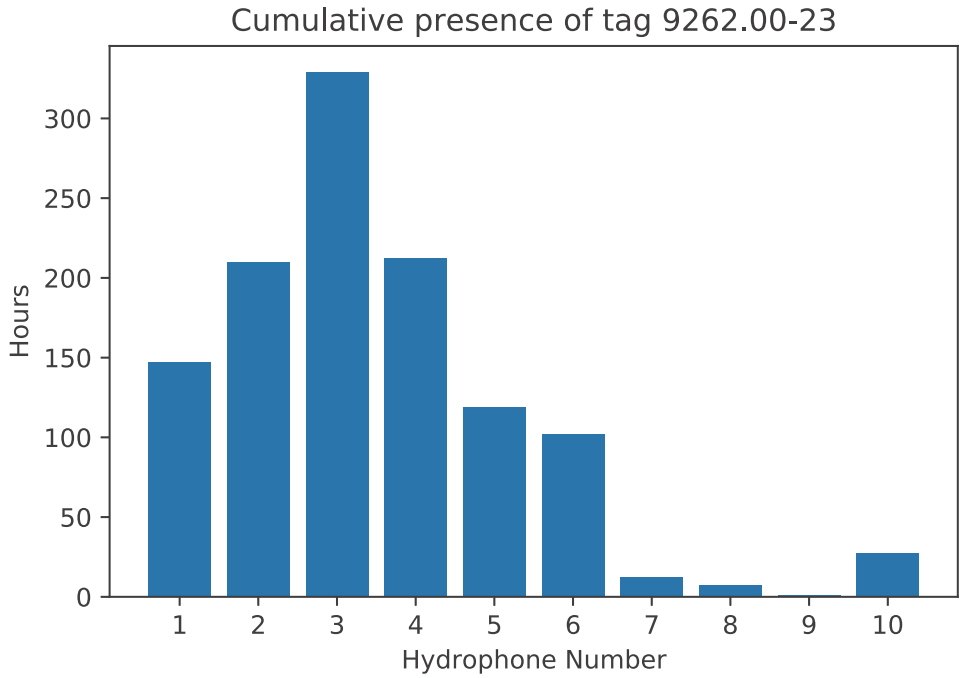


Figure 5.5: Cumulative presence of tag 9262.00-23

different distributions for different hydrophones. For generating models, we used a subset of data that has most of the data and another subset of data that has most as well as comparatively fewer data. Thereby creating the images from most frequently appearing as well as the rarely appearing tags.

The total duration of the three chosen tags is divided into 30-minute intervals, and the data is split into training, validation, and test sets using the procedure mentioned in section 4.1. There are 7918 30-minute intervals in this data. 60% of the intervals are divided into training data, 20% into validation data, and the remaining 20% into test data. There are 4756 training intervals, 1598 validation intervals, 1564 test intervals for each tag period. These 30-minute intervals are converted to images for each of the three tags selected. The procedure to create images is explained in section 4.1. The two augmentation techniques discussed in section 4.4 are applied to these images in training and validation data sets. After augmentation is applied, the total number of training samples is 85608; validation samples are 28764, and 4692 test samples. Similarly, the binary labels are generated for training, validation, and test samples, as mentioned in section 4.2.

5.2 Training an Xception Neural Network

An 'Xception' neural network[38] is trained on these 85608 training samples and 28764 validation samples. The model is trained for 20 epochs until the training and validation accuracy reaches saturation, and the best-performing model with maximum validation accuracy is saved. The plots of the accuracy of training and validation data are given in figure 5.6.

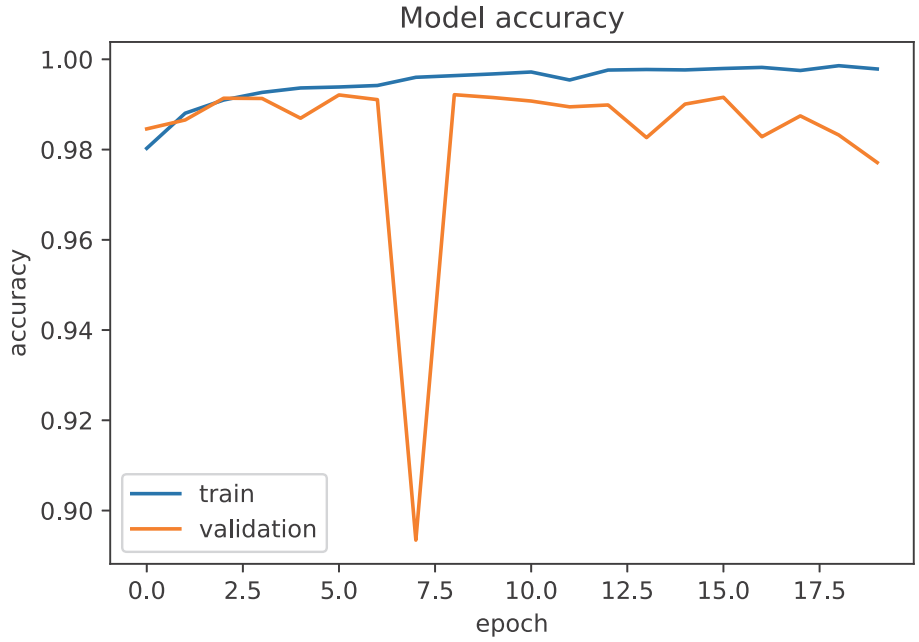


Figure 5.6: Accuracy plot of training and validation data of Xception Neural network

As seen in figure 5.6, there is a dip in the validation accuracy at epoch 7. Though training the model multiple times and with different random initialization of the model, the dip seems to appear over the course of the training. The reason for the dip was unknown. Despite the dip, all models trained with different random seed initialization converge to the same solution in a reproducible manner. The saved model with the highest validation accuracy is tested on 4692 test samples. The model is found to be 98.31% accurate in predicting the labels correctly. The other model evaluation parameters, precision, and recall are 0.9743 and 0.9724, respectively. The distribution of positive and negative samples is given in table 5.1. This table shows that there are a large number of negative samples in the training and validation

datasets. This is because the second augmentation technique we implemented generates a more number of negative samples. For example, out of six samples generated for a tag period, four of them will be negatively labeled in augmentation 2. The implications of unbalanced training set for the inference were not systematically investigated. For further explanation, please refer to section 4.4.2. Since the augmentation is not implemented in the test set, it looks more balanced.

Threshold=10 pings	Positive Samples	Negative samples
Training data	11360	74248
Validation data	3852	24912
Test data	2030	2662

Table 5.1: Distribution of positive and negative samples among datasets for Xception Neural network analysis

To determine if the model is trained with 'good fit' or 'under fit' or 'over fit,' validation data plays an important role. In training, we compute the validation accuracy and save the model when the validation accuracy is maximum to avoid overfitting. To check if the model is overfitted, we also tested the trained model on a different subset of data to see if the model can generalize the data well.

The Xception neural network model saved with the highest validation accuracy is now tested on different data that is not used for training, and the model performance is evaluated. The results are shown in Table 5.2.

Hydrophone and Tag	Accuracy	Precision	Recall	Number of positive samples	Number of negative samples
Hydrophone:2 Tag:9891.00-27	99.86%	97.38%	99%	301	7617
Hydrophone:2 Tag:9234.00-28	99.83%	94.68%	91.75%	97	7821
Hydrophone:2 Tag:9289.00-25	95.18%	99.62%	90.41%	3841	4077
Hydrophone:3 Tag:9289.00-25	94.92%	97.53%	84.28%	2253	5665

Table 5.2: Xception neural network performance on various test dataset

The accuracy is simply the ratio of correctly predicted observations to the total number of observations. Whereas precision is the ratio of correctly predicted positive observations to the total number of positive observations, and recall is the ratio of correctly predicted positive observations to all the observations in the actual class (not predicted class). The use of the Xception Neural network is to classify the images for the presence of the chosen tag.

The Xception Neural network was used to test if the acoustic data involving images can be used for Deep learning approaches. Since the deep neural network like Xception neural network detected the pings with more than 95% accuracy, we used UNet, another Deep neural network for image segmentation tasks. We feed the images to the UNet model for segmentation in our project final pipeline once the images are created.

5.3 Analysis with UNet Model

5.3.1 Training the model

Training the UNet model involves the model to segment the image to find the pings from a chosen tag. To train the UNet model, for all the training, validation, and test samples, binary masks are generated using the procedure mentioned in section 4.3. For UNet analysis, another tag, '9316.00-25', with a comparatively smaller number

of pings than the other chosen tags is chosen. The choice of four tags with three tags having the greater number of pings and the fourth tag having fewer pings is made to maximize the number of images containing pings while still exposing the model to both frequently and rarely appearing tags in order to achieve satisfactory performance. Once the binary masks are generated for all the images, a UNet model[11] is trained for a significant number of epochs(20 epochs in our case) until the validation loss and accuracy reaches saturation. However, the validation accuracy reached saturation after two epochs. The accuracy is close to 0.99. This happens because of class imbalance[53]. Since the pixels from the tag (pixels with label '1') are less compared to the other pixels (pixels with label '0'), i.e., for example, in the case of a full track, the fraction of positive pixels to negative pixels is low which is close to 1%, the accuracy of the model will be more than 99% even if the model identifies all the positive pixels wrongly. So, accuracy is not a correct measure if there is a large class imbalance, as in our case. Hence, we used F1-score as our performance evaluation metric. Since the accuracy is close to 0.99, we plotted (1-accuracy) at each epoch in a logarithmic scale as shown in figure 5.7. The validation accuracy continues to improve slightly after the first epoch.

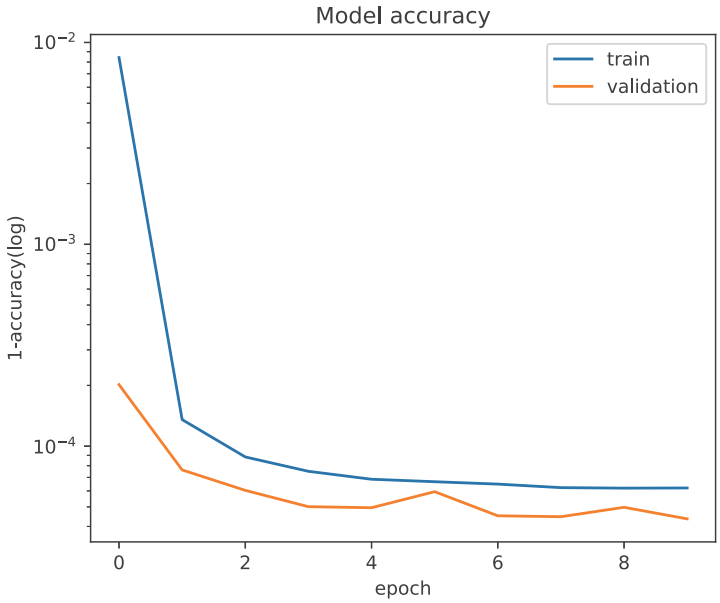


Figure 5.7: Accuracy plot for training and validation data on UNet model

The best-performing model with the highest validation accuracy during model training is saved and used on test data. An example of how the UNet segments the pixels in the image is shown in figures 5.8, 5.9, 5.10.

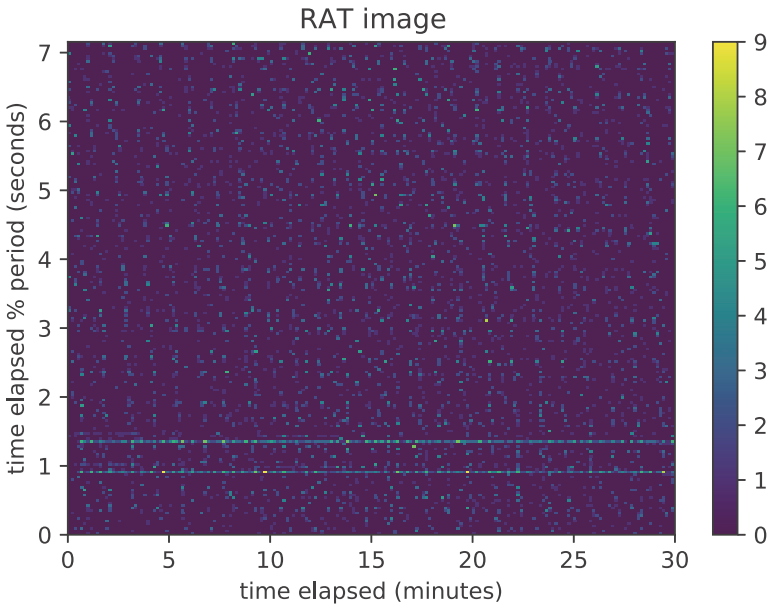


Figure 5.8: Image generated from RAT data

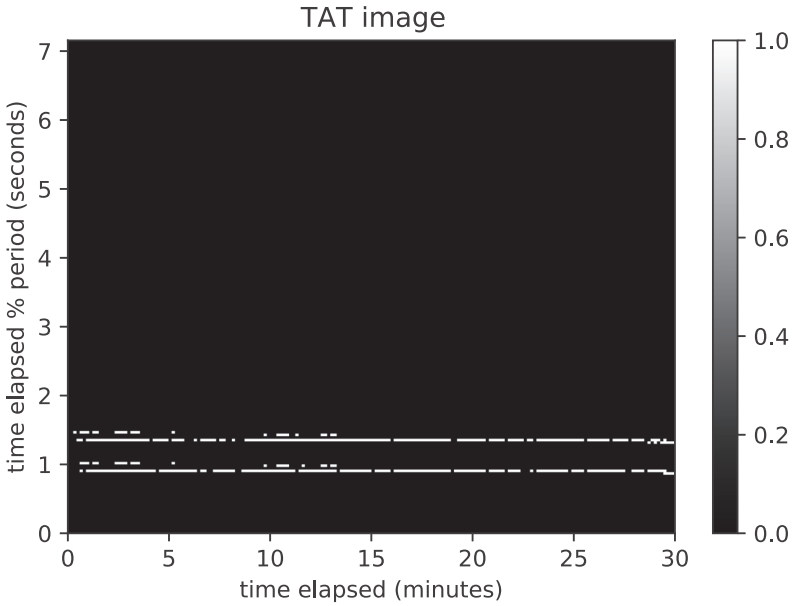


Figure 5.9: Binary mask generated from TAT data

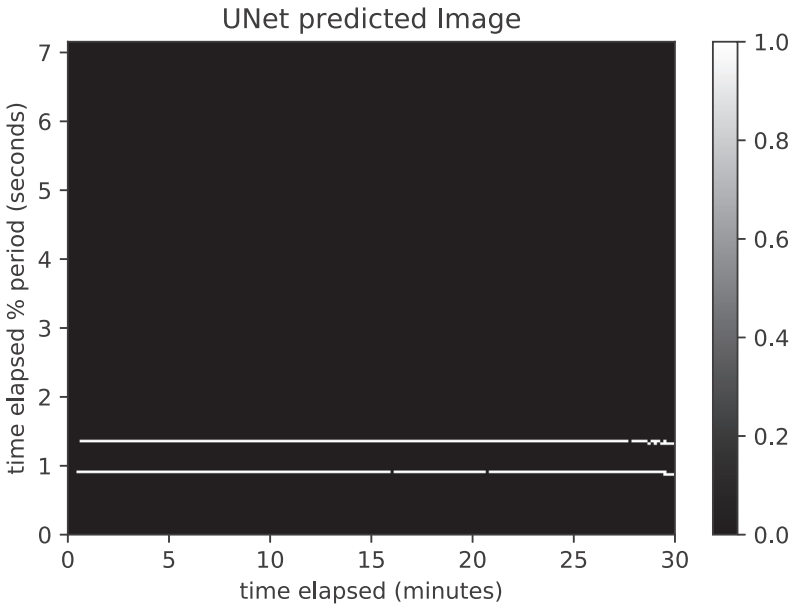


Figure 5.10: Image predicted by the UNet model

The UNet mask predicted by the model is an image, and each pixel in the image has a value in the range of 0 to 1. All the pixels in the image are converted to binary labels by carefully choosing a suitable threshold value. If the pixel value in the image is greater than the threshold, the pixel is labeled 1 and 0 otherwise. Hence we obtain the UNet mask as a 2-dimensional binary array.

To choose a threshold that converts UNet output mask to a binary 2-dimensional mask, we trained two models, one (Model-1) with a small subset of data which contains the data of only a single hydrophone and a tag combination, and another model (Model-2) was trained with a large subset of data compared to the model-1 with data containing multiple tags and hydrophones.

- Model-1 is trained with Hydrophone:1 and Tag: 9289.00-25
- Model-2 is trained with Hydrophones: 1 to 6 and Tags: 9289.00-25, 9152.00-24, 9262.00-23, 9316.00-25

5.3.2 Choosing Optimal Threshold

The two models are tested on six different test datasets, and the Precision-Recall curve is plotted by varying thresholds from 0.0001 to 0.9 in steps of 0.05.

The test datasets to plot Precision-Recall curves are:

- Hydrophone:1, Tag:9180.00-29
- Hydrophone:3, Tag:9180.00-29
- Hydrophone:5, Tag:9180.00-29
- Hydrophone:1, Tag:9234.00-28
- Hydrophone:3, Tag:9234.00-28
- Hydrophone:5, Tag:9234.00-28

The Precision-Recall plots are given in figures 5.11 and 5.12 for the two different models tested on six different test datasets.

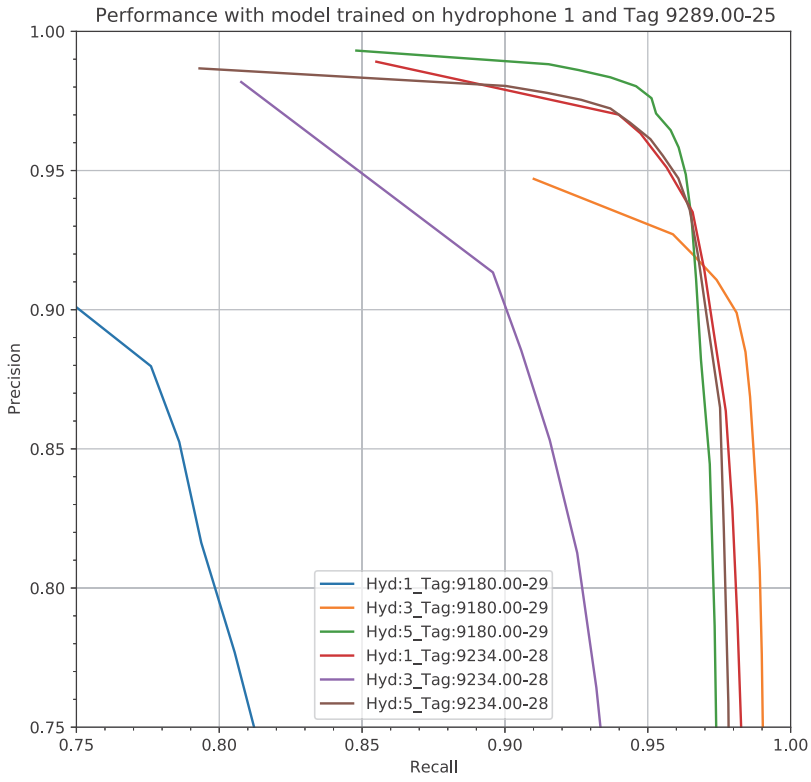


Figure 5.11: Precision-Recall plot for model trained on single hydrophone and single tag (Model-1)

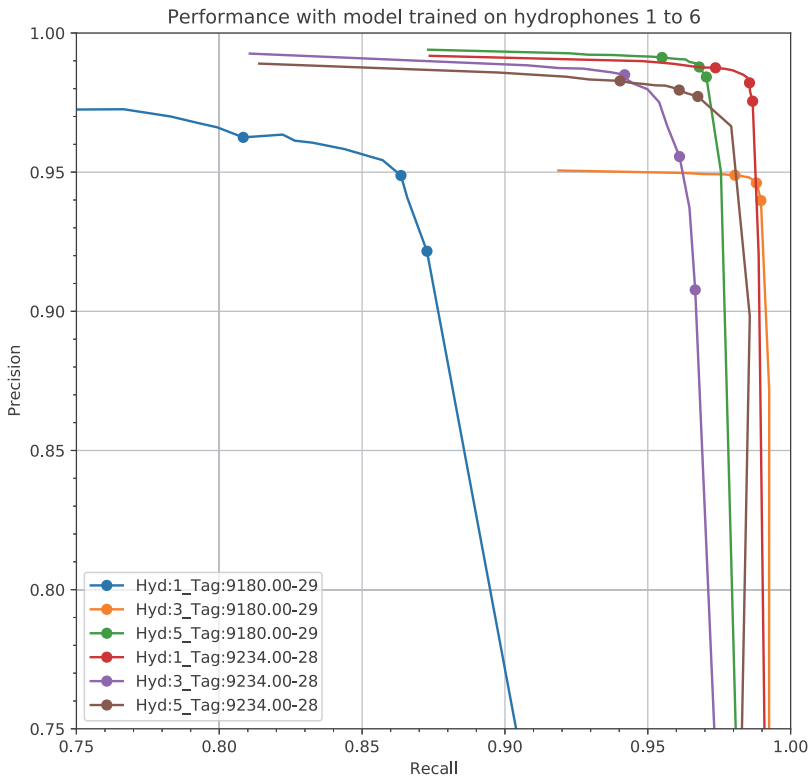


Figure 5.12: Precision-Recall plot for model trained on multiple hydrophones and multiple tags (Model-2).The points indicate threshold 0.1, 0.2 and 0.5 respectively from right to left

Figures 5.11 and 5.12 show that the precision and recall are more than 90-95%. These two plots summarize the trade-off between the true positive rate and the positive predictive value for a model using different thresholds. In other words, the Model-1 is able to detect atleast 80% of pings in an image and correctly predict atleast 90% of the detected pings. Similarly, the Model-2 is able to detect atleast 91% of the pings and correctly predict atleast 95% of the detected pings in all the datasets. Clearly Model-2 performs better than the Model-1. The curves in figure 5.12 have a sharp bend for the thresholds 0.1 to 0.2. So, an optimal threshold of 0.2 is chosen to label the pixels in the image.

5.3.3 Performance at Pixel and Ping Level

Two types of analysis are done after the UNet segmentation is done.

- Performance analysis at the pixel level.
- Performance analysis at the ping level.

For pixel-level analysis, the pixel values of the binary mask generated from TAT data and the mask predicted by the UNet are considered. For each of the images, precision, recall, Jaccard index, and F1 score are calculated. These metrics are explained in section 3.7.

Test Data

A test dataset containing different Hydrophone – Tag combinations is created to perform pixel-level and ping-level analysis. The test data contains below Hydrophone – Tag combinations.

- Hydrophone:1, Tag: 9234.00-28
- Hydrophone:2, Tag: 9891.00-27
- Hydrophone:3, Tag: 9508.00-19
- Hydrophone:4, Tag: 9234.00-28
- Hydrophone:5, Tag: 9180.00-29
- Hydrophone:6, Tag: 9508.00-19

For the above six test sets, the overall precision and recall are computed at the pixel level. Then the post-processing filtering algorithms are applied to compute the metrics at the ping level. To compute the precision and recall, the data in the TAT files is taken as the ground truth labels.

The test sets are tested on the models trained on small data (Model-1) and the model trained on large data (Model-2). Figures 5.13 and 5.14 are the plots showing the performance of the UNet model on the different test datasets at pixel level compared to manual marking. The dashed line at 100% shows the level of manual marking.

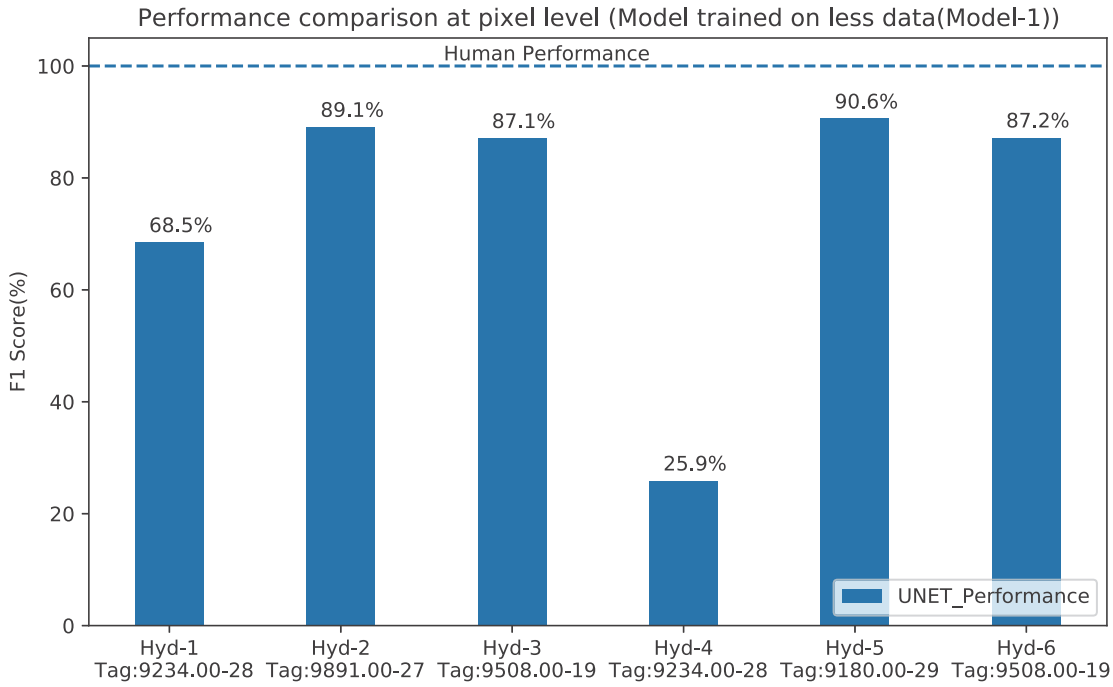


Figure 5.13: Performance comparison at pixel level for the Model-1

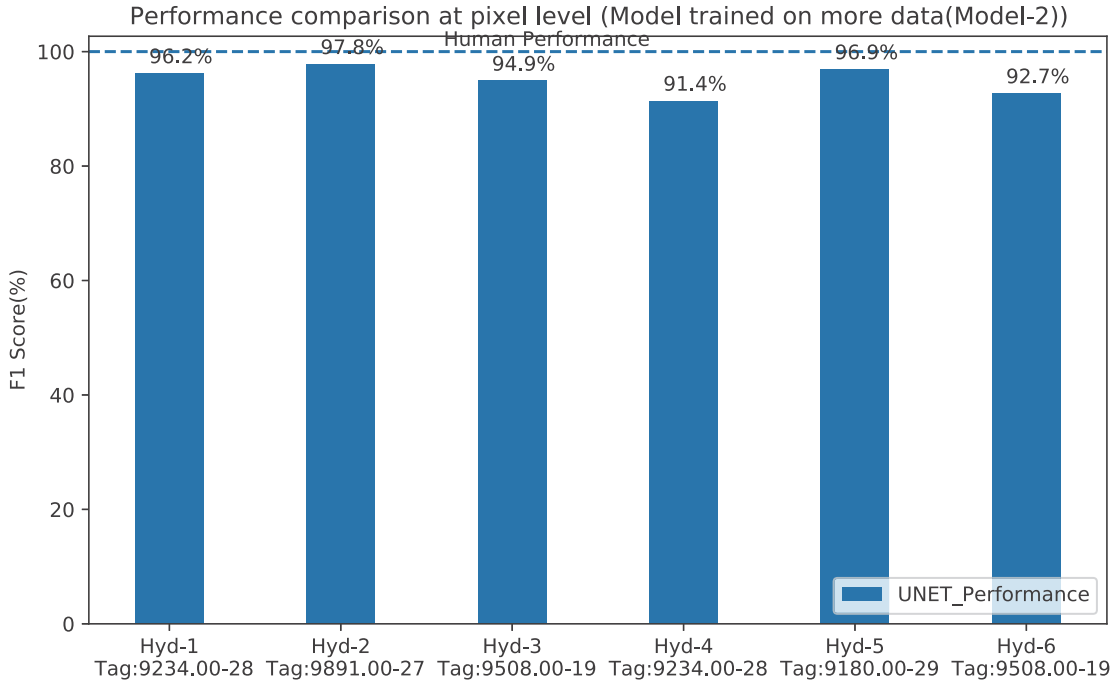


Figure 5.14: Performance comparison at pixel level for the Model-2

Now the pixels are converted to pings using the inverse mapping procedure mentioned in section 4.7, and both the models are tested on the test datasets. The results are given in figures 5.15, 5.16.

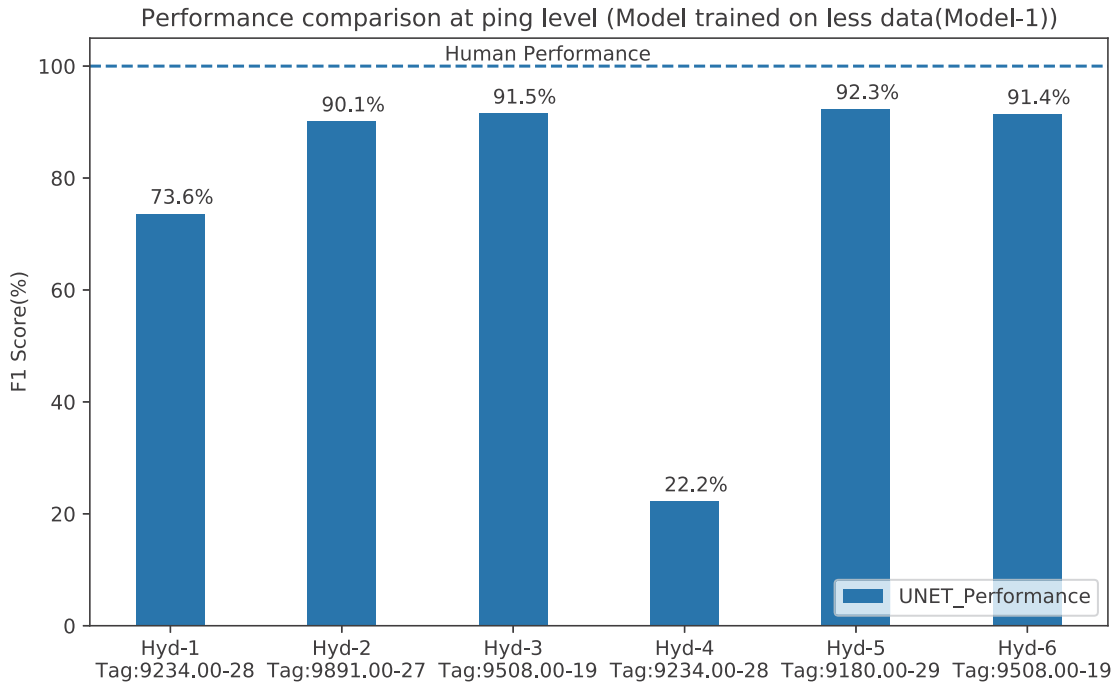


Figure 5.15: Performance comparison at ping level for Model-1

Increasing the training data improved the model's performance, as seen in plots 5.13 and 5.14 in all the test cases at pixel and ping levels. Looking at the plots 5.15 and 5.16, the UNet performance on all the test sets is between 94% to 99%, which indicates that the model can segment the pings close to human-level annotations. However, the ping level performance is low compared to the pixel level performance in few cases. When the image is created, multiple pings can be assigned in a single pixel, and when inverse mapping is done, the pixel to ping conversion can map only one pixel to a ping, ignoring the other pings that should have been marked. Hence, the performance at the ping level can be low in a few cases. The models perform poorly in a few cases in which the tracks are faint in the images. These models are nearly sufficient in producing the study results that involve the presence or absence of a fish tag. Also, the same performance of the models may not be reproducible on other datasets as the other datasets might have data recorded in different environments or

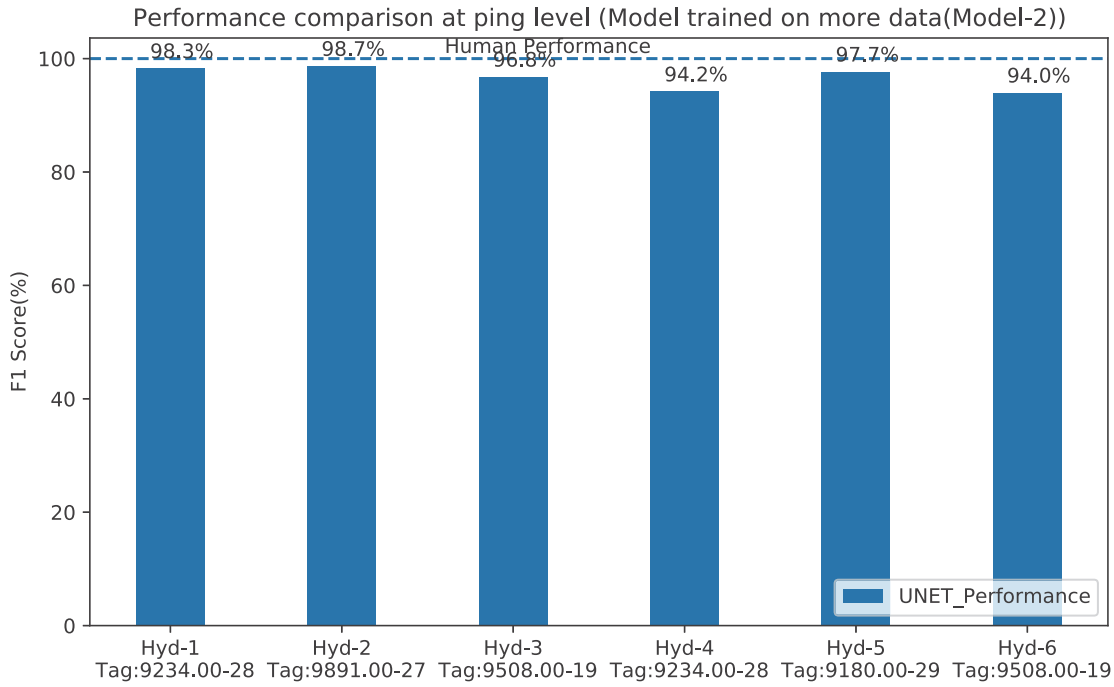


Figure 5.16: Performance comparison at ping level for the Model-2

may have different data distributions. The analysis of the models on other datasets is discussed in the subsection 5.3.5.

Moreover, in the six test datasets, two datasets, Hydrophone:4, Tag: 9234.00-28 and Hydrophone:6, Tag: 9508.00-19, have less performance than the other test sets. Most of the images in these Hydrophone-Tag combinations were noisy images. The pings in the images were not clearly visible, and the model could not identify the pings correctly. Hence there were many false positive and false negative pings in the images resulting in low precision and recall. Therefore, the F1-score is low for these two test datasets.

Bar plots in figures 5.17, 5.18, 5.19 show how the precision, recall, and the Jaccard index are distributed in all the test datasets.

From figures 5.17, 5.18, 5.19, we can infer that more than 75% of the images have high precision, recall, and Jaccard coefficient. In comparison, the plots say that many images have zero precision, recall, and Jaccard coefficient. The images in which the metrics are zero do not have any pings present, but the images predicted by the UNet have a couple of pings predicted by the model, which are false positives. Most of the

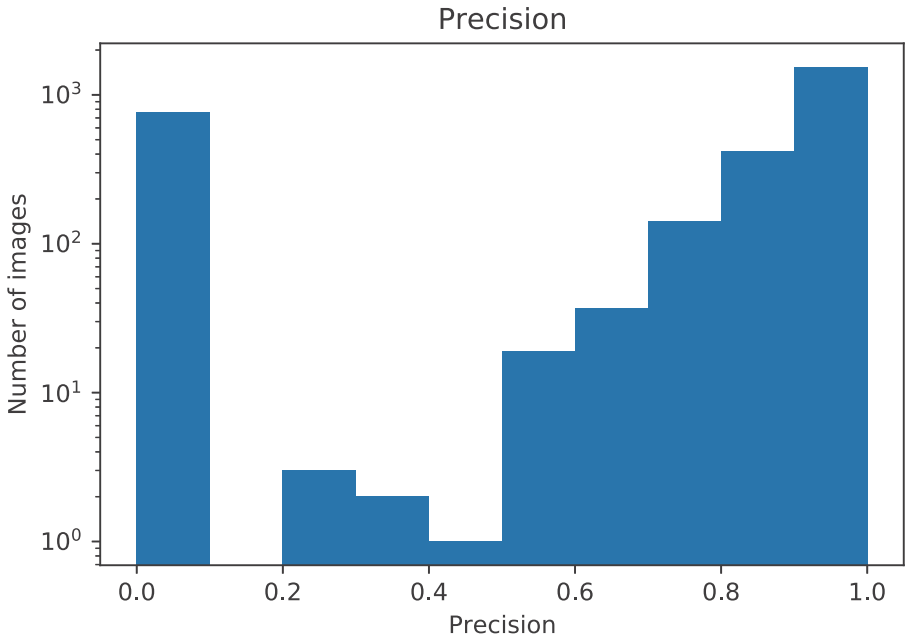


Figure 5.17: Bar plot showing the distribution of precision

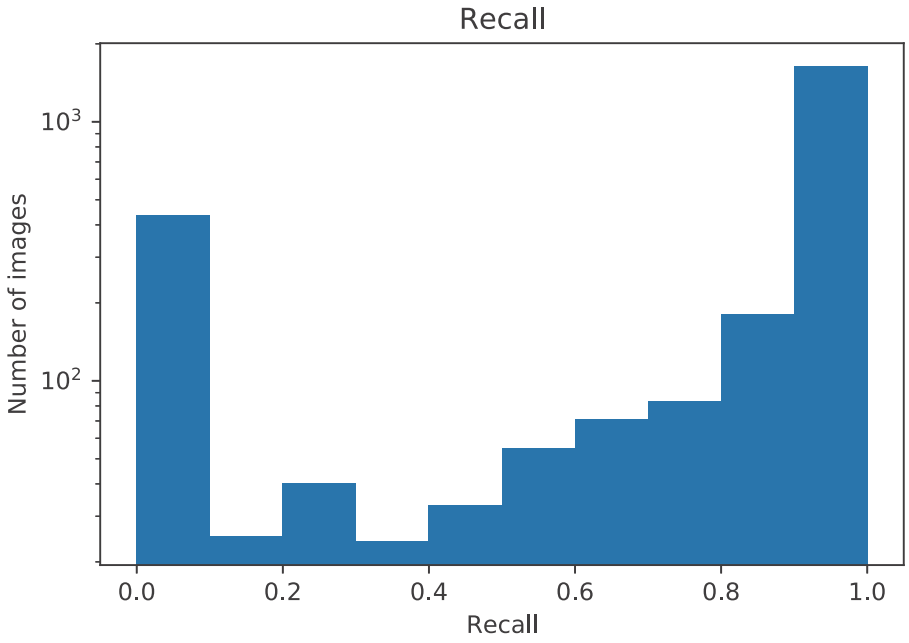


Figure 5.18: Bar plot showing the distribution of Recall

images were noisy, so the pings were not visible, resulting in many false positives and false negatives. A noisy sample image is shown in figure 5.23, where no pings were

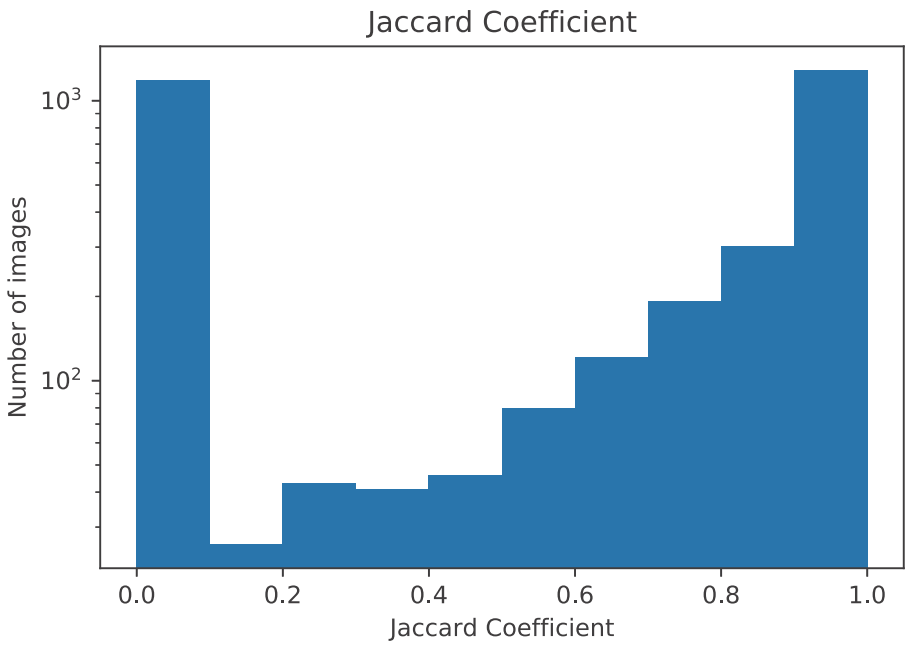


Figure 5.19: Bar plot showing the distribution of Jaccard Index

visible, but the model identified a couple of pings in a similar type of images.

5.3.4 Models Performance Compared to Auto Marking

The same dataset which had the marking done by the automatic process in the 'MarkTags' application was provided by Innovasea. The marking in this dataset is compared with the markings done manually by the annotator, and the performance is computed. The marking in the manually marked files is taken as the ground truth labels to compute the performance metrics of auto-marking.

A plot showing the comparison of the performance of auto marking and the UNet marking to the manual marking at ping level is given in figure 5.20.

In all the cases, the UNet model outperforms the auto marking annotations by more than 45%, and also, the UNet model performance is close to the human-level performance. The auto marking and UNet marking results show that UNet produces better auto marking results than the MarkTags' auto marking approach. However, though the UNet marking is close to manual annotations in the SRE dataset, its performance on other datasets is comparatively less. The results of the UNet performance on other dataset is discussed in the next subsection.

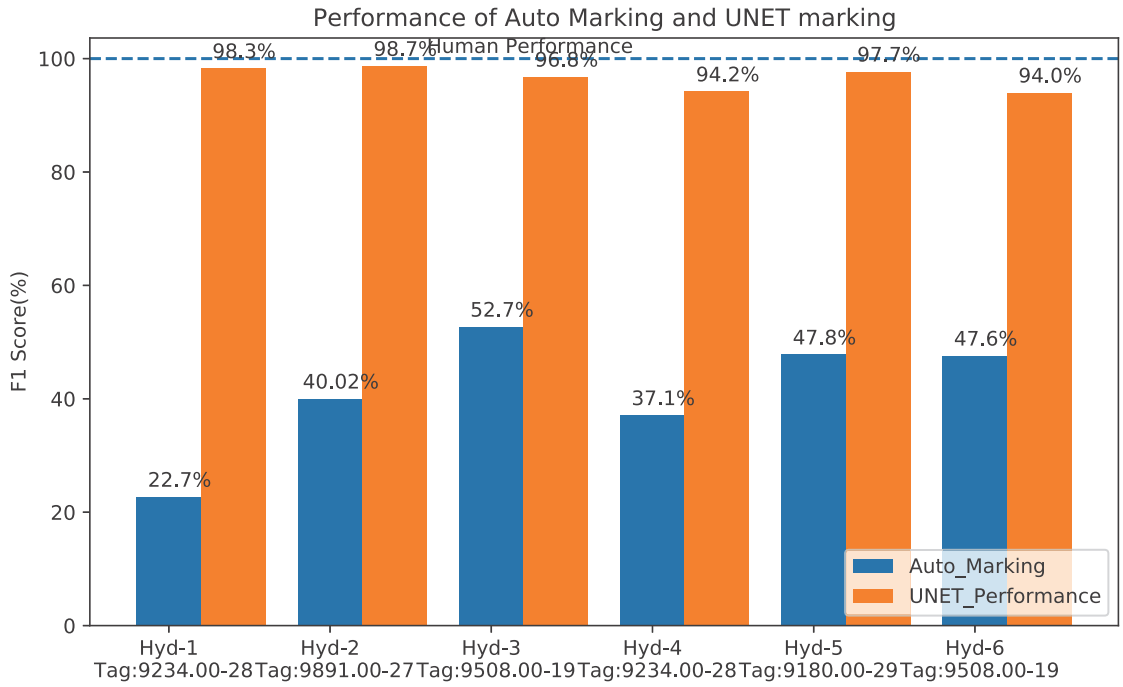


Figure 5.20: Performance comparison of auto marking and UNet marking to manual marking

5.3.5 Reproducibility on Other Datasets

The model trained on SRE data is tested on a completely different dataset called 'Hydrodam' data recorded in a highly noisier environment. The dataset is recorded near a hydropower dam. Hence the data recorded contains noise from external sources, and also the dataset contains many multi-path pings. The explanation about multi-path pings is given in section 3.3. Another three models trained on different data distributions of the hydrodam data are compared with the model trained on SRE data. This comparison analysis is done to see if the model trained in one environment can identify the pings in another environment. The performance comparison is made at both pixel and ping level on five different test datasets of hydrodam data, and the plot showing the F1-score of four models on five test sets is plotted. The details about the models and the comparison plot are shown in figures 5.21 and 5.22.

- Model-1: SRE Model: Best performing model of SRE dataset.
  - Training data: Trained on Hydrophones: 1 to 6 and Tags: 9289.00-25, 9152.00-24, 9262.00-23, 9316.00-25 of SRE data.

  - Image dimensions: 192 x 192 for each image.

- Model-2: Hydrodam Model-1: Trained on fewer data.
  - Training data: Model trained on Hydrophones: 203, 204, 205, Tags: 5875.13, 6547.15, 7156.14, 7408.26. It contains data of three hydrophones and four tags but has less number of pings. The number of pings used for training is 79427.
  - Image dimensions: 192 x 192 for each image.

- Model-3: Hydrodam Model-2: Trained on more data.
  - Training data: Model trained on Hydrophones: 206, 207, Tags: 5329.17, 7009.06, 6505.13. It contains two hydrophones and three tags but has more pings than the data used for Model-2. The number of pings used for training is 636876.
  - Image dimensions: 192 x 192 for each image.

- Model-4: Hydrodam Model-3: Trained on the same data as Model-2 but with increased image size.
  - Training data: The training data is the same as the Model-3. Hydrophones: 206, 207 and Tags: 5329.17, 7009.06, 6505.13 are used as training data.
  - Image dimensions: 384 x 384 for each image, double the standard size of each image used in other models.

The Hydrodam Model-1 and Hydrodam Model-2 were trained on fewer and more data respectively to investigate if the amount of training data would have an impact on the model performance. Similarly the dimensions of the image is doubled in training the Hydrodam Model-3 to see if this would impact the model performance. Doubling the image size will make the pings visible more clearly and provides more detailed information to the model but makes the computation slower.

The plots showing the comparison of the F1-score of five different test datasets tested using the above four models are given in figures 5.21 and 5.22.

The performance of the models on the hydrodam data is in the range of 70-80%. From plots 5.21 and 5.22, the below conclusions can be drawn. On hydrodam

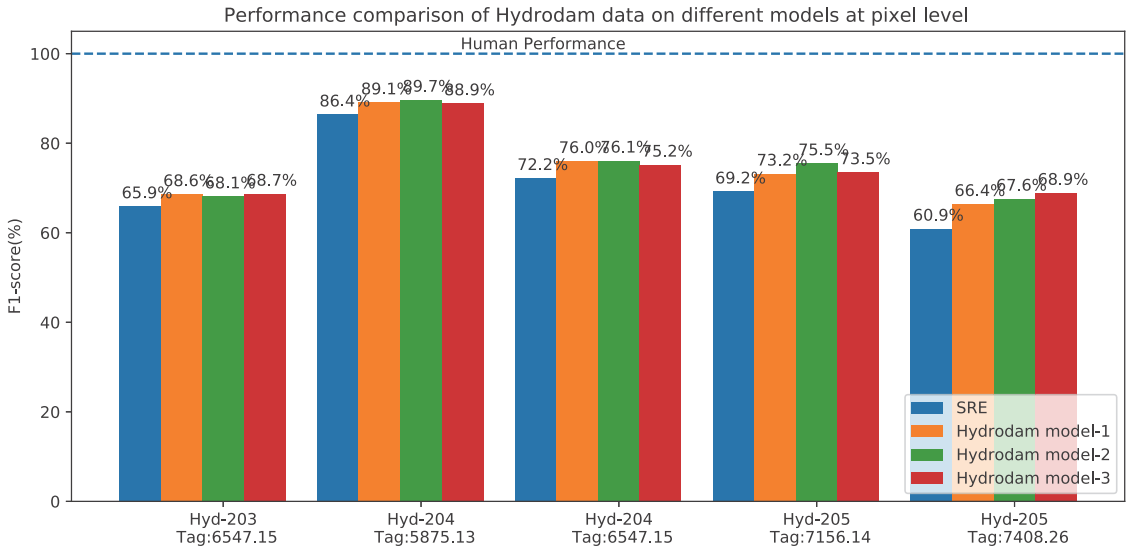


Figure 5.21: Performance comparison of hydrodam data on different models at pixel level.

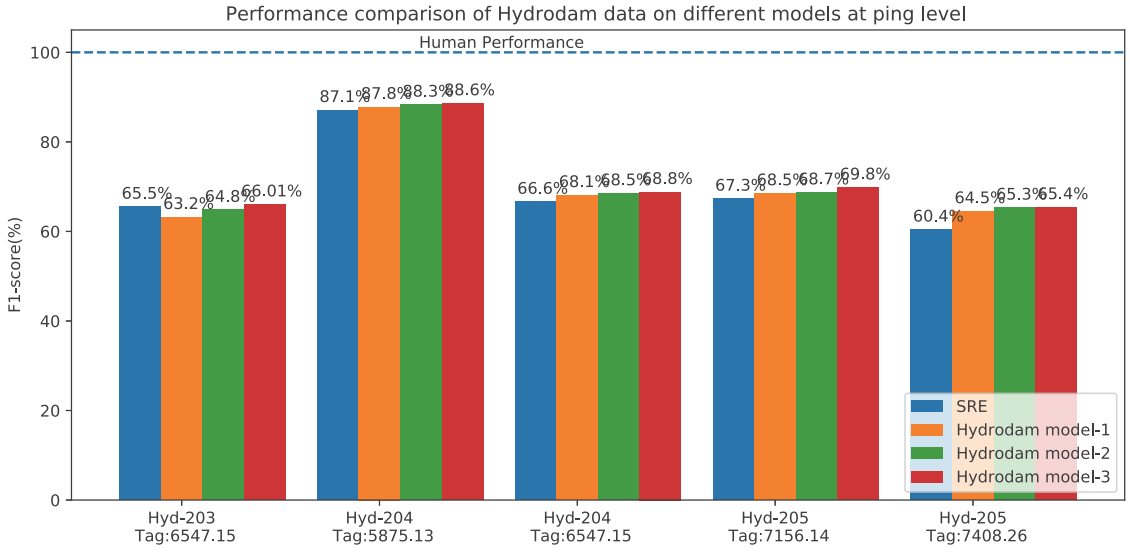


Figure 5.22: Performance comparison of hydrodam data on different models at ping level.

test data, the hydrodam models seem to perform better than the SRE model since the training data of hydrodam models include noisy data compared to the SRE data. There was a 1% improvement in the model's performance when the hydrodam model-1 and hydrodam model-2 are compared. This is because the data used for training model-2 has more pings. Increasing the image's dimensions by two times showed an

improvement of 1% in the performance when the hydrodam model-2 and hydrodam model-3 are compared at both ping level and pixel-level analysis.

The overall performance of the models on Hydrodam data is poor compared to the SRE data because of the noisy nature of the data. The model struggles to identify the pings from the noisy images, and we also identified that the manual annotations were not accurate. Few images had pings visible in the image but were not marked by the annotator. Sample images that are noisy and which show incorrect marking are given in figures 5.23 and 5.24.

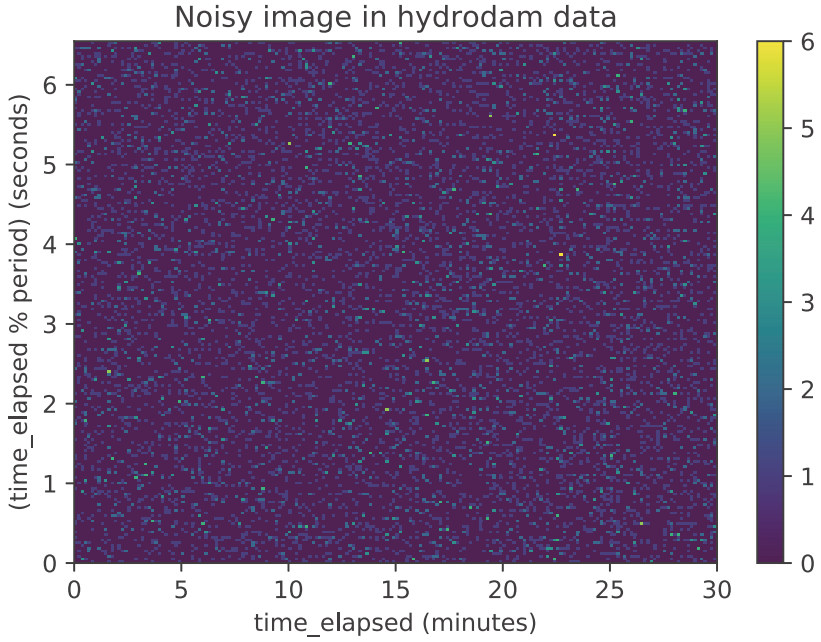


Figure 5.23: Sample noisy image in hydrodam data

Looking at the images in figures 5.24 and 5.25, in the duration 0 to 10 minutes of the image 5.24, few pings are visible but were not marked by the manual annotator (figure 5.25) while providing the dataset. There are many similar images in which the data was not marked correctly in the hydrodam data.

5.4 Remarking Analysis

Another expert at Innovasea was requested to re-annotate a subset of whole data which had marking errors.

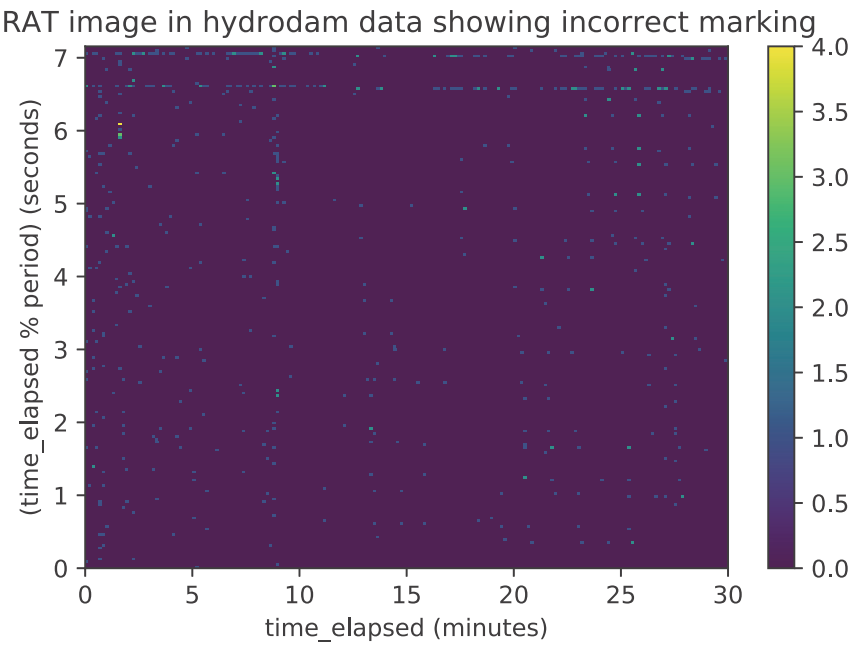


Figure 5.24: Sample RAT image showing incorrect marking

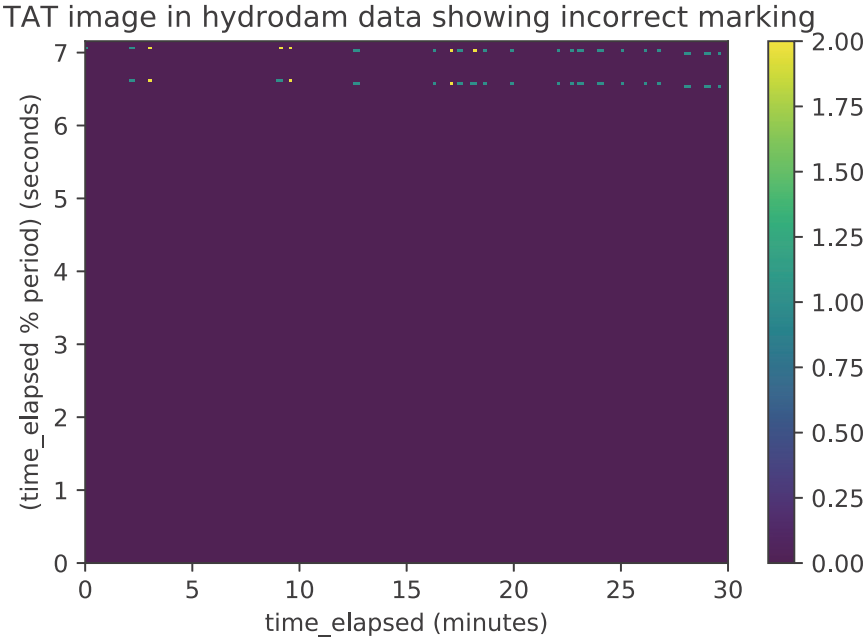


Figure 5.25: Sample TAT image showing incorrect marking

The information about the remarked data is given in below.

- Total duration of the data: 24 hours. (Day 310 in the hydrodam dataset).
- Hydrophones: 203 and 204; Tag: 6547.15 .

The original and the remarked data are tested on the hydrodam model and SRE model, and the performance is computed. Also, the level of agreement between the original and the remarking efforts is also computed. The results are given in figures 5.26 and 5.27.

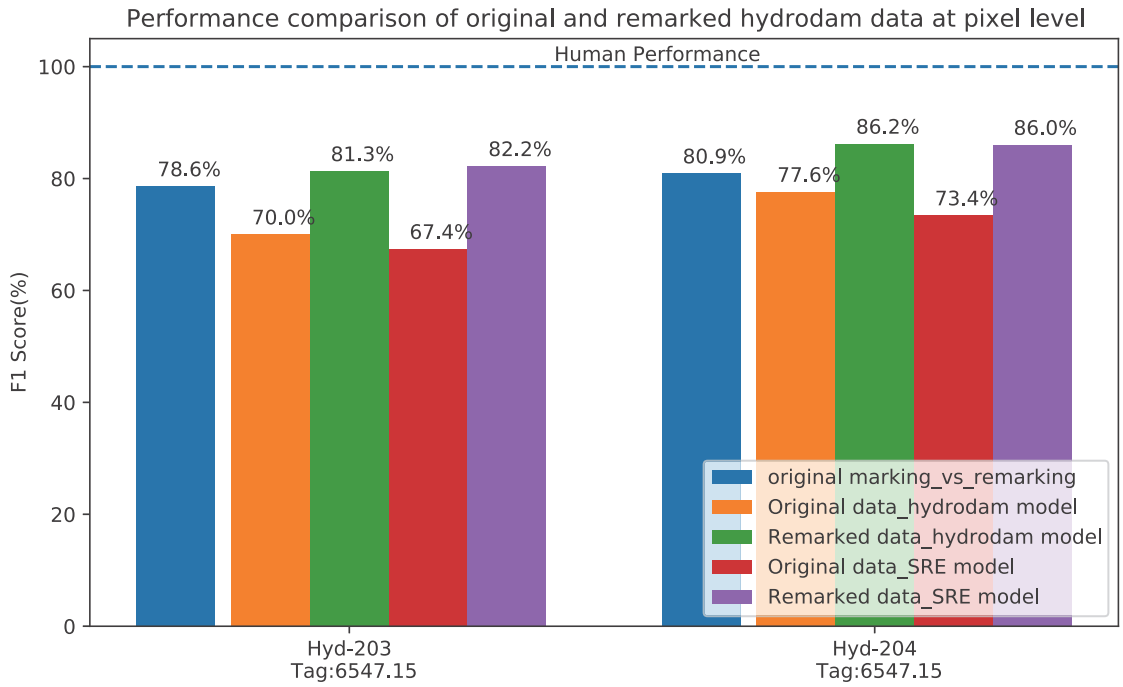


Figure 5.26: Performance comparison of original and remarked data on hydrodam and SRE models at pixel level

The figures 5.26 and 5.27 can be explained below. Bar-1(Blue) indicates the level of original marking compared to the remarking effort. (Only 75-80% of the remarked pings were present in the original data(previous marking). The rest 20-25% of the pings were not marked previously. Bar-2(Orange) indicates the performance of the UNet model(trained on Hydrodam data(Hydrodam Model-2)) on the original data. Bar-3(Green) indicates the performance of the UNet model(trained on Hydrodam data(Hydrodam Model-2)) on the remarked data. Bar-4(Red) indicates the

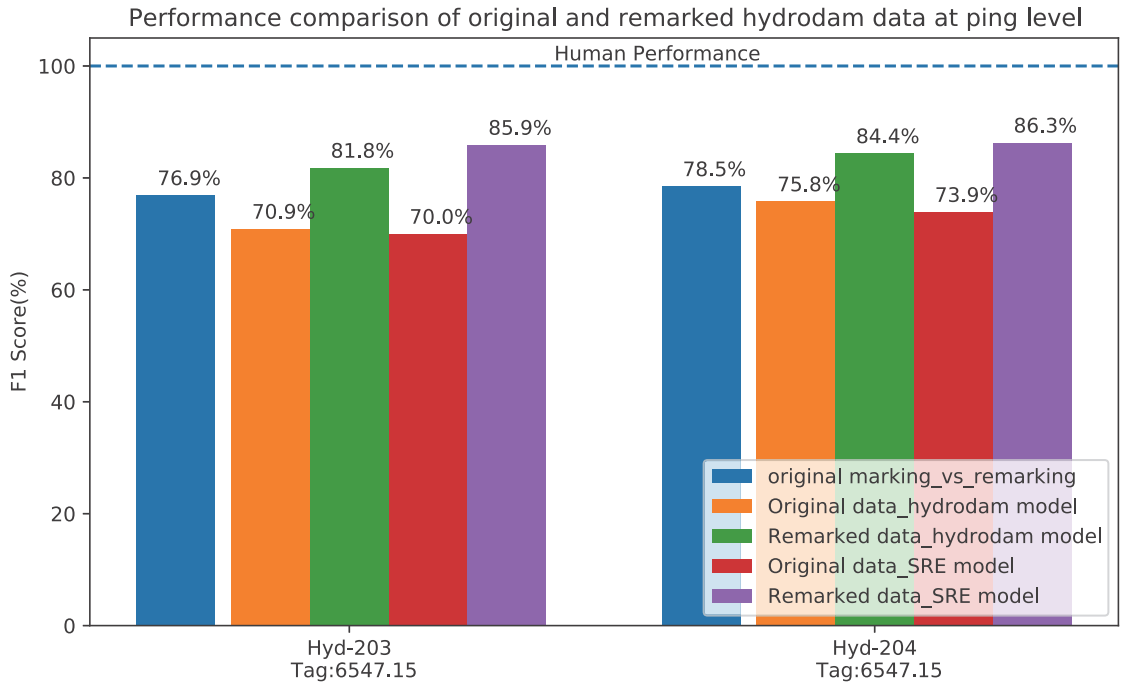


Figure 5.27: Performance comparison of original and remarked data on hydrodam and SRE models at ping level

performance of the UNet model(trained on SRE data(SRE Model-2)) on the original data. Bar-5(Purple) indicates the performance of the UNet model(trained on SRE data(SRE Model-2)) on the remarked data.

Below inferences can be drawn from figures 5.26 and 5.27. Both at pixel level and ping level, the original marking only matches 75 to 80% of the data. In other words, 20-25% of the actual pixels were not marked in the previous annotations. Hydrodam model's performance improved by nearly 10% after remarking the data. The performance of the SRE model on the remarked data improved by nearly 15% than the original data, i.e., the model was performing better than the original annotations. Also, the performance of the SRE model is better than the hydrodam model on remarked data. This is because the data used to train the SRE model is more accurately annotated than the hydrodam data. The remarking of data improved the performance of existing models. Parts of the data which had marking errors can be remarked and the models can be re-trained. Training the model on well annotated and on the data recorded in different environment may reproduce the similar performance metrics as in SRE dataset.

Chapter 6

Conclusion and Future Work

6.1 Discussion

In this section, the research questions that were asked in the Introduction chapter are answered.

1. How are the pings in data represented as an image? We create images by plotting a 2D histogram with pings on the horizontal axis and the remainders obtained by dividing the pings with tag period on the vertical axis. These 2D histograms are used to train neural networks.

2. How are the images classified for the presence of a tag? The number of pings from a chosen tag is considered, and if the number of pings is greater than the desired threshold, the image is assigned a positive label, and if the number of pings is less than the threshold, the image is assigned a negative label. We use 'Xception Neural Network' for this classification task.

3. What are the augmentation techniques implemented in this thesis? We implemented an augmentation technique of adding a random offset to the remainders to shift the pixels vertically upwards in an image making the pings visible clearly in an image. In another augmentation technique, negatively labeled images are generated by tweaking the tag period by 10 milliseconds so that the augmented period is outside the expected deviation of the tag period caused due to the acoustic doppler effect.

4. How is the image segmentation technique used in fish tracking with the images generated from the data? Once the images are generated, these images are segmented using the UNet neural network. UNet segments the image for pixels from the tag.

5. What is the performance of our proposed machine learning method to automark data compared to the manual marking process? As shown in the results chapter, the annotations of automarking in the MarkTags software are less than 50% to the human level annotations, whereas the machine learning approach annotations are 96% accurate and close to the same human level annotations. This shows the usefulness of the novel approach.

6. How can the performance of our proposed method improve? As shown in the results, training the model with a large subset(data containing more pings) of data results in improved model performance. Also, to overcome the model's poor performance in the noisier environment, training the model with noisier and well-annotated data may improve the model's performance.

6.2 Conclusion

For marine life researchers, the concept of tracking and monitoring the fish is important to know where they are as fish tracking has many advantages, for example, to know the behaviour of fish and also it informs about the fisheries operations and marine conservation efforts. Depending on the type of data involved, the methods used in various fish tracking techniques differ. To track the fish, besides using other types of tags, Innovasea uses a new technology in which high-frequency tags are used with a coding scheme called pulse rate encoding scheme to identify the tags uniquely. In this scheme, the tags are identified by the separation between the pulses being transmitted. These high-frequency tags are smaller in size, making the tags suitable for smaller fish and also produce precise trajectories than the old tags, which are larger in size.

To track the high-frequency fish tags, which use a pulse rate encoding scheme, Innovasea uses a visual analytics system called 'MarkTags' where the marking is done automatically, or the user manually marks the pings from the fish tag, which is time-consuming. This is a problem as it prevents scaling up the approach to deal with larger volumes of data. In this thesis, we discussed our proposed method of using the acoustic time series periodic data and mark the pings from the tags that eliminate human involvement to a great extent. Our proposed method of marking

the pings from acoustic period data to track fish is a novel image approach as it uses an image segmentation technique to track the fish from acoustic time series data. From the pings, also known as peak locations, we create an image using the 2D histogram approach. Then, we segmented the images for the pings from the tag using a UNet neural network. The segmentation by UNet identifies the pixels associated with the pings in the input image. The segmented pixels are inversely mapped to pings from the tag by converting the image into the pairs of pings and residuals (obtained by dividing pings with tag period) and taking only the pairs which UNet segmented. The uniqueness of our method is that we use deep learning techniques to find the pings from the tag by generating images with the pings. The image augmentation techniques we developed are also new which helped in improving the model's performance.

When the trained UNet was asked to identify the tags not seen during training, it was able to do so with an accuracy of 95% which is close to human-level annotations. These results show that the deep learning can, if not entirely eliminate, drastically reduce the need for manual analysis of the data generated by Innovasea's new encoding scheme. The results also show that the weaker predictions are due to poor marking of the data or noisy data. The existing model has shown close to human level annotations in the study that involves identifying the pings. However, the existing model is not sufficiently accurate in the fish movement reconstruction study which requires better handling of tail or faint tracks in an image. These limitations of the model can be handled in the future to improve and optimize the UNet model so that it can be integrated into Innovasea's operations.

6.3 Future Work

Our work is the first approach in which the semantic image segmentation technique is applied on acoustic time series data for fish tracking to the best of our knowledge. Therefore, many aspects of our approach can be explored in the future or implemented for similar tasks. Specific tasks that can be done in future works for our method are as follows:

- Our work requires prior knowledge of the tag period to create images. However, in the future, a model that does not require the information of the tag period can

be developed for some applications, like detecting a tagged fish that has traveled long distances. A possible solution would be to compute the autocorrelation of the raw ping time series to detect acoustic tags with unknown periods; however, this technique is unlikely to work if the signal-to-noise ratio is poor or the tag suffers frequent and large Doppler shifts. Some of these issues could be alleviated by a double-pass approach that combines autocorrelation with the UNet.

- The current approach marks the pings from only a single hydrophone-tag combination at a time. An approach that can mark the pings from multiple tags and receivers can be developed. One approach to do this is to mark the pings of each hydrophone-tag combination at a time and combine the results of multiple hydrophone-tag combinations. It is a matter of parallel computing required to the task efficiently.

- Based on the task and type of data, the architectures of the Xception neural network and UNet can be modified to improve the overall performance. For example, for an input image with multiple channels, the network's architecture should be modified to support the input data.

- For the image segmentation task, a different version of CNN can be tested and see if the model performs better than the UNet model used in our project. Other advanced neural networks like R-CNN, UNet++, known for image segmentation, can be tested as these architectures can provide enhanced feature maps that may more accurately detect pings in our data.

- A more robust model trained on data recorded in different conditions like different water bodies and different background noise levels can be developed. Our approach now has models trained on each of the datasets, but to produce a model that can work on all conditions, a single model can be trained on the data recorded in all conditions.

- To detect the pings in a noisy image, as future work, two input images, one image with the original data and the same image with a subcode filter applied, can be added to the training set of UNet. Applying the subcode filter may

remove the noise from the image, and only the pings with their pairs at the subcode distance will be displayed.

Overall, our machine learning solution produced better results than the existing automarking approach of marking the data in MarkTags. If the few edge cases like identifying the pings in faint tracks and noisy images with the existing model are handled, our approach can be integrated into Innovasea's operations.

Bibliography

[1] L. Keeling and H. Gonyou. *The social behaviour of fish," in Social Behaviour in Farm Animals*. CABI Publishing, 2001.

[2] Fish habitat. `https://www.biologyonline.com/dictionary/fish-habitat`, Feb 2021.

[3] Margarida Barcelo-Serra, Sebastià Cabanellas, M. Palmer, M. Bolgan, and J. Alós. A state-space model to derive motorboat noise effects on fish movement from acoustic tracking data. *Scientific Reports*, 11, 2021.

[4] G. La Manna, M. Manghi, F. Perretti, and G. Sarà. Behavioral response of brown meagre (sciaena umbra) to boat noise. *Marine pollution bulletin*, 110 1:324–334, 2016.

[5] Harcourt et.al. Animal-Borne Telemetry: An Integral Component of the Ocean Observing Toolkit. *Frontiers in Marine Science*, 6(326):1–21, June 2019.

[6] V Thorsteinsson. Tagging methods for stock assessment and research in fisheries. *Report*, Report of(FAIR CT.96.1394 (CATAG)):183, 2002.

[7] Mark Johnson, N.A. de Soto, and P.T. Madsen. Studying the behaviour and sensory ecology of marine mammals using acoustic recording tags: a review. *Marine Ecology Progress Series*, 395:55–73, December 2009.

[8] Midwood J. D. Thiem J. D. Klimley P. Lucas M. C. Thorstad E. B. ... Ebner B. C. Cooke, S. J. Tracking animals in freshwater with electronic tags: past, present and future. animal biotelemetry. *Animal Biotelemetry*, 2009.

[9] D.M Heupel, M.R.; Webber. Trends in acoustic tracking: Where are the fish going and how will we follow them? *Am. Fish. Soc. Symp.*, 76:219–231, 2012.

[10] Greg DeCelles and Doug Zemeckis. Acoustic and Radio Telemetry. In *Stock Identification Methods: Applications in Fishery Science: Second Edition*, pages 397–428. Elsevier Inc., 2013.

[11] O. Ronneberger, P.Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. (available on arXiv:1505.04597 [cs.CV]).

[12] R. L. Merrick and T. R. Loughlin. Foraging behavior of adult female and young-of-the-year Steller sea lions in Alaskan waters. *Canadian Journal of Zoology*, 75(5):776–786, 1997.

[13] Eva B. Thorstad, Audun H. Rikardsen, Ahmet Alp, and Finn Okland. The use of electronic tags in fish research – an overview of fish telemetry methods. *Turkish Journal of Fisheries and Aquatic Sciences*, 13(5), 2014.

[14] M.C. Lucas and E. Baras. Methods for studying spatial behaviour of freshwater fishes in the natural environment. *Fish and Fisheries*, 1(4):283–316, 2000.

[15] S.J Cooke and Thorstad. Is radio telemetry getting washed downstream? The changing role of radio telemetry in studies of freshwater fish relative to other tagging and telemetry technology. *American Fisheries Society Symposium*, 76:349–369, 2012.

[16] D.G. Pincock and S.V Johnston. Acoustic telemetry overview. In Adams, N.S., Beeman, J.W. and Eiler, J.H. (eds.). Telemetry techniques: A users guide for fisheries research. *American Fisheries Society, Bethesda, Maryland*, 2012.

[17] John E. Ehrenberg and Tracey W. Steig. A study of the relationship between tag-signal characteristics and achievable performances in acoustic fish-tag studies. *ICES Journal of Marine Science*, 66(6):1278–1283, 2009.

[18] Hartmann. K. Jumppanen. P. Cooper S.P. Bradford R. Hartog. J., Patterson. T. Developing Integrated Database Systems for the Management of Electronic Tagging Data. 2009.

[19] Thomas Grothues. *A Review of Acoustic Telemetry Technology and a Perspective on its Diversification Relative to Coastal Tracking Arrays*, volume 9, pages 77–90. 2009.

[20] Johan Leander, Jonatan Klaminder, Micael Jonsson, Tomas Brodin, Kjell Leonardsson, and Gustav Hellström. The old and the new: Evaluating performance of acoustic telemetry systems in tracking migrating Atlantic salmon (Salmo salar) smolt and European eel (Anguilla anguilla) around hydropower facilities. *Canadian Journal of Fisheries and Aquatic Sciences*, 77(1):177–187, 2020.

[21] Jacob R. Krause, Joseph E. Hightower, Jeffrey A. Buckel, Jason T. Turnure, Thomas M. Grothues, John P. Manderson, John E. Rosendale, and Jeffrey P. Pessutti. Using Acoustic Telemetry to Estimate Weakfish Survival Rates along the U.S. East Coast. *Marine and Coastal Fisheries*, 12(5):241–257, 2020.

[22] Barbara A. Block, Rebecca Whitlock, Robert J. Schallert, Steve Wilson, Michael J.W. Stokesbury, Mike Castleton, and Andre Boustany. Estimating Natural Mortality of Atlantic Bluefin Tuna Using Acoustic Telemetry. *Scientific Reports*, 9(1):1–14, 2019.

[23] M Føre, E Svendsen, J A Alfredsen, I Uglem, N Bloecher, H Sveier, L M Sunde, and K Frank. Using acoustic telemetry to monitor the effects of crowding and delousing procedures on farmed Atlantic salmon (Salmo salar). *Aquaculture*, 495:757–765, 2018.

[24] Natalie V. Klinard, Jordan K. Matley, Silviya V. Ivanova, Sarah M. Larocque, Aaron T. Fisk, and Timothy B. Johnson. Application of machine learning to identify predators of stocked fish in Lake Ontario: using acoustic telemetry predation tags to inform management. *Journal of Fish Biology*, 98(1):237–250, 2021.

[25] Leo Breiman Statistics and Leo Breiman. Random forests. In *Machine Learning*, pages 5–32, 2001.

[26] Innovasea. Private communication . 2020.

[27] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[28] Ross Girshick. Fast r-cnn. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[30] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[31] Antoni Burguera and Francisco Bonin-Font. On-Line multi-class segmentation of side-scan sonar imagery using an autonomous underwater vehicle. *Journal of Marine Science and Engineering*, 8(8), 2020.

[32] Satyanarayana Yegireddi and Nitheesh Thomas. Segmentation and classification of shallow subbottom acoustic data, using image processing and neural networks. *Marine Geophysical Research*, 35(2):149–156, 2014.

[33] S. V. Il'in and M. N. Rychagov. Segmentation of acoustic images by neural network processing. *Akusticheskij Zhurnal*, 50(5):619–627, 2004.

[34] Tony Burnett. *The Doppler effect*. Zenith Pub., 2005.

[35] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, Jan 2015.

[36] Henry J Kelley. Gradient theory of optimal flight paths. *Ars Journal*, 30(10):947–954, 1960.

[37] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[38] François Chollet. Xception: Deep learning with depthwise separable convolutions. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:1800–1807, 2017.

[39] Keiron O'Shea and Ryan Nash. An introduction to convolutional neural networks. *CoRR*, abs/1511.08458, 2015.

[40] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6, 2017.

[41] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem:770–778, 2016.

[42] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[43] Xiaomeng Li, Hao Chen, Xiaojuan Qi, Qi Dou, Chi Wing Fu, and Pheng Ann Heng. H-DenseUNet: Hybrid Densely Connected UNet for Liver and Tumor Segmentation from CT Volumes. *IEEE Transactions on Medical Imaging*, 37(12):2663–2674, 2018.

[44] Le Anh Tran and My Ha Le. Robust u-net-based road lane markings detection for autonomous driving. *Proceedings of 2019 International Conference on System Science and Engineering, ICSSE 2019*, pages 62–66, 2019.

[45] Joe McGlinchy, Brian Johnson, Brian Muller, Maxwell Joseph, and Jeremy Diaz. Application of UNet Fully Convolutional Neural Network to Impervious Surface Segmentation in Urban Environment from High Resolution Satellite Imagery. pages 3915–3918, 2019.

[46] Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3642–3649, 2012.

[47] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *ArXiv*, abs/1603.07285, 2016.

[48] Tom Fawcett. An introduction to roc analysis. *Pattern Recognit. Lett.*, 27:861–874, 2006.

[49] A. Kent, M. M. Berry, F. Luehrs, and J. W. Perry. Machine literature searching viii. operational criteria for designing information retrieval systems. *American Documentation*, 6:93–101, 1955.

[50] P. Jaccard. The distribution of the flora in the alpine zone.1. *New Phytologist*, 11:37–50.

[51] D. M. W. Powers. Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2(1):37–63, 2011.

[52] Christopher K. I. Williams. The effect of class imbalance on precision-recall curves. *Neural Computation*, pages 1–5, 2020.

[53] Zhaobin Wang, E. Wang, and Ying Zhu. Image segmentation evaluation: a survey of methods. *Artificial Intelligence Review*, pages 1 – 38, 2020.