

INTELLIGENT OBJECT RECOGNITION IN UNDERWATER IMAGES USING IMAGE PROCESSING

Submitted in partial fulfilment of requirements for the award of
Bachelor of Engineering in
Electronics and Communication Engineering

by

KUDAPA SUCHARITHA (37130212)
MUNGAMURU SAI DHANA VARSHINI(37130262)



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

SCHOOL OF ELECTRICAL AND ELECTRONICS

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119**

APRIL – 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **KUDAPA SUCHARITHA (37130212)** and **MUNGAMURU SAI DHANA VARSHINI (37130262)** who carried out the project entitled **"Intelligent Object Recognition in Underwater Images using Image Processing"** under our supervision from December 2020 to March 2021.

INTERNAL GUIDE

Dr.T.VINO, M.E., Ph.D.

**Head of the Department
(DR. T. RAVI)**

Submitted for Viva voce Examination held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

We, KUDAPA SUCHARITHA (37130212) **and** MUNGAMURU SAI DHANAVARSHINI (37130262) hereby declare that the Project Report entitled **“Intelligent ObjectRecognition in Underwater Images using Image Processing ”** done by us under the guidance of **Dr.T.Vino** is submitted in partial fulfillment of the requirements fortheaward of Bachelor of Engineering degree in **Electronics and Communication Engineering**.

DATE:

PLACE:

SIGNATURE OF CANDIDATE

- 1.
- 2.

ACKNOWLEDGEMENT

We are pleased to acknowledge my sincere thanks to Board of Management of SATHYABAMA for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

We convey our thanks to **Dr. N. Nandhitha M.E.,Ph.D., Prof. & Dean, School of Electrical and Electronics Engineering** and **Dr. T. Ravi M.E.,Ph.D., Head of the Department, Dept. of ELECTRONICS AND COMMUNICATION ENGINEERING** for providing me necessary support and details at the right time during the progressive reviews.

We would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.T.Vino M.E.,Ph.D., Associate Professor, Dept of ELECTRONICS AND COMMUNICATION ENGINEERING** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

We wish to express my thanks to all Teaching and Non-teaching staff members of the Department of **Electronics and Communication Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Now-a-days automatic detection of aquatic species and trash is very important. To recognize underwater objects like live fish, vast amount of data is required which makes the task much more challenging. Plastic is being thrown into seas, oceans where it pollutes the water, so this should be detected to take cleaning steps. Fish detection and counting is emerging application for underwater fish-based applications. This project describes about recognition of underwater fishes and count of fishes in a given image as well as video. Images are converted into gray scale and applied edge detection technique to find fishes. The recognized fishes are applied with bounding boxes and count is obtained. Similar application can be used for recognition of water bottle. We applied image morphology technique for recognition of objects (fish or water bottle) correctly.

TABLE OF CONTENT

CHAPTER NO	TITLE	PAGE NO
		1
1	Introduction	
	1.1. Background Subtraction	4
	1.2. Fish Detection	5
2	Literature Survey	9
	2.1. Detecting, tracking and counting fish in low quality unconstrained underwater videos.	9
	2.2. Congruence between automatic fish counter Data and rod catches of atlantic salmon	9
	2.3 Development of Hydroacoustic Methods for Fish Detection in Shallow Water.	11
	2.4 A system for video surveillance and monitoring.	12
	2.5 Airborne traffic surveillance systems: video surveillance of highway traffic.	14
	2.6 Automated detection and recognition of wildlife using thermal cameras.	14
	2.7 Moving object detection and tracking in video surveillance system.	16
	2.8 Moving Object Detection Using Background Subtraction	17

3	Proposed System	18
	3.1. Underwater Images	18
	3.2. Gray Scale Conversion	19
	3.3. Edge Detection	19
	3.4. Morphological Operations	19
	3.5. Applying Bounding Box to Image	20
4	Result Analysis	21
5	Software Environment	25
	5.1. Introduction	25
	5.2. Applications of MATLAB	25
	5.3. MATLAB	25
	5.4. Basic Building Blocks of MATLAB	26
	5.4.1. MATLAB Window	26
	5.4.2. Command Window	26
	5.4.3. Work Space Window	27
	5.4.4. Three Current Directory Window	27
	5.4.5. Four Command History Window	27
	5.4.6. Editor Window	27
	5.4.7. Graphics or Figure Window	28
	5.4.8. Online Help Window	28
	5.5. MATLAB Files	28
	5.6. MAT-Files	29
	5.7. The MATLAB System	29

	5.8. Development Environment	29
	5.9. MATLAB Mathematical Function	29
	5.10. MATLAB Language	29
	5.11. Four Graphics	29
	5.12. MATLAB Application Program Interface(API)	29
	5.13. Some Basic Plot Commands	30
	5.14. Algebraic Operations In MATLAB	31
	5.15. MATLAB Working Environment	32
	5.14.1. MATLAB Desktop	32
	5.14.2. MATAB Editor to Create M-Files	33
	5.16. Three Getting Help	33
6	Digital Image Processing	34
	6.1. Introduction	34
	6.2. Types of Images	35
	6.3. Coordinate Convention	35
	6.4. Images as Matrices	36
	6.5. Reading Images	37
	6.6. Displaying Images	39
	6.7. Writing Images	41
7	Conclusion	47
8	Future Work	48
9	References	49

LIST OF FIGURES

FIG NO	TITLE	PAGE NO
1.1	Example Images	2
1.2	Block Diagram	4
1.3	BS process	5
1.4	Example images	6
1.5	Tracking example with two frames grabbed at two consecutive times	7
1.6	Difficult cases for fish detection.	8
3.1	Block Diagram for fish detection and counting from Image	18
4.1	Fish Detection from underwater images	21
4.2	Detection and counting Number of Fishes (Applied Bounding Box)	22
4.3	GUI selected input video	22
4.4	Detected Fishes from underwater video	23
4.5	Underwater water bottle Image	23
4.6	Detected water bottle from underwater image	24

CHAPTER 1

INTRODUCTION

Counting and tracking fish populations is important for conservation purposes as well as for the fishing industry. Usually, biologists determine the situation under the water by casting nets for collecting and examining fish or by human underwater observation [1]. Such methods are not able to provide a comprehensive observation, cannot capture the real fish state and behavior, and require much time and expenses. In addition, there is a big risk to kill or damage fish and their habitat during the collection process. These drawbacks force researchers to develop alternative ways of marine life observation [2]. One of the most commonly used approaches for underwater observation is to collect videos and analyse them to solve different tasks of marine life monitoring, for instance, fish counting.

Many systems for underwater fish observation located in rivers and fish farms involve the manual counting of the passing fish, while the automation can speed up the process significantly, reduce the costs associated with the involvement of human experts and provide an opportunity for fish recognition. Various automatic fish counters already exist, such as resistive counters that rely on the fact that the resistivity of a fish is lower than that of water [3], optical counters that use light beams [3], and hydro acoustic counters that operate using the principles of sonar [4]. Such approaches, based on different physical principles, cope with the task of counting passing objects quite successfully. However, such methods typically cannot make distinction between fish and other passing objects, and moreover, cannot recognize different species. Computer vision techniques provide an attractive alternative for building a more robust and versatile fish counting systems.



Fig 1.1 Example images: (a) High illumination; (b) Low illumination.

Barely any countries are concentrating on developing advances. The greater part of the advances is centered around complex life for people. Be that as it may, the impact of innovation is poor. Because of the absence of innovative ability, the fishing market is reliant on past based information or chance which brings about disappointments or not suitable yield. Analysts additionally need to go through more cash to consider the natural examples. A definitive point of the task is to recognize the fishes submerged and attempt to group them. By the distinguishing proof of the various species, we can consider the diverse way of life example of fishes and how It influences the earth and help in business angling. For the AI process, a profound learning technique is utilized.

Underwater video is used for studying marine and riverine ecosystems, mainly for observing animal abundance and behavior in the context of environmental monitoring. Optical video provides detailed information in a form that is easy to interpret because humans naturally interpret their world visually. But human analysis of video takes time, so some level of automation is needed to process large video data sets in a timely, efficient manner in order to provide information needed for permitting and adaptive management decisions. Computer vision and machine learning have proven effective for automating similar monitoring applications, such as video surveillance. Underwater imagery presents unique challenges that include sudden illumination changes, unequal spectral propagation (i.e., color content is affected by distance), low contrast, clutter in the form of floating vegetation, and changes in visibility due to turbidity.

Object Detection Methods Video surveillance systems are currently widely used for analysing video sequences and applied, for example, in security control [5], traffic

monitoring [6], anomaly detection [7] and animal monitoring and conservation [8]. Some systems include human intervention, but involving of a human operator may slow down the process of video analysis significantly and make the task impracticable.

Therefore, the creation of autonomous video surveillance systems is one of the most actively developing areas of computer vision. The term autonomous in this case implies that the system receives a video as an input and performs both the low-level tasks, like motion detection, and high-level decisions like object recognition, without human expert involvement. [9] Regardless the type of high-level task, the crucial step which must be done for further analysis is the moving object detection. It separates moving objects, which form the foreground, from the stationary background. Different approaches for the moving object detection from videos can be classified in the following way [10]:

- **Background subtraction (BS)** is commonly used in static scenes. It uses a modelled background image and subtract the current image pixel-by-pixel from it to detect regions of motion. With each new image a background model is updated to adapt to scene changes.
- **Temporal differencing** is usually applied in cases where the camera is moving and the background is not stable. Moving regions are detected by taking pixel-by-pixel difference of consecutive frames in a video sequence.
- **Statistical approaches** are a modification of BS methods, that use statistical information of each pixel and compare pixels' statistics to identify foreground.
- **Optical flow** can be used with the moving camera and moving background, but are computationally complex and time consuming. The basic idea is to compute the apparent velocity and direction of every pixel to find moving objects.

Foreground pixels from the background by using the defined background model. During the following step pixel level processing, including, for example, morphological operations and low pass filter, is performed to remove the noise. Then the foreground pixels are grouped into connected regions and the bounded boxes of the regions are calculated. Finally, after a set of individual regions is obtained, all false regions, which do not correspond to objects, are removed based on, for example, size.

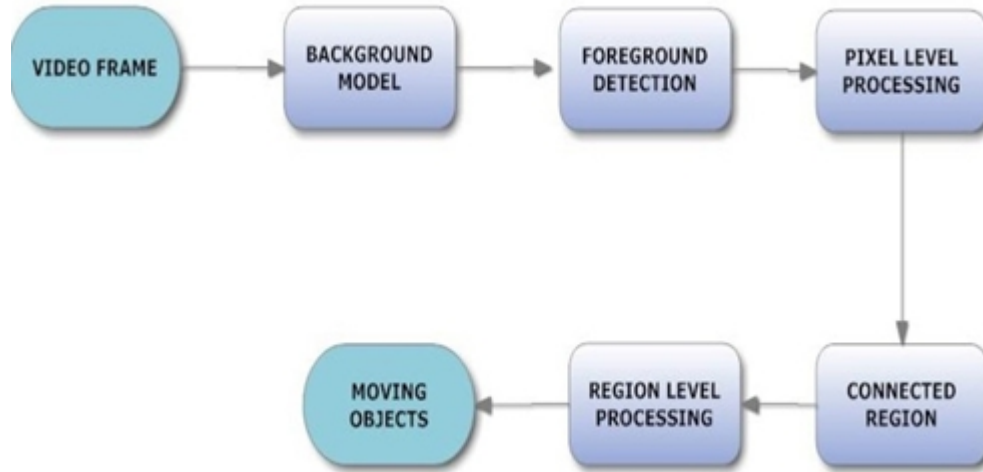


Fig 1.2. Block Diagram

1.1 Background Subtraction

BS methods are widely used for solving the task of moving object detection. These algorithms are relatively fast, which makes it possible to detect objects in real time. Moreover, they do not require a lot of memory and show robust results. BS approach is mostly applicable when both the camera and the background are stable, but can be easily modified for the cases with moving camera or background [12,13]. Due to its benefits, BS methods have gained popularity among researches, which led to the emergence of a large number of different algorithms [14–16].

Most of the BS methods are implemented by the same scheme, which consists of three main steps (Figure 3):

- **Background modeling**

This is the first step of every BS algorithm. For detecting foreground objects, a background image that does not contain any moving objects needs to be estimated. For that purpose, a background model is built using a several number of successive frames.

- **Foreground detection.** After the background image is obtained, all subsequent frames are extracted from that model in order to find pixels that do not belong to the background and therefore can be considered as a foreground.

- **Background update.** During the detection process a background model must be updated with each new frame, because in real cases a background does not remain still throughout the whole video. The model must be adaptive for various changes (e.g. some moving object does not move for a long time) so that they do not affect the result of detection.

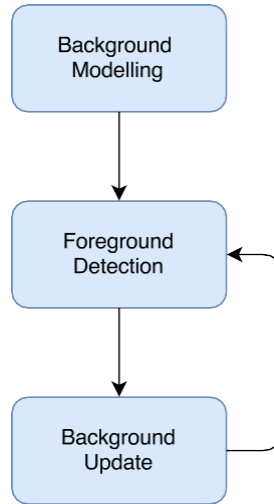


Fig 1.3. BS process

Different BS methods are usually classified by the way they initialize a background model and how they compute the distance between background and foreground pixels [17, 18]. In accordance with this type of classification, the most commonly used BS techniques are the following:

- **Basic Motion Detection.** A background is represented by a single image taken when there is no motion or estimated by a use of temporal median filter [19, 20]. Difference between foreground and background pixels is calculated on the basis of pixels' colors.
- **One Gaussian.** The idea of this approach is that background pixels are modelled with a probability density function (PDF) and the distribution is considered to be normal [21]. The parameters of PDF are estimated with a number of training frames. In this case, a pixel is assumed to be a foreground if it has low probability.
- **Gaussian Mixture Model (GMM).** In contrast to the previous method, GMM based methods make use of a several Gaussians in modeling process [22].
- **Kernel Density Estimation (KDE).** KDE is a non-parametric statistical approach, in which the PDF is estimated directly from the data [23].
- **Other methods.** There is a vast variety of different techniques that are based on, for example, fuzzy logic [24], neural networks [25], codebook [26] and eigenvalues and eigenvectors [27].

1.2 Fish Detection

Object detection algorithms have various applications. For instance, computer vision techniques provide a great opportunity to make animal monitoring more accurate, less time consuming and fully automated.

In particular, different approaches and methods of moving object detection have been used for the task of fish detection, which is the fundamental step in building a fish observation system. Palazzo et al. [28] proposed an approach for object detection that imply explicit modelling both the background and the foreground for each frame. This allows to avoid misdetections when the background is not stationary or when target objects have the same color as the background. The latter problem is also solved by introducing texture information in models as well as the color. The algorithm was evaluated on real underwater videos and showed high and stable performance.

In [29], an Expectation-maximization (EM) algorithm was proposed for fish detection. Given intensity images of fish (Figure 4), a certain threshold is selected, such that pixels with higher intensities are assumed to belong to the fish. Then the sets of x and y locations corresponding to these pixels are defined (Figure 5). The shape of each fish is assumed to be a multivariate Gaussian, and then an image is modeled as GMM.

And finally, the parameters of GMM, including the number of fish, are estimated using an EM algorithm. The method was tested on southern bluefin tuna fish, an individual fish was modelled by using a two-dimensional Gaussian distribution with x and y pixel locations as an input. Spampinato et al. [2] presented an automated video processing system for underwater video surveillance. The system is able to solve the tasks of texture and color analysis,



Fig 1.4 Example images: (a) Binary Image; (b) Grey Image.

fish detection and fish tracking by using three image processing subsystems. The first one calculates the average texture and color properties of the frame. It considers the brightness, smoothness and identifies the green tone. Fish detection subsystem is based on a BS approach. It uses the combination of moving average algorithm with the Adaptive

GMM [30] which shows high performance on both stable and dynamic background scenes. The last subsystem tracks the fish throughout the video. The proposed system was tested on 20 underwater video sequences and the results were considered to be excellent when compared to other similar methods. An example of the system outcome is shown in Figure 6.

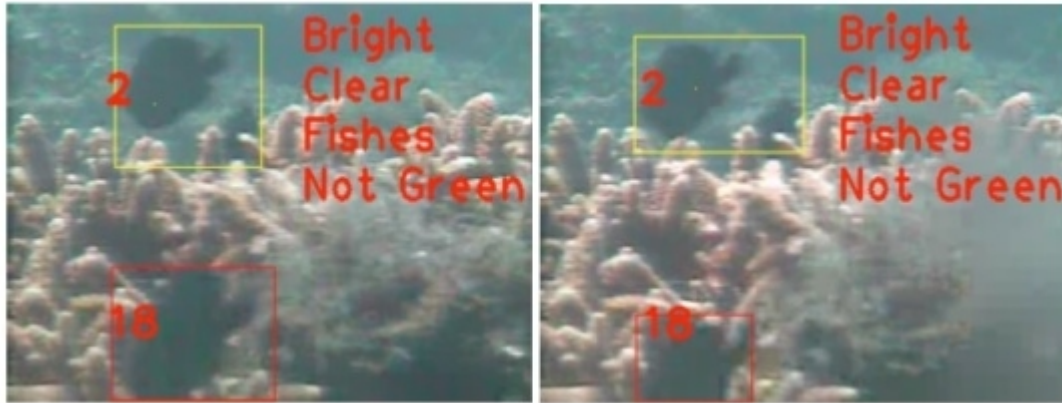


Fig 1.5 Tracking example with two frames grabbed at two consecutive times [2].

Challenges in Fish Detection Despite the variety of the proposed approaches, it is a challenging task to find a robust and high performance method for fish detection. Usually video sequences may contain various attributes that make the task more difficult. Some of the possible problems connected with the detection are listed below [10]:

- **Illumination changes.** During the video capturing the illumination may change gradually (e.g. intensity of the sun changes throughout the day) or suddenly (e.g. the lamp has been turned off), so the background model must take the light into account.
- **Dynamic background.** Often the background is not a stable scene and the algorithm should distinguish moving objects of interest from those that can be regarded as background.
- **Occlusion.** Object of interest cannot be fully seen and detected on some frames if it is occluded by other objects.
- **Clutter.** It hinders the process of background modelling and complicates the segmentation.
- **Camouflage.** Target objects may have the similar color and pattern as a background, which makes them hardly distinguished even by the human.
- **Camera motions.** If the camera is unstable it may affect the quality of the captured video.
- **Noise.** Video may contain noise which also affects the results.
- **Environmental issues.** The quality of water, underwater garbage, presence of moving plants, etc. may also hamper the fish detection.

In the case of fish detection, different types of videos are used for experiments (Figure 7). In low quality video, there is noise and fish do not differ sufficiently from the background in the terms of color. High quality video on the opposite allows all fish to be distinguished easily, but that may cause clutter and occlusions. Illumination changes and sudden appearance of glares are common attributes of underwater videos as well. This kind of troubles should be considered while choosing an appropriate detection method



Fig 1.6 Difficult cases for fish detection.

CHAPTER 2

LITERATURE SURVEY

2.1 Detecting, tracking and counting fish in low quality unconstrained underwater videos.

Concetto Spampinato, Yun-Heh Chen-Burger, Gayathri Nadarajan, and Robert B Fisher. Detecting, tracking and counting fish in low quality unconstrained underwater videos. In Proceedings of the Third International Conference on Computer Vision Theory and Applications, volume 2, pages 514–519. Citeseer, 2008.

In this work a machine vision system capable of analysing underwater videos for detecting, tracking and counting fish is presented. The real-time videos, collected near the Ken-Ding sub-tropical coral reef waters are managed by EcoGrid, Taiwan and are barely analysed by marine biologists. The video processing system consists of three subsystems: the video texture analysis, fish detection and tracking modules. Fish detection is based on two algorithms computed independently, whose results are combined in order to obtain a more accurate outcome. The tracking was carried out by the application of the CamShift algorithm that enables the tracking of objects whose numbers may vary over time. Unlike existing fish-counting methods, our approach provides a reliable method in which the fish number is computed in unconstrained environments and under several scenarios (murky water, algae on camera lens, moving plants, low contrast, etc.). The proposed approach was tested with 20 underwater videos (More)

2.2 Congruence between automatic fish counter data and rod catches of atlantic salmon

Thorley, DMR Eatherley, AB Stephen, I Simpson, JC MacLean, and AF Youngson. Congruence between automatic fish counter data and rod catches of atlantic salmon (salmosalar) in scottish rivers. ICES Journal of Marine Science: Journal du Conseil, 62(4):808–817, 2005.

We used over 154 000 scale samples collected from salmon fisheries in the large River Teno system over a 40-year period to quantify life history diversity and long-term trends. We identified 120 different life history strategies, including combinations of smolt (2–8) and sea ages (1–5) and previous spawning events. Most strategies were rare; 60% of individuals matured after 1 year at sea following 3–5 years in fresh water. Age at maturity changed with an increase in two-sea-winter salmon and previous spawners and a decline in

three-sea-winter fish. Smolt age distribution showed a decreasing proportion of age-3 smolts, while that of age-5 smolts increased. Fishing gear and fishing season times selected for fish differing in life history strategies. Temporal variation in life histories reflected changes in both fisheries and the changing environment. There was an inverse relationship between years spent in fresh water and sea age. Bio complexity was manifested by the multiple year classes (6–11) present in annual runs, which increased with years, reflecting an increase both in previous spawners and sampling effort. The high number of cohorts spawning simultaneously each year indicates strong generational overlap, which has been suggested to maintain genetic diversity and thereby resilience via the portfolio effect.

Bio-complexity in fish populations, i.e. their diversity and variation in life-history and genetic characteristics, provides resilience to disturbances by spreading risk temporally and spatially and thus contributes to long-term sustainability (Stearns 1992; Hilborn et al. 2003; Schindler et al. 2010). Sufficient diversity in populations is also a prerequisite for sustainable fisheries (Hilborn et al. 2003; Hutchinson 2008). Typically, diversity comprises variation in the age at maturity. Downloaded from www.nrcresearchpress.com by University of Chicago Library on 04/15/18 For personal use only.

This Just-IN manuscript is the accepted manuscript prior to copy editing and page composition. It may differ from the final official version of record. Individuals from the same cohort reproducing in different years – or iteroparity with the possibility of repeated spawning events for individual fish; both allocating the reproductive effort across multiple years and stabilizing populations under environmental variability (Schindler et al. 2010; Moore et al. 2014). It is therefore important to characterize the levels of, and changes in, diversity in exploited populations so that the potential importance of bio-complexity for sustaining productivity can be assessed. Such data also provide a baseline for assessing the potential effects of climate change. Anadromous salmonid fish exhibit especially wide variation in life histories among fish species, both in time spent in freshwater before the oceanic migration and in duration of the marine feeding migration and connected timing of return migration to freshwater for spawning (e.g. Klemetsen et al. 2003). The simplest set of different salmonid life histories is that of pink salmon (*Oncorhynchus gorbuscha*) and the most complex those of steelhead trout (*O. mykiss*; Thorpe 1998; Moore et al. 2014) and Atlantic salmon (*Salmo salar*; Klemetsen et al. 2003; Niemelä et al. 2006a).

2.3 Development of Hydroacoustic Methods for Fish Detection in Shallow Water.

Helge Balk. Development of Hydroacoustic Methods for Fish Detection in Shallow Water.

The main objective of this thesis was measurement of fish side-aspect target strength, which was determined with simulations and under experimental conditions with both immobilized and swimming fish. In addition, target strength was measured in the natural environment of the fish. Furthermore, horizontal hydroacoustics was utilized to establish the effects of environmental factors on fish migration. Target-strength simulations with a swimbladder model demonstrated that the relationship between target strength and swimbladder length was not linear. Furthermore, with the concave bending of the swimbladder model, the interference between wavelets caused fluctuation in target strength, which could be over 15 dB. A single log-linear regression for target strength as a function of fish length was fitted.

The ex situ measurements of immobilized Atlantic salmon (*Salmo salar* L.) and other species showed that the linear relationship between mean target strength and the logarithm length of fish increases with a slope larger than 20. In addition, whitefish (*Coregonus lavaretus* (L) s.l.) has a target strength that is about 5 dB higher than salmon of the same size. The variation in target strength with the side-aspect angle was very large. The best fit of the all-side-aspect models between mean target strength and side-aspect angle (α) was obtained using the $\cos^3(2\alpha)$ function. A statistical approach was utilized to predict target-strength distributions of salmon as they ascended a river. With a large number of fish, the large value for standard error of the estimate (S.E.E) in a log-linear regression for target strength and fish length smoothed out these distributions. The probability of detection of salmon with threshold target strength (-29 dB) increased as length of the fish increased. Few statistically significant correlations were observed between entry of salmon into the river and six environmental factors measured in the Tornionjoki River. In Äijälänsalmi Channel, migration of cyprinids and percids through the channel was triggered by the temperature difference between lakes, and migration occurred around dawn and dusk.

In fisheries research, one purpose of echo sounding is to estimate the density or number of aquatic organisms, especially fish (MacLennan & Simmonds 1992). The classification and identification of these organisms generally combines knowledge of the distribution and behavior patterns of constituent species with collection and analysis of acoustic and catch data (Horne 2000). In echo-sounding systems, acoustic energy is transmitted in pulses at a particular frequency (usually 38 – 420 kHz) by means of a transducer producing a directional sound beam. When a sound wave encounters a density difference, e.g. a fish, an echo propagates outward from this target back to the receiver.

Echoes returning to the transducer are said to be backscattered sound (Midttun 1984). Accurate interpretation of the acoustic data from fisheries surveys requires an understanding of the scattering properties of fish. Traditionally in fisheries acoustics, the acoustic beam has been aimed vertically.

However, the echo sounder can also be used to transmit the sound beam horizontally across water. During the past decade, a side-looking (horizontal), fixed location split-beam or dual-beam echo sounder has been used for assessment of fish stock in rivers and shallow channels (Kubecka 1996, Ransom et al. 1998). Use of an acoustic system in a river offers the advantage that it is made up of noninvasive and nondestructive devices for counting number of fish and observing the behavior of fish. In a fixed location, horizontal echo sounding has basically four different sources of echoes. These are echoes (a) from passing fish, (b) from drifting debris, (c) from stationary targets like bottom and stones, and (d) from noise or reverberation. Echoes from fish are naturally the main objective in fisheries acoustics; the rest are usually unwanted sources of echoes (Balk 2001). In rivers, the downstream migration of fish may be passive or active, but upstream migration must be active. Therefore, it can be assumed that all targets that are moving upstream are fish. Selection of sampling site is the most important factor for successful hydroacoustic monitoring in rivers.

2.4 A system for video surveillance and monitoring.

Robert T Collins, Alan J Lipton, Takeo Kanade, Hironobu Fujiyoshi, David Duggins, YanghaiTsin, David Tolliver, Nobuyoshi Enomoto, Osamu Hasegawa, Peter Burt, et al. A system for video surveillance and monitoring. 2000.

The thrust of CMU research under the DARPA Video Surveillance and Monitoring (VSAM) project is cooperative multi-sensor surveillance to support battlefield awareness [17]. Under our VSAM Integrated Feasibility Demonstration (IFD) contract, we have developed automated video understanding technology that enables a single human operator to monitor activities over a complex area using a distributed network of active video sensors. The goal is to automatically collect and disseminate real-time information from the battlefield to improve the situational awareness of commanders and staff. Other military and federal law enforcement applications include providing perimeter security for troops, monitoring peace treaties or refugee movements from unmanned air vehicles, providing security for embassies or airports, and staking out suspected drug or terrorist hide-outs by collecting time-stamped pictures of everyone entering and exiting the building. Automated video surveillance is an important research area in the commercial sector as well.

Technology has reached a stage where mounting cameras to capture video imagery is

cheap, but finding available human resources to sit and watch that imagery is expensive. Surveillance cameras are already prevalent in commercial establishments, with camera output being recorded to tapes that are either rewritten periodically or stored in video archives. After a crime occurs – a store is robbed or a car is stolen – investigators can go back after the fact to see what happened, but of course by then it is too late. What is needed is continuous 24-hour monitoring and analysis of video surveillance data to alert security officers to a burglary in progress, or to a suspicious individual loitering in the parking lot, while options are still open for avoiding the crime. Keeping track of people, vehicles, and their interactions in an urban or battlefield environment is a difficult task. The role of VSAM video understanding technology in achieving this goal is to automatically “parse” people and vehicles from raw video, determine their geolocations, and insert them into a dynamic scene visualization. We have developed robust routines for detecting and tracking moving objects.

Detected objects are classified into semantic categories such as human, human group, car, and truck using shape and color analysis, and these labels are used to improve tracking using temporal consistency constraints. Further classification of human activity, such as walking and running, has also been achieved. Geolocations of labeled entities are determined from their image coordinates using either wide-baseline stereo from two or more overlapping camera views, or intersection of viewing rays with a terrain model from monocular views. These computed locations feed into a higher level tracking module that tasks multiple sensors with variable pan, tilt and zoom to cooperatively and continuously track an object through the scene. All resulting object hypotheses from all sensors are transmitted as symbolic data packets back to a central operator control unit, where they are displayed on a graphical user interface to give a broad overview of scene activities. These technologies have been demonstrated through a series of yearly demos, using a testbed system developed on the urban campus of CMU.

2.5 Airborne traffic surveillance systems: video surveillance of highway traffic.

Suman Srinivasan, Haniph Latchman, John Shea, Tan Wong, and Janice McNair. Airborne traffic surveillance systems: video surveillance of highway traffic. In Proceedings of the ACM 2nd International Workshop on Video Surveillance & Sensor Networks, pages 131–135. ACM, 2004.

Timely information about highway traffic conditions is very important for the Department of Transportation (DOT) and other relevant agencies. Such live information would be very important when traffic incidents or accidents occur. An aerial view is the best for traffic situations, particularly over highways. Unmanned aircraft equipped with video

cameras and/or other sensors may be able to deliver the necessary information through video images with relatively low operational costs and risks to human life. ATSS (Airborne Traffic Surveillance System), a project funded by the Florida Department of Transportation, attempts to make this vision a reality. This paper describes how the University of Florida research team implemented a system for ATSS from the ground up, using unmanned aerial vehicles, digital video encoding, and transmission of data and multimedia video streams over FDOT's microwave IP networks.

This paper provided a brief overview of the ATSS project funded by the FDOT and implemented by the UFL research team. Other than providing details on how video surveillance of state highways can be implemented using unmanned aerial vehicles and the existing microwave network, the paper also describes how the UFL research team developed in-house software that enabled the transmission of signal strengths over the network, and implemented automatic switching of highway traffic video using the signal strengths obtained from the video signal. We believe that the ATSS project is a ground-breaking project and could revolutionize highway traffic surveillance in the near future. This would enable the Department of Transportation and other emergency services to respond more quickly when accidents occur and for other operations that require urgent feedback. Some of the documents referenced in this paper are available for download from [17].

2.6 Automated detection and recognition of wildlife using thermal cameras.

Peter Christiansen, Kim Arild Steen, RasmusNyholmJørgensen, and Henrik Karstoft. Automated detection and recognition of wildlife using thermal cameras. Sensors, 14(8):13778–13793, 2014.

In agricultural mowing operations, thousands of animals are injured or killed each year, due to the increased working widths and speeds of agricultural machinery. Detection and recognition of wildlife within the agricultural fields is important to reduce wildlife mortality and, thereby, promote wildlife-friendly farming. The work presented in this paper contributes to the automated detection and classification of animals in thermal imaging. The methods and results are based on top-view images taken manually from a lift to motivate work towards unmanned aerial vehicle-based detection and recognition. Hot objects are detected based on a threshold dynamically adjusted to each frame. For the classification of animals, we propose a novel thermal feature extraction algorithm. For each detected object, a thermal signature is calculated using morphological operations.

The thermal signature describes heat characteristics of objects and is partly invariant to translation, rotation, scale and posture. The discrete cosine transform (DCT) is used to

parameterize the thermal signature and, thereby, calculate a feature vector, which is used for subsequent classification. Using a k-nearest-neighbor (kNN) classifier, animals are discriminated from non-animals with a balanced classification accuracy of 84.7% in an altitude range of 3–10 m and an accuracy of 75.2% for an altitude range of 10–20 m. To incorporate temporal information in the classification, a tracking algorithm is proposed. Using temporal information improves the balanced classification accuracy to 93.3% in an altitude range 3–10 of meters and 77.7% in an altitude range of 10–20 m

We have introduced a method for the automatic detection and recognition of wildlife using thermal cameras for UAV technology. Based on a dynamic threshold, hot objects are detected and subsequent feature extraction is performed. The novel feature extraction method, presented in this paper, consist of an extraction of thermal signatures for each detected object and a parameterization of this based on DCT. Methods for classification using measurements from both single and multiple frames is presented. Combining measurements from multiple frames achieves the best performance, with a balanced classification accuracy of 93.5% in the altitude range of 3–10 m and 77.7% in the altitude range of 10–20 m, thus demonstrating a clear relationship between the performance of detection and classification relative to altitude. The simulated and limited dataset is favorable in terms of performance for the given algorithms. The actual applicability of the system should therefore be determined using footage from an actual UAV. The proposed detection and classification scheme is based on top-view images of wildlife, as seen by a UAV. The use of UAV-technology for automatic detection and recognition of wildlife is currently part of ongoing research towards wildlife-friendly agriculture.

2.7 Moving object detection and tracking in video surveillance system.

Kinjal A Joshi and Darshak G Thakore. A survey on moving object detection and tracking in video surveillance system. International Journal of Soft Computing and Engineering, 2(3):44–48, 2012.

Moving object detection and tracking are the more important and challenging task in video surveillance and computer vision applications. Object detection is the procedure of finding the non-stationary entities in the image sequences. Detection is the first step towards tracking the moving object in the video. Object representation is the next important step to track. Tracking is the method of identifying, the position of the moving object in the video. Identifying the position is much more challenging task then detecting the moving object in a video. Object tracking is applied in numerous applications like in robot vision, monitoring the traffic, Video surveillance, Video in-painting and Simulation. Here we are going to

present a brief review of numerous object detection, object classification and object tracking algorithms available.

Video scrutiny is a more challenging task in today's environment. Tracking the moving object has attracted many researchers in the field of computer vision and image processing. Video surveillance is the process of observing the behaviour, events and other necessary evidence, typically of the people for supervision and protecting them. Surveillance is mainly used by governments for gathering information, for investigating and preventing the crime. Video surveillance activity can be divided into three types namely. Manual video surveillance, Semi-autonomous surveillance and Fully-autonomous surveillance. In manual surveillance human is responsible for analysing the contents of the video. Semi-autonomous surveillance system involves video processing by system with the interaction of the human when necessary. Fully-autonomous surveillance system, the system will perform every task such as detection of motion, tracking etc., without the need of human interaction. Automated monitoring procedure also known as the Intelligence Visual Surveillance (IVS) involves the scrutiny and interpretation of objects activities, in addition to object detection and tracking to recognize the visual actions of the scene. The main task of IVS consists of wide area surveillance control and scene interpretation. Object tracking must deal with several illumination changes and well-known challenges. Mainly video analysis is categorized into three basic phases: moving entity detecting, finding the trajectory of object from one frame to another frame and scrutiny of entity tracks to identify their performance. Tracking objects in a static environment is much easier than tracking objects in dynamic environment. In common, in a dynamic ecological system together background and entity varies [1]. In principle, to resolve this common unconstrained problem is rigid. One can place a group. The rest of the paper is systematized as follows. Section II presents methodology in which object detection, representation and tracking Techniques is illustrated. Performance analysis is described in Section III. Conclusion about the paper is presented in Section IV.

2.8 Moving Object Detection Using Background Subtraction

Soharab Hossain Shaikh, Khalid Saeed, and Nabendu Chaki. Moving Object Detection Using Background Subtraction. Springer, 2014.

Background subtraction is a widely used approach for detecting moving objects from videos captured with static a camera. This chapter introduces the basic concept behind this approach using a simple frame differencing method. A survey on existing literature on this topic is also reported in this chapter.

CHAPTER 3

PROPOSED SYSTEM

The proposed system is to recognize an object in a noisy and dense condition. Live fish and water bottle is recognized and image count is also given. It has a better approach and morphology techniques done in Matlab to give accurate and efficient results.

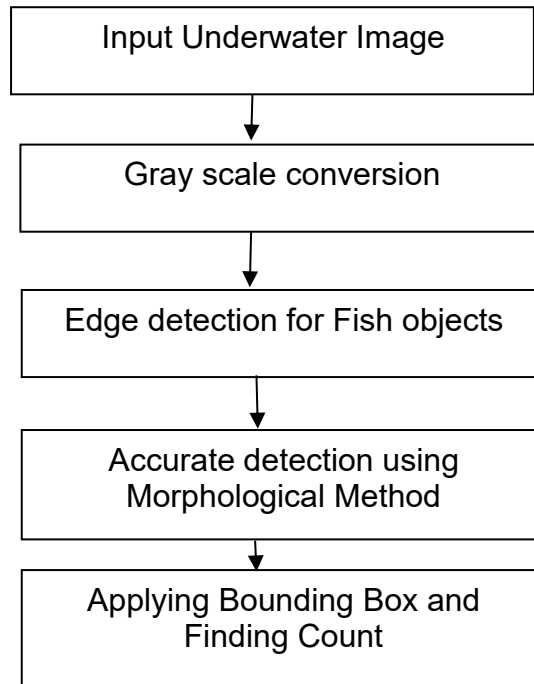


Fig.3.1 Block Diagram for fish detection and counting from Image

Here we detected fish and counted from underwater images. Same procedure can be applied for detection and counting of fishes from video. Video is nothing but frames in sequence.

3.1 Underwater Images

The underwater image processing area has received considerable attention within the last decades, showing important achievements. In this paper we review some of the most recent methods that have been specifically developed for the underwater environment. These techniques are capable of extending the range of underwater imaging, improving image contrast and resolution. After considering the basic physics of the light propagation in the water medium, we focus on the different algorithms available in the literature.

3.2 Gray scale conversion

An RGB image can be viewed as three images(a red scale image, a green scale image and a blue scale image) stacked on top of each other. In MATLAB, an RGB image is basically a $M \times N \times 3$ array of colour pixel, where each colour pixel is a triplet which corresponds to red, blue and green colour component of RGB image at a specified spatial location. Similarly, A Grayscale image can be viewed as a single layered image. In MATLAB, a grayscale image is basically $M \times N$ array whose values have been scaled to represent intensities.

3.3 Edge detection

Edge detection includes a variety of mathematical methods that aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed edges. The same problem of finding discontinuities in one-dimensional signals is known as step detection and the problem of finding signal discontinuities over time is known as change detection. Edge detection is a fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction.

3.4 Morphological Operations

Morphological image processing is a collection of non-linear operations related to the shape or morphology of features in an image. According to Wikipedia, morphological operations rely only on the relative ordering of pixel values, not on their numerical values, and therefore are especially suited to the processing of binary images. Morphological operations can also be applied to greyscale images such that their light transfer functions are unknown and therefore their absolute pixel values are of no or minor interest. Morphological techniques probe an image with a small shape or template called a structuring element. The structuring element is positioned at all possible locations in the image and it is compared with the corresponding neighbourhood of pixels.

3.5 Applying Bounding Boxes to Image

In object detection, we usually use a bounding box to describe the target location. The bounding box is a rectangular box that can be determined by the x and y axis coordinates in

the upper-left corner and the x and y axis coordinates in the lower-right corner of the rectangle. Another commonly used bounding box representation is the x and y axis coordinates of the bounding box center, and its width and height. Here we define functions to convert between these two representations, `box_corner_to_center` converts from the two-corner representation to the center-width-height presentation, and `box_center_to_corner` vice versa.

CHAPTER 4

Result Analysis

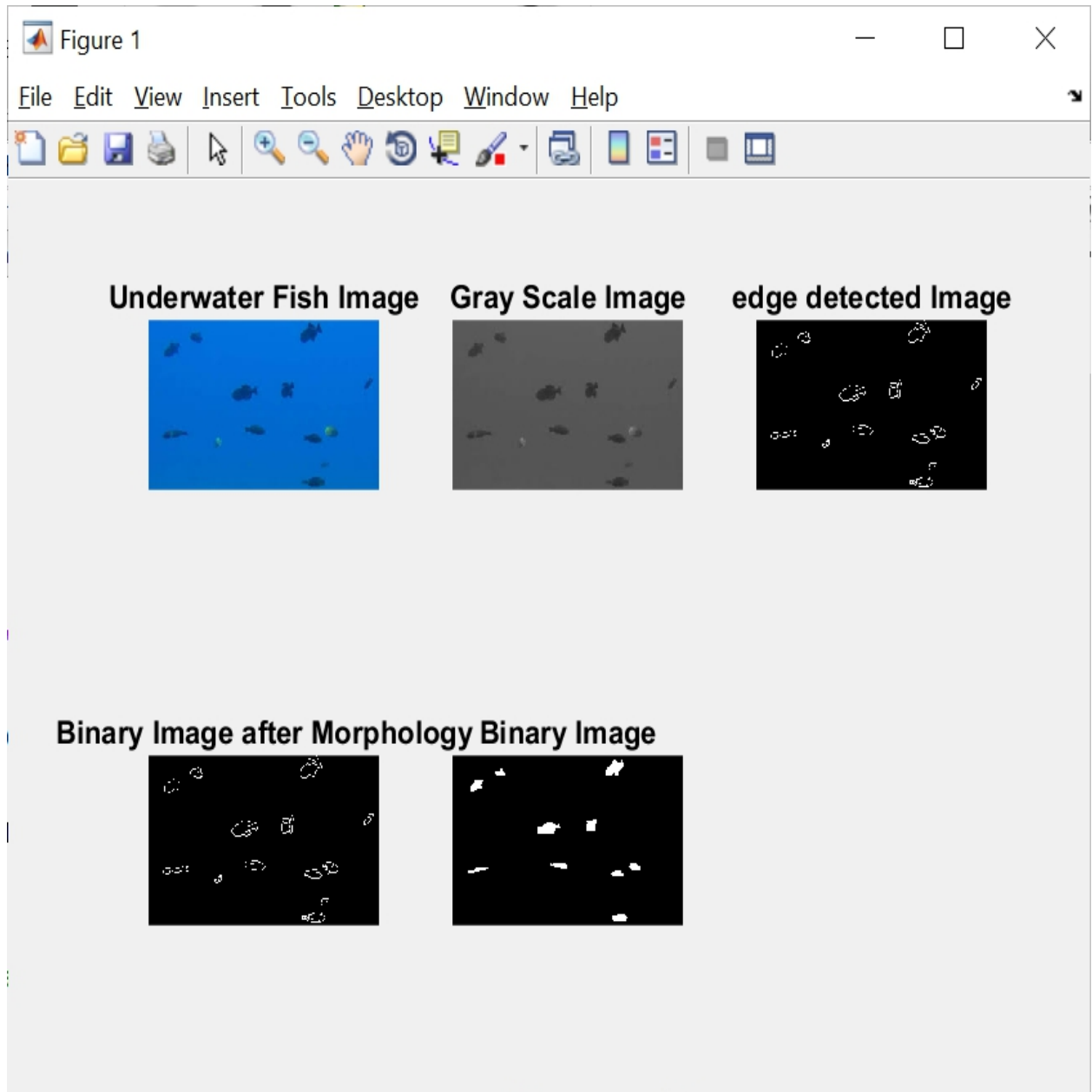


Fig.4.1 Fish Detection from underwater images

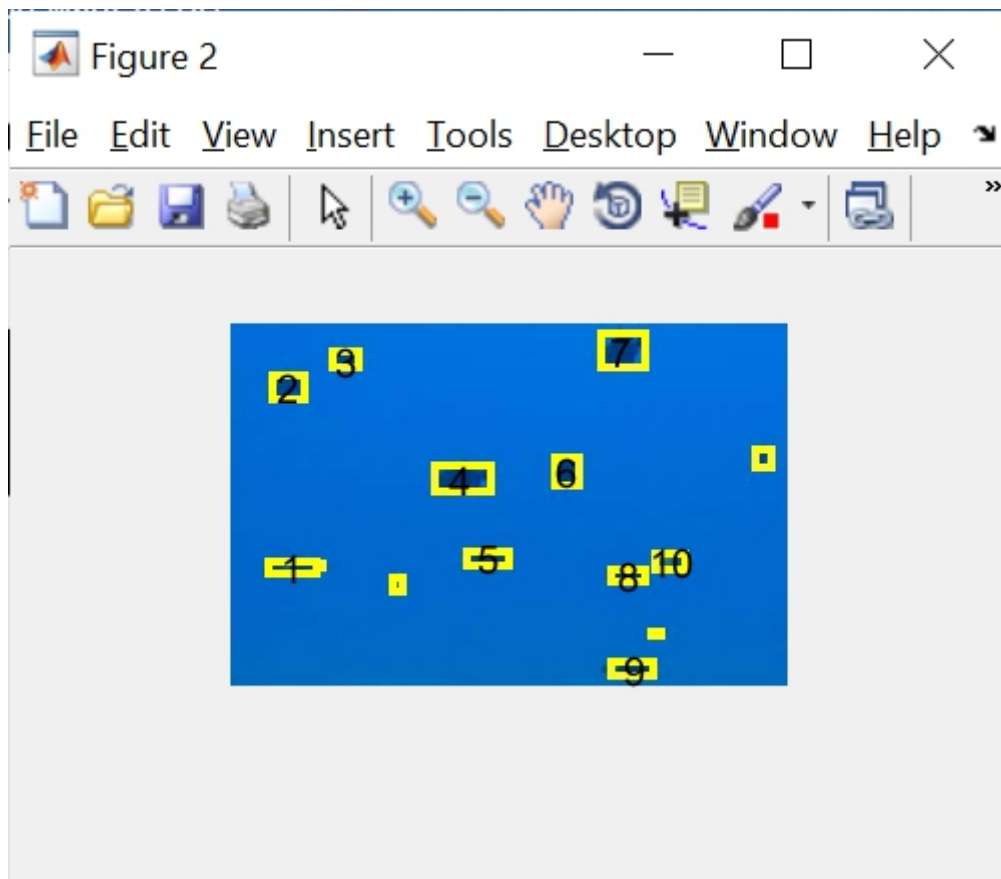


Fig.4.2 Detection and counting Number of Fishes (Applied Bounding Box)

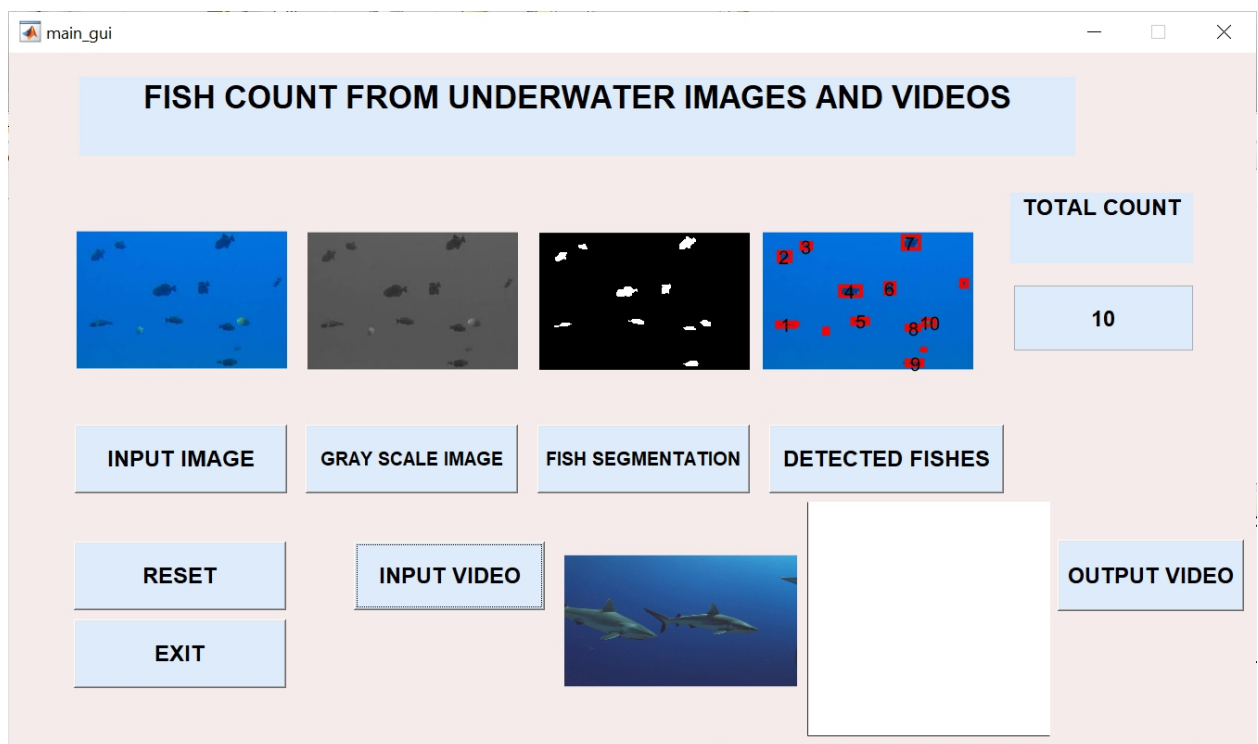


Fig.4.3 GUI selected input video

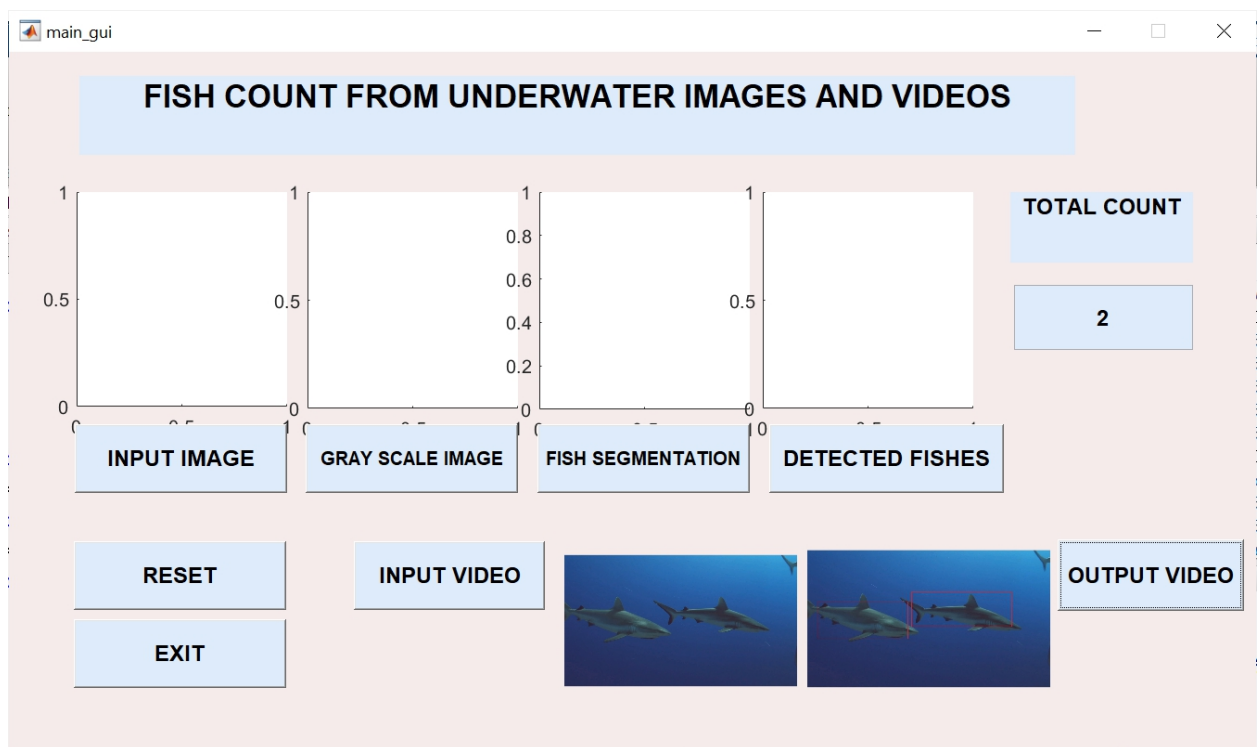


Fig.4.4 Detected Fishes from underwater video

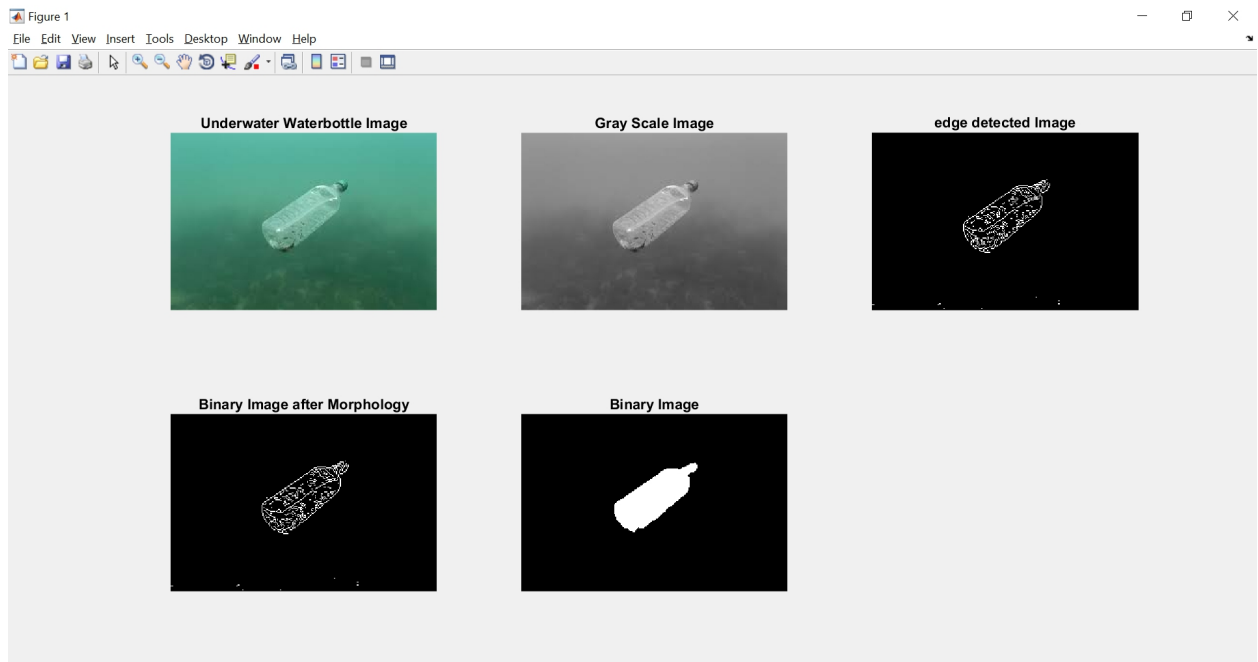


Fig.4.5 Underwater water bottle Image

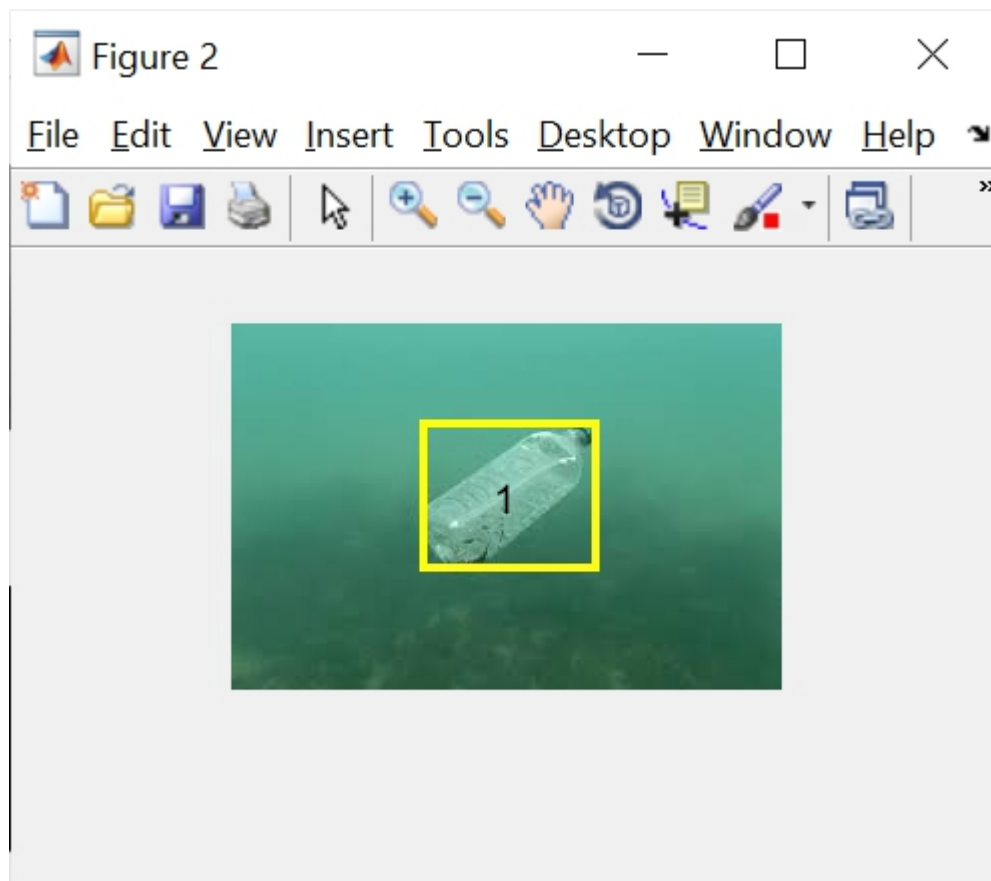


Fig.4.6 Detected water bottle from underwater image

CHAPTER 5

SOFTWARE ENVIRONMENT

5.1 Introduction

MATLAB is a excessive-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment wherein issues and answers are expressed in familiar mathematical notation. Typical uses encompass

5.2 Applications of MATLAB:

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data evaluation, exploration, and visualization
- Scientific and engineering pictures
- Application improvement, along with graphical consumer interface constructing.

5.3 MATLAB

MATLAB is a immoderate-overall performance language for technical computing. It integrates computation, visualization, and programming in an clean-to-use surroundings wherein troubles and answers are expressed in acquainted mathematical notation. MATLAB stands for matrix laboratory, and become written first off to offer smooth get entry to to matrix software application advanced by using LINPACK (linear system bundle) and EISPACK (Eigen system package deal deal deal) responsibilities.

MATLAB is consequently constructed on a basis of sophisticated matrix software program application in which the important element is array that does not require pre dimensioning which to resolve many technical computing issues, especially people with matrix and vector formulations, in a fragment of time. MATLAB talents a own family of programs specific answers referred to as toolboxes. Very important to most customers of MATLAB, toolboxes permit analyzing and utilizing specialised era. These are complete collections of MATLAB

features (M-documents) that growth the MATLAB environment to remedy unique commands of problems. Areas in which toolboxes are available embody sign processing, manipulate device, neural networks, fuzzy appropriate judgment, wavelets, simulation and plenty of others.

Typical uses of MATLAB encompass: Math and computation, Algorithm improvement, Data acquisition, Modeling, simulation, prototyping, Data evaluation, exploration, visualization, Scientific and engineering pics, Application development, along with graphical character interface constructing. Boom the MATLAB environment to remedy unique commands of problems. Areas wherein toolboxes are to be had embody signal processing, control device, neural networks, fuzzy appropriate judgment, wavelets, simulation and lots of others.

Typical uses of MATLAB include: Math and computation, Algorithm improvement, Data acquisition, Modeling, simulation, prototyping, Data analysis, exploration, visualization, Scientific and engineering photos, Application development, along side graphical character interface constructing.

5.4 Basic Building Blocks of MATLAB

The fundamental building block of MATLAB is MATRIX. The essential information type is the array. Vectors, scalars, real matrices and complicated matrix are dealt with as particular magnificence of this fundamental information kind. The built in functions are optimized for vector operations. No size statements are required for vectors or arrays.

5.4.1 MATLAB Window

The MATLAB works based totally on five windows: Command window, Workspace window, Current directory window, Command history window, Editor Window, Graphics window and Online-help window.

5.4.2 Command Window

The command window is in which the person sorts MATLAB instructions and expressions on the spark off (`>>`) and in which the output of those commands is displayed. It is opened when the utility program is released. All instructions inclusive of user-written applications are typed in this window at MATLAB activate for execution.

5.4.3 Work Space Window

MATLAB defines the workspace as the set of variables that the person creates in a piece consultation. The workspace browser indicates these variables and a few records about them.

Double clicking on a variable inside the workspace browser launches the Array Editor, which can be used to gain facts.

5.4.4 Three Current Directory Window

The present day Directory tab suggests the contents of the present day listing, whose route is hooked up inside the present day listing window. For example, within the domestic windows working tool the path is probably as follows: C:\MATLAB\Work, indicating that directory “paintings” is a subdirectory of the precept list “MATLAB”; it's far mounted in pressure C. Clicking at the arrow in the contemporary listing window shows a listing of currently used paths. MATLAB uses a are in search of for route to discover M-files and different MATLAB related documents. Any document run in MATLAB need to live inside the modern listing or in a list that is on trying to find path.

5.4.5 Four Command History Window

The Command History Window consists of a file of the instructions someone has entered within the command window, which includes each modern and previous MATLAB periods. Previously entered MATLAB instructions may be selected and re-accomplished from the command records window through right clicking on a command or collection of commands. This is beneficial to pick out various alternatives further to executing the instructions and is useful characteristic while experimenting with numerous commands in a piece consultation.

5.4.6 Editor Window

The MATLAB editor is every a text editor specialized for growing M-documents and a graphical MATLAB debugger. The editor can seem in a window through the use of itself, or it may be a sub window inside the desktop. In this window you can clearly write, edit, create and preserve applications in documents called M-files.

MATLAB editor window has numerous pull-down menus for responsibilities collectively with saving, viewing, and debugging documents. Because it performs a few smooth assessments and additionally makes use of shade to differentiate among severa factors of code, this newsletter editor is recommended because the device of desire for writing and improving M-functions.

5.4.7 Graphics or Figure Window

The output of all photo instructions typed inside the command window is seen in this window.

5.4.8 Online Help Window

MATLAB presents online assist for all it's constructed in features and programming language constructs. The primary manner to get help on-line is to apply the MATLAB assist browser, opened as a separate window each with the useful resource of clicking on the question mark image (?) at the desktop toolbar, or with the aid of typing help browser at the activate inside the command window. The assist Browser is a web browser blanketed into the MATLAB computing tool that shows a Hypertext Markup Language (HTML) documents. The Help Browser includes planes, the assist navigator pane, used to find out statistics, and the display pane, used to view the statistics. Self-explanatory tabs other than navigator pane are used to carry out a are seeking.

5.5 MATLAB Files

MATLAB has 3 varieties of files for storing facts. They are: M-documents and MAT-files.

A.Three.1 M-Files

These are widespread ASCII text record with 'm' extension to the document call and growing own matrices using M-files, which are text documents containing MATLAB code. MATLAB editor or any other text editor is used to create a file containing the same statements which are typed at the MATLAB command line and save the record underneath a call that leads to .M. There are sorts of M-documents:

1. Script Files

It is an M-file with a hard and fast of MATLAB commands in it and is achieved by way of typing call of record on the command line. These documents paintings on worldwide variables presently found in that surroundings.

2. Function Files

A characteristic report is also an M-file besides that the variables in a feature record are all local. This type of files starts with a characteristic definition line.

5.6 MAT-Files

These are binary information documents with .Mat extension to the record which can be created through MATLAB when the records is saved. The information written in a unique format that only MATLAB can examine. These are located into MATLAB with 'load' command.

5.7 The MATLAB System:

The MATLAB machine consists of five important components:

5.8 Development Environment:

This is the set of tools and centers that assist you use MATLAB functions and files. Many of those gear are graphical person interfaces. It consists of the MATLAB computer and Command Window, a command history, an editor and debugger, and browsers for viewing assist, the workspace, files, and the quest path.

5.9 MATLAB Mathematical Function:

This is a large series of computational algorithms ranging from elementary features like sum, sine, cosine, and complex mathematics, to greater sophisticated features like matrix inverse, matrix eigen values, Bessel functions, and rapid Fourier transforms.

5.10 MATLAB Language:

This is a excessive-stage matrix/array language with manage glide statements, capabilities, statistics systems, input/output, and item-orientated programming features. It lets in both "programming in the small" to rapidly create quick and dirty throw-away packages, and "programming within the massive" to create complete big and complicated software applications.

5.11 Four Graphics:

MATLAB has large centers for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It consists of high-degree features for two-dimensional and 3-dimensional statistics visualization, picture processing, animation, and presentation pics. It also includes low-level functions that let you fully customize the appearance of photographs as well as to build complete graphical consumer interfaces to your MATLAB packages.

5.12 MATLAB Application Program Interface (API):

This is a library that allows you to put in writing C and FORTRAN packages that have interaction with MATLAB. It consists of centers for calling exercises from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

Demo demonstrates what is possible in **Mat lab**

Who lists all of the variables in your Mat lab workspace?

Whose list the variables and describes their matrix size

clear erases variables and functions from memory

clear x erases the matrix 'x' from your workspace

close by itself, closes the current figure window

figure creates an empty figure window

hold on holds the current plot and all axis properties so that subsequent graphing commands add to the existing graph

hold off sets the next plot property of the current axes to "replace"

find find indices of nonzero elements e.g.:

`d = find(x>100)` returns the indices of the vector x that are greater than 100

break terminate execution of m-file or WHILE or FOR loop

5.13 BASIC PLOT COMMANDS:

Kinds of plots:

`plot(x,y)` creates a Cartesian plot of the vectors x & y

`plot(y)` creates a plot of y vs. The numerical values of the factors inside the y-vector

`semilogx(x,y)` plots $\log(x)$ vs y

`semilogy(x,y)` plots x vs $\log(y)$

`loglog(x,y)` plots $\log(x)$ vs $\log(y)$

`polar(theta,r)` creates a polar plot of the vectors r & theta wherein theta is in radians

`bar(x)` creates a bar graph of the vector x. (Note also the command `stairs(x)`)

`bar(x, y)` creates a bar-graph of the elements of the vector y, finding the bars

consistent with the vector factors of 'x'

Plot description: `grid` creates a grid at the images plot

5.14 ALGEBRIC OPERATIONS IN MATLAB:

Scalar Calculations:

+ Addition

- Subtraction

* Multiplication

/ Right department (a/b approach $a \div b$)

left division (ab manner $b \div a$)

^ Exponentiation

For instance `three*four` carried out in 'matlab' gives `ans=12`

`4/five` offers `ans=0.Eight`

Array products: Recall that addition and subtraction of matrices involved addition or subtraction of the character factors of the matrices. Sometimes it's far favored to definitely multiply or divide every element of an matrix by way of the corresponding element of any other matrix 'array operations'.

Array or detail-by way of-element operations are accomplished when the operator is preceded via a '.' (Period):

`a.*b` multiplies each detail of a by using the respective element of b

`a./b` divides each element of a by using the respective detail of b

`a./B` divides every detail of b by using the respective detail of a

`a.^b` raise each detail of a with the aid of the respective b element

5.15 MATLAB WORKING ENVIRONMENT:

5.15.1 MATLAB DESKTOP

Matlab Desktop is the principle Matlab software program window. The laptop includes 5 sub domestic home windows, the command window, the workspace browser, the current list window, the command history window, and one or greater determine home windows, which can be tested most effective when the individual displays a photo.

The command window is in which the person kinds MATLAB instructions and expressions at the spark off (`>>`) and wherein the output of those commands is displayed. MATLAB defines the workspace as the set of variables that the patron creates in a work session. The

cutting-edge Directory tab above the workspace tab suggests the contents of the present day listing, whose path is shown in the current list window.

For example, in the domestic home windows running device the course is probably as follows: C:\MATLAB\Work, indicating that listing “art work” is a subdirectory of the primary listing “MATLAB”; WHICH IS INSTALLED IN DRIVE C. Clicking at the arrow within the modern list window indicates a list of these days used paths. Clicking at the button to the proper of the window permits the user to change the modern-day listing. MATLAB uses a are seeking for route to locate M-files and other MATLAB related files, which is probably arrange in directories within the laptop report system. Any document run in MATLAB want to are living inside the cutting-edge list or in a list this is on are seeking route. By default, the documents supplied with MATLAB and math works toolboxes are protected within the are looking for direction.

The simplest manner to look which directories are soon the quest path, or to characteristic or alter a seek direction, is to choose set path from the File menu the laptop, after which use the set direction communication box. It is right exercise to feature any normally used directories to the search course to avoid time and again having the change the cutting-edge list.

The Command History Window includes a file of the commands a consumer has entered in the command window, along with every modern and previous MATLAB periods. Previously entered MATLAB commands can be determined on and re-accomplished from the command history window by the use of right clicking on a command or sequence of commands.

This motion launches a menu from which to choose out numerous options further to executing the instructions. This is useful to choose out severa options in addition to executing the instructions. This is a beneficial feature at the same time as experimenting with various commands in a piece consultation.

5.15.2 MATLAB Editor to create M-File:

The MATLAB editor is each a textual content editor specialized for growing M-files and a graphical MATLAB debugger. The editor can appear in a window via itself, or it could be a sub window within the laptop. M-files are denoted by using the usage of the extension .M, as in pixelThe MATLAB editor window has numerous pull-down menus for tasks together with

saving, viewing, and debugging files. Because it plays a few simple assessments and also makes use of coloration to distinguish among several elements of code, this newsletter editor is recommended due to the fact the tool of desire for writing and modifying M-abilities.

To open the editor, kind edit at the prompt opens the M-record filename.M in an editor window, equipped for enhancing. As stated earlier, the report must be inside the modern-day listing, or in a listing within the are looking for route.

5.16 Three Getting Help:

The most critical manner to get assist online is to use the MATLAB help browser, opened as a separate window either through clicking on the question mark symbol (?) at the computer toolbar, or through typing help browser on the set off in the command window. The help Browser is an internet browser incorporated into the MATLAB laptop that presentations a Hypertext Markup Language(HTML) files. The Help Browser includes two panes, the help navigator pane, used to find records, and the display pane, used to view the statistics. Self-explanatory tabs apart from navigator pane are used to carry out a search.

CHAPTER 6

INTRODUCTION TO DIGITAL IMAGE PROCESSING

6.1 Digital Image Processing

An image may be defined as a two-dimensional function $f(x, y)$, where x & y are spatial coordinates, & the amplitude of at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x , y & the amplitude values of f are all finite discrete quantities, we call the image a digital image. The field of DIP refers to processing digital image by means of digital computer. Digital image is composed of a finite number of elements, each of which has a particular location & value. The elements are called pixels.

Vision is the most advanced of our sensor, so it is not surprising that image play the single most important role in human perception. However, unlike humans, who are limited to the visual band of the EM spectrum imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves. They can operate also on images generated by sources that humans are not accustomed to associating with image.

There is no general agreement among authors regarding where image processing stops & other related areas such as image analysis & computer vision start. Sometimes a distinction is made by defining image processing as a discipline in which both the input & output at a process are images. This is limiting & somewhat artificial boundary. The area of image analysis (image understanding) is in between image processing & computer vision.

There are no clear-cut boundaries in the continuum from image processing at one end to complete vision at the other. However, one useful paradigm is to consider three types of computerized processes in this continuum: low-, mid-, & high-level processes. Low-level process involves primitive operations such as image processing to reduce noise, contrast enhancement & image sharpening. A low-level process is characterized by the fact that both its inputs & outputs are images. Mid-level process on images involves tasks such as segmentation, description of that object to reduce them to a form suitable for computer processing & classification of individual objects. A mid-level process is characterized by the fact that its inputs generally are images but its outputs are attributes extracted from those images. Finally higher-level processing involves “Making sense” of an ensemble of recognized objects, as in image analysis & at the far end of the continuum performing the cognitive functions normally associated with human vision.

Digital image processing, as already defined is used successfully in a broad range of areas of exceptional social & economic value.

6.2 Types of Images

An image is represented as a two dimensional function $f(x, y)$ where x and y are spatial co-ordinates and the amplitude of 'f' at any pair of coordinates (x, y) is called the intensity of the image at that point.

Gray scale image:

A grayscale image is a function $I(x, y)$ of the two spatial coordinates of the image plane.

$I(x, y)$ is the intensity of the image at the point (x, y) on the image plane.

$I(x, y)$ takes non-negative values assume the image is bounded by a rectangle $[0, a] \times [0, b]$:
 $[0, a] \times [0, b] \rightarrow [0, \text{info})$

Color image:

It can be represented by three functions, $R(x, y)$ for red, $G(x, y)$ for green and $B(x, y)$ for blue.

An image may be continuous with respect to the x and y coordinates and also in amplitude. Converting such an image to digital form requires that the coordinates as well as the amplitude to be digitized. Digitizing the coordinate's values is called sampling. Digitizing the amplitude values is called quantization.

6.3 Coordinate convention:

The result of sampling and quantization is a matrix of real numbers. We use two principal ways to represent digital images. Assume that an image $f(x, y)$ is sampled so that the resulting image has M rows and N columns. We say that the image is of size $M \times N$. The values of the coordinates (x, y) are discrete quantities. For notational clarity and convenience, we use integer values for these discrete coordinates. In many image processing books, the image origin is defined to be at $(x, y) = (0, 0)$. The next coordinate values along the first row of the image are $(x, y) = (0, 1)$. It is important to keep in mind that the notation $(0, 1)$ is used to signify the second sample along the first row. It does not mean that these are the actual values of physical coordinates when the image was sampled. Following figure shows the coordinate convention. Note that x ranges from 0 to $M-1$ and y from 0 to $N-1$ in integer increments.

The coordinate convention used in the toolbox to denote arrays is different from the preceding paragraph in two minor ways. First, instead of using (xylem) the toolbox uses the notation (race) to indicate rows and columns. Note, however, that the order of coordinates is the same as the order discussed in the previous paragraph, in the sense that the first element of a coordinate topples, (alb), refers to a row and the second to a column. The other difference is that the origin of the coordinate system is at $(r, c) = (1, 1)$; thus, r ranges from 1 to M and c from 1 to N in integer increments. IPT documentation refers to the coordinates. Less frequently the toolbox also employs another coordinate convention called spatial coordinates which uses x to refer to columns and y to refers to rows. This is the opposite of our use of variables x and y .

6.4 Image as Matrices:

The preceding discussion leads to the following representation for a digitized image function:

$$\begin{array}{cccc}
 f(0, 0) & f(0, 1) & \dots\dots\dots & f(0, N-1) \\
 f(1, 0) & f(1, 1) & \dots\dots\dots & f(1, N-1) \\
 \vdots & \vdots & & \vdots \\
 f(M-1, 0) & f(M-1, 1) & \dots\dots\dots & f(M-1, N-1)
 \end{array}$$

The right side of this equation is a digital image by definition. Each element of this array is called an image element, picture element, pixel or pel. The terms image and pixel are used throughout the rest of our discussions to denote a digital image and its elements.

A digital image can be represented naturally as a MATLAB matrix:

$$\begin{array}{cccc}
 f(1, 1) & f(1, 2) & \dots\dots\dots & f(1, N) \\
 f(2, 1) & f(2, 2) & \dots\dots\dots & f(2, N) \\
 \vdots & \vdots & & \vdots \\
 f(M, 1) & f(M, 2) & \dots\dots\dots & f(M, N)
 \end{array}$$

Where $f(1, 1) = f(0, 0)$ (note the use of a monospace font to denote MATLAB quantities). Clearly the two representations are identical, except for the shift in origin. The notation $f(p, q)$ denotes the element located in row p and the column q . For example $f(6, 2)$ is the element in the sixth row and second column of the matrix f . typically we use the letters M and N respectively to denote the number of rows and columns in a matrix. A $1 \times N$ matrix is called a row vector whereas an $M \times 1$ matrix is called a column vector. A 1×1 matrix is a scalar.

Matrices in MATLAB are stored in variables with names such as A , a , RGB , real array and so on. Variables must begin with a letter and contain only letters, numerals and underscores. As noted in the previous paragraph, all MATLAB quantities are written using monospace characters. We use conventional Roman, italic notation such as $f(x, y)$, for mathematical expressions

6.5 Reading Images:

Images are read into the MATLAB environment using function `imread` whose syntax is

```
imread('filename')
```

Format name	Description	recognized extension
TIFF	Tagged Image File Format	.tif, .tiff
JPEG	Joint Photograph Experts Group	.jpg, .jpeg
GIF	Graphics Interchange Format	.gif
BMP	Windows Bitmap	.bmp
PNG	Portable Network Graphics	.png
XWD	X Window Dump	.xwd

Here filename is a string containing the complete of the image file(including any applicable extension).For example the command line

```
>> f = imread ('8.jpg');
```

reads the JPEG (above table) image chestxray into image array f. Note the use of single quotes (') to delimit the string filename. The semicolon at the end of a command line is used by MATLAB for suppressing output. If a semicolon is not included, MATLAB displays the results of the operation(s) specified in that line. The prompt symbol(>>) designates the beginning of a command line, as it appears in the MATLAB command window.

When as in the preceding command line no path is included in filename, imread reads the file from the current directory and if that fails it tries to find the file in the MATLAB search path. The simplest way to read an image from a specified directory is to include a full or relative path to that directory in filename.

For example,

```
>> f = imread( 'D:\myimages\chestxray.jpg');
```

reads the image from a folder called my images on the D: drive, whereas

```
>> f = imread( '.\myimages\chestxray.jpg');
```

reads the image from the my images subdirectory of the current of the current working directory. The current directory window on the MATLAB desktop toolbar displays MATLAB's current working directory and provides a simple, manual way to change it. Above table lists some of the most of the popular image/graphics formats supported by imread and imwrite.

Function size gives the row and column dimensions of an image:

```
>> size (f)
```

```
ans = 1024 * 1024
```

This function is particularly useful in programming when used in the following form to determine automatically the size of an image:

```
>>[M,N]=size(f);
```

This syntax returns the number of rows(M) and columns(N) in the image.

The whole function displays additional information about an array. For instance ,the statement

```
>>whos f
```

gives

Name	size	Bytes	Class
F	1024*1024	1048576	unit8 array

Grand total is 1048576 elements using 1048576 bytes

The unit8 entry shown refers to one of several MATLAB data classes. A semicolon at the end of a whose line has no effect ,so normally one is not used.

6.6 Displaying Images:

Images are displayed on the MATLAB desktop using function imshow, which has the basic syntax:

```
imshow(f,g)
```

Where f is an image array, and g is the number of intensity levels used to display it. If g is omitted ,it defaults to 256 levels .using the syntax

```
Imshow (f, {low high})
```

Displays as black all values less than or equal to low and as white all values greater than or equal to high. The values in between are displayed as intermediate intensity values using the default number of levels .Finally the syntax

```
Imshow(f,[ ]) )
```

Sets variable low to the minimum value of array f and high to its maximum value. This form of imshow is useful for displaying images that have a low dynamic range or that have positive and negative values.

Function `pixval` is used frequently to display the intensity values of individual pixels interactively. This function displays a cursor overlaid on an image. As the cursor is moved over the image with the mouse the coordinates of the cursor position and the corresponding intensity values are shown on a display that appears below the figure window. When working with color images, the coordinates as well as the red, green and blue components are displayed. If the left button on the mouse is clicked and then held pressed, `pixval` displays the Euclidean distance between the initial and current cursor locations.

The syntax form of interest here is `Pixval` which shows the cursor on the last image displayed. Clicking the X button on the cursor window turns it off.

The following statements read from disk an image called `rose_512.tif` extract basic information about the image and display it using `imshow` :

```
>>f=imread('rose_512.tif');
```

```
>>whos f
```

Name	Size	Bytes	Class
F	512*512	262144	unit8 array

Grand total is 262144 elements using 262144 bytes

```
>>imshow(f)
```

A semicolon at the end of an `imshow` line has no effect, so normally one is not used. If another image, `g`, is displayed using `imshow`, MATLAB replaces the image in the screen with the new image. To keep the first image and output a second image, we use function `figure` as follows:

```
>>figure ,imshow(g)
```

Using the statement

```
>>imshow(f),figure ,imshow(g) displays both images.
```

Note that more than one command can be written on a line ,as long as different commands are properly delimited by commas or semicolons. As mentioned earlier, a semicolon is used whenever it is desired to suppress screen outputs from a command line.

Suppose that we have just read an image `h` and find that using `imshow` produces the image. It is clear that this image has a low dynamic range, which can be remedied for display purposes by using the statement.

```
>>imshow(h,[ ])
```

6.7 WRITING IMAGES:

Images are written to disk using function `imwrite`, which has the following basic syntax:

```
Imwrite (f,'filename')
```

With this syntax, the string contained in `filename` must include a recognized file format extension .Alternatively, the desired format can be specified explicitly with a third input argument. >>`imwrite(f,'patient10_run1','tif')`

Or alternatively

For example the following command writes `f` to a TIFF file named `patient10_run1`:

```
>>imwrite(f,'patient10_run1.tif')
```

If `filename` contains no path information, then `imwrite` saves the file in the current working directory.

The `imwrite` function can have other parameters depending on e file format selected. Most of the work in the following deals either with JPEG or TIFF images ,so we focus attention here on these two formats.

More general `imwrite` syntax applicable only to JPEG images is

```
imwrite(f,'filename.jpg,,'quality',q)
```

where q is an integer between 0 and 100(the lower number the higher the degradation due to JPEG compression).

For example, for q=25 the applicable syntax is

```
>>imwrite(f,'bubbles25.jpg','quality',25)
```

The image for q=15 has false contouring that is barely visible, but this effect becomes quite pronounced for q=5 and q=0. Thus, an expectable solution with some margin for error is to compress the images with q=25. In order to get an idea of the compression achieved and to obtain other image file details, we can use function `imfinfo` which has syntax.

`Imfinfo filename`

Here filename is the complete file name of the image stored in disk.

For example,

```
>>imfinfo bubbles25.jpg
```

outputs the following information(note that some fields contain no information in this case):

Filename: 'bubbles25.jpg'

FileModDate: '04-jan-2003 12:31:26'

FileSize: 13849

Format: 'jpg'

Format Version: ''

Width: 714

Height: 682

Bit Depth: 8

Color Depth: 'grayscale'

Format Signature: ‘ ‘

Comment: { }

Where file size is in bytes. The number of bytes in the original image is corrupted simply by multiplying width by height by bit depth and dividing the result by 8. The result is 486948. Dividing this file size gives the compression ratio: $(486948/13849)=35.16$. This compression ratio was achieved. While maintaining image quality consistent with the requirements of the appearance. In addition to the obvious advantages in storage space, this reduction allows the transmission of approximately 35 times the amount of uncompressed data per unit time.

The information fields displayed by `imfinfo` can be captured in to a so called structure variable that can be for subsequent computations. Using the receding an example and assigning the name K to the structure variable.

We use the syntax `>>K=imfinfo('bubbles25.jpg');`

To store in to variable K all the information generated by command `imfinfo`, the information generated by `imfinfo` is appended to the structure variable by means of fields, separated from K by a dot. For example, the image height and width are now stored in structure fields K.Height and K.width.

As an illustration, consider the following use of structure variable K to compute the compression ratio for bubbles25.jpg:

```
>> K=imfinfo('bubbles25.jpg');
```

```
>> image_bytes = K.Width* K.Height* K.Bit Depth /8;
```

```
>> Compressed_bytes = K.FileSize;
```

```
>> Compression_ratio=35.162
```

Note that `iminfo` was used in two different ways. The first was to type `imfinfo('bubbles25.jpg')` at the prompt, which resulted in the information being displayed on the screen. The second was to type `K=imfinfo('bubbles25.jpg')`, which resulted in the

information generated by `imfinfo` being stored in `K`. These two different ways of calling `imfinfo` are an example of `command_ function` duality, an important concept that is explained in more detail in the MATLAB online documentation.

More general `imwrite` syntax applicable only to `tifimages` has the form

```
Imwrite(g,'filename.tif','compression','parameter',....'resloution',[colres rows] )
```

Where 'parameter' can have one of the following principal values: 'none' indicates no compression, 'pack bits' indicates pack bits compression (the default for non 'binary images') and 'ccitt' indicates ccitt compression. (the default for binary images).The 1*2 array [colres rows]

Contains two integers that give the column resolution and row resolution in dot per_ unit (the default values). For example, if the image dimensions are in inches, colres is in the number of dots(pixels)per inch (dpi) in the vertical direction and similarly for rows in the horizontal direction. Specifying the resolution by single scalar, res is equivalent to writing [res res].

```
>>imwrite(f,'sf.tif','compression','none','resolution',.....[300 300])
```

the values of the vector[colures rows] were determined by multiplying 200 dpi by the ratio 2.25/1.5, which gives 30 dpi. Rather than do the computation manually, we could write

```
>> res=round(200*2.25/1.5);
```

```
>>imwrite(f,'sf.tif','compression','none','resolution',res)
```

where its argument to the nearest integer.It function round rounds is important to note that the number of pixels was not changed by these commands. Only the scale of the image changed. The original 450*450 image at 200 dpi is of size 2.25*2.25 inches. The new 300_dpi image is identical, except that is 450*450 pixels are distributed over a 1.5*1.5_inch area. Processes such as this are useful for controlling the size of an image in a printed document with out sacrificing resolution.

Often it is necessary to export images to disk the way they appear on the MATLAB desktop. This is especially true with plots. The contents of a figure window can be exported to disk in two ways. The first is to use the file pull-down menu in the figure window and then choose export. With this option the user can select a location, filename, and format. More control over export parameters is obtained by using print command:

`Print-fno-dfileformat-rresno filename`

Where no refers to the figure number in the figure window interest, file format refers one of the file formats in table above. 'resno' is the resolution in dpi, and filename is the name we wish to assign the file.

If we simply type print at the prompt, MATLAB prints (to the default printer) the contents of the last figure window displayed. It is possible also to specify other options with print, such as specific printing device.

CHAPTER 7

Conclusion and Future Work

We successfully designed Intelligent Object Recognition In Underwater Images Using Image Processing. It is used for fish detection and counting of underwater images as well as underwater videos. Fish detection and counting is emerging application for underwater fish-based applications. In this project we detected underwater fishes and detected count of fishes in a given image as well as video. We converted image into gray scale and applied edge detection technique to find fishes. The detected fishes are applied with bounding boxes and count is obtained. Similar application can be used for detection of water bottle. We applied image morphology technique for detection of objects (fish or water bottle) correctly. An outcome of the fish detection system is the fish characterization, which can be used for fish species recognition. Therefore, future work can be concentrated mainly on the classification task. Given the sufficient number of training results, a classification system based on fish size and texture can be build. Also, the proposed process of size estimation can be improved by introducing the information about camera parameters, which allows to perform the camera calibration.

REFERENCES

- [1] Andrews Sobral and Antoine Vacavant. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Computer Vision and Image Understanding*, 122:4–21, 2014.
- [2] Carl Schlieper. *Research methods in marine biology*. Sidgwick & Jackson, 1972.
- [3] Concetto Spampinato, Yun-Heh Chen-Burger, Gayathri Nadarajan, and Robert B Fisher. Detecting, tracking and counting fish in low quality unconstrained underwater videos. In *Proceedings of the Third International Conference on Computer Vision Theory and Applications*, volume 2, pages 514–519. Citeseer, 2008.
- [4] Helge Balk. *Development of Hydroacoustic Methods for Fish Detection in Shallow Water*. PhD thesis, Faculty of Mathematics and Natural Science, University of Oslo, 2001.
- [5] Janne Heikkilä and Olli Silvén. A real-time system for monitoring of cyclists and pedestrians. *Image and Vision Computing*, 22(7):563–570, 2004.
- [6] JL Thorley, DMR Eatherley, AB Stephen, I Simpson, JC MacLean, and AF Youngson. Congruence between automatic fish counter data and rod catches of atlantic salmon (*salmo salar*) in scottish rivers. *ICES Journal of Marine Science: Journal du Conseil*, 62(4):808–817, 2005.
- [7] Kinjal A Joshi and Darshak G Thakore. A survey on moving object detection and tracking in video surveillance system. *International Journal of Soft Computing and Engineering*, 2(3):44–48, 2012.
- [8] Massimo Piccardi. Background subtraction techniques: a review. In *Proceedings of the 2004 IEEE international conference on Systems, man and cybernetics*, volume 4, pages 3099–3104. IEEE, 2004.
- [9] Nils T Siebel and Steve Maybank. The advisor visual surveillance system. In *Proceedings of the 8th European Conference on Computer Vision Workshop Applications of Computer Vision*, volume 1, 2004.
- [10] Peter Christiansen, Kim Arild Steen, Rasmus Nyholm Jørgensen, and Henrik Karstoft. Automated detection and recognition of wildlife using thermal cameras. *Sensors*, 14(8):13778–13793, 2014.
- [11] Rita Cucchiara, Costantino Grana, Massimo Piccardi, and Andrea Prati. Detecting moving objects, ghosts, and shadows in video streams. *IEEE transactions on pattern analysis and machine intelligence*, 25(10):1337–1342, 2003.

- [12] Robert T Collins, Alan J Lipton, Takeo Kanade, Hironobu Fujiyoshi, David Duggins, Yanghai Tsin, David Tolliver, Nobuyoshi Enomoto, Osamu Hasegawa, Peter Burt, et al. A system for video surveillance and monitoring. 2000.
- [13] Shireen Y Elhabian, Khaled M El-Sayed, and Sumaya H Ahmed. Moving object detection in spatial domain using background removal techniques-state-of-art. *Recent patents on computer science*, 1(1):32–54, 2008.
- [14] Soharab Hossain Shaikh, Khalid Saeed, and Nabendu Chaki. *Moving Object Detection Using Background Subtraction*. Springer, 2014.
- [15] Suman Srinivasan, Haniph Latchman, John Shea, Tan Wong, and Janice McNair. Airborne traffic surveillance systems: video surveillance of highway traffic. In *Proceedings of the ACM 2nd International Workshop on Video Surveillance & Sensor Networks*, pages 131–135. ACM, 2004.
- [16] Tejinder Thind Bharti. Background subtraction technique review. *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, 2(3):166– 168, 2013.
- [17] Thierry Bouwmans. Recent advanced statistical background modeling for foreground detection-a systematic survey. *Recent Patents on Computer Science*, 4(3):147–176, 2011.
- [18] Vijay Mahadevan and Nuno Vasconcelos. Background subtraction in highly dynamic scenes. In *Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–6. IEEE, 2008.
- [19] Yannick Benezeth, Pierre-Marc Jodoin, Bruno Emile, H  l  ne Laurent, and Christophe Rosenberger. Comparative study of background subtraction algorithms. *Journal of Electronic Imaging*, 19(3):033003–033003, 2010.
- [20] Yaser Sheikh, Omar Javed, and Takeo Kanade. Background subtraction for freely moving cameras. In *Proceedings of the 2009 IEEE 12th International Conference on Computer Vision*, pages 1219–1225. IEEE, 2009.

140_212.docx

by 140_212.docx 140_212.docx

Submission date: 15-Apr-2021 05:16 PM (UTC-0400)

Submission ID: 1553534477

File name: 140_212_262.docx (1.9 MB)

Word count: 13,246

Character count: 72,406

INTELLIGENT OBJECT RECOGNITION IN UNDERWATER IMAGES USING IMAGE PROCESSING

Submitted in partial fulfilment of requirements for the award of Bachelor of Engineering in
Electronics and Communication Engineering

by

KUDAPA SUCHARITHA (37130212) MUNGAMURU SAI DHANA
VARSHINI(37130262)



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING SCHOOL OF
ELECTRICAL AND ELECTRONICS

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY (DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI – 600119 MARCH – 2021



ANNA

UNIVERSITY OF SCIENCE AND TECHNOLOGY

UNIVERSITY)

Grade “A” by NAAC

ANNA NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119



DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **KUDAPA SUCHARITHA (37130212)** and **MUNGAMURU SAI DHANA VARSHINI (37130262)** who carried out the project entitled “**Intelligent Object Recognition in Underwater Images using Image Processing**” under our supervision from December 2020 to March 2021.

INTERNAL GUIDE

Dr.T.VINO, M.E., Ph.D.

Head of the Department (DR. T. RAVI)

Submitted for Viva voce Examination held on

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

We, KUDAPA SUCHARITHA (37130212) and MUNGAMURU SAI DHANA VARSHINI (37130262) hereby declare that the Project Report entitled “Intelligent Object Recognition in Underwater Images using Image Processing ” done by us under the guidance of Dr.T.Vino is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Electronics and Communication Engineering.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE 1.

2.

ACKNOWLEDGEMENT

We are pleased to acknowledge my sincere thanks to Board of Management of SATHYABAMA for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

We convey our thanks to **Dr. N. Nandhitha M.E.,Ph.D., Prof. & Dean, School of Electrical and Electronics Engineering** and **Dr. T. Ravi M.E.,P.hD., Head of the Department, Dept. of ELECTRONICS AND COMMUNICATION ENGINEERING** for providing me necessary support and details at the right time during the progressive reviews.

We would like to express my sincere and deep sense of gratitude to my Project Guide **Dr.T.Vino M.E.,Ph.D., Associate Professor, Dept of ELECTRONICS AND COMMUNICATION ENGINEERING** for his valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

WewishtoexpressmythankstoallTeachingandNon-teachingstaffmembersof the Department of **Electronics and Communication Engineering** who were helpful in many ways for the completion of the project.

TABLE OF CONTENT

CHAPTE R NO	TITLE	PAG E NO
1	Introduction	1
1.1.	Background Subtraction	4
1.2.	Fish Detection	5
2	Literature Survey	9
2.1.	Detecting, tracking and counting fish in low quality unconstrained underwater videos.	9
2.2.	Congruence between automatic fish counter Dataand rod catches of atlantic salmon	9
2.3	Development of Hydroacoustic Methods for Detection in Shallow Water.	11
2.4	A system for video surveillance and monitoring.	12
2.5	Airborne traffic surveillance systems: video surveillance of highway traffic.	14
2.6	Automated detection and recognition of wildlife using thermal cameras.	14
2.7	Moving object detection and tracking in video surveillance system.	16
2.8	Moving Object Detection Using Background Subtraction	17
	Proposed System	18

3.1. Underwater Images	18
3.2. Gray Scale Conversion	19
3.3. Edge Detection	19
3.4. Morphological Operations	19
3.5. Applying Bounding Box to Image	20
Result Analysis	21
Software Environment	25

5.1. Introduction	25
5.2. Applications of MATLAB	25
5.3. MATLAB	25
5.4. Basic Building Blocks of MATLAB	26
5.4.1. MATLAB Window	26
5.4.2. Command Window	26
5.4.3. Work Space Window	27
5.4.4. Three Current Directory Window	27
5.4.5. Four Command History Window	27
5.4.6. Editor Window	27
5.4.7. Graphics or Figure Window	28
5.4.8. Online Help Window	28
5.5. MATLAB Files	28
5.6. MAT-Files	29
5.7. The MATLAB System	29
5.8. Development Environment	29
5.9. MATLAB Mathematical Function	29
5.10. MATLAB Language	29
5.11. Four Graphics	29
5.12. MATLAB Application Program Interface(API)	29
5.13. Some Basic Plot Commands	30
5.14. Algebraic Operations In MATLAB	31
5.15. MATLAB Working Environment	32
5.14.1. MATLAB Desktop	32
5.14.2. MATAB Editor to Create M-Files	33
5.16. Three Getting Help	33

6.1. Introduction	34
6.2. Types of Images	35
6.3. Coordinate Convention	35
6.4. Images as Matrices	36
6.5. Reading Images	37
6.6. Displaying Images	39
6.7. Writing Images	41
Conclusion	47
Future Work	48
References	49

LIST OF FIGURES

FIG NO	TITLE	PAGE NO
1.1	Example Images	2
1.2	Block Diagram	4
1.3	BS process	5
1.4	Tracking example with two frames grabbed at two consecutive times	16
1.5	Difficult cases for fish detection.	16
3.1	Block Diagram for fish detection and counting from Image	18
4.1	Fish Detection from underwater images	21
4.2	Detection and counting Number of Fishes (Applied Bounding Box)	22
4.3	GUI selected input video	23
4.4	Detected Fishes from underwater video	23

4.5	Underwater water bottle Image	24
4.6	Detected water bottle from underwater image	24

CHAPTER 1

INTRODUCTION

Counting and tracking fish populations is important for conservation purposes as well as for the fishing industry. Usually, biologists determine the situation under the water by casting nets for collecting and examining fish or by human underwater observation [1]. Such methods are not able to provide a comprehensive observation, cannot capture the real fish state and behavior, and require much time and expenses. In addition, there is a big risk to kill or damage fish and their habitat during the collection process. These drawbacks force researchers to develop alternative ways of marine life observation [2]. One of the most commonly used approaches for underwater observation is to collect videos and analyse them to solve different tasks of marine life monitoring, for instance, fish counting.

Many systems for underwater fish observation located in rivers and fish farms involve the manual counting of the passing fish, while the automation can speed up the process significantly, reduce the costs associated with the involvement of human experts and provide an opportunity for fish recognition. Various automatic fish counters already exist, such as resistive counters that rely on the fact that the resistivity of a fish is lower than that of water [3], optical counters that use light beams [3], and hydro acoustic counters that operate using the principles of sonar [4]. Such approaches, based on different physical principles, cope with the task of counting passing objects quite successfully. However, such methods typically cannot make distinction between fish and other passing objects, and moreover, cannot recognize different species. Computer vision techniques provide an attractive alternative for building a more robust and versatile fish counting systems.

(b)

Fig 1.1 Example images: (a) High illumination; (b) Low illumination.

Barely any countries are concentrating on developing advances. The greater part of the advances is centered around complex life for people. Be that as it may, the impact of innovation is poor. Because of the absence of innovative ability, the fishing market is reliant on past based information or chance which brings about disappointments or not suitable yield. Analysts additionally need to go through more cash to consider the natural examples. 3 A definitive point of the task is to recognize the fishes submerged and attempt to group them. By the distinguishing proof of the various species, we can consider the diverse way of life example of fishes and how It influences the earth and help in business angling. For the AI process, a profound learning technique is utilized.

Underwater video is used for studying marine and riverine ecosystems, mainly for observing animal abundance and behavior in the context of environmental monitoring. Optical video provides detailed information in a form that is easy to interpret because humans naturally interpret their world visually. But human analysis of video takes time, so some level of automation is needed to process large video data sets in a timely, efficient manner in order to provide information needed for permitting and adaptive management decisions. Computer vision and machine learning have proven effective for automating similar monitoring applications, such as video surveillance. Underwater imagery presents unique challenges that include sudden illumination changes, unequal spectral propagation (i.e., color content is affected by distance), low contrast, clutter in the form of floating vegetation, and changes in visibility due to turbidity.

Object Detection Methods Video surveillance systems are currently widely used for analysing video sequences and applied, for example, in security control [5], traffic monitoring [6], anomaly detection [7] and animal monitoring and conservation [8]. Some systems include human intervention, but involving of a human operator may slow down the process of video analysis significantly and make the task impracticable.

Therefore, the creation of autonomous video surveillance systems is one of the most actively developing areas of computer vision. The term autonomous in this case implies that the system receives a video as an input and performs both the low-level tasks, like motion detection, and high-level decisions like object recognition, without human expert involvement. [9] Regardless the type of high-level task, the crucial step which must be done for further analysis is the moving object detection. It separates moving objects, which form the foreground, from the stationary background. Different approaches for the moving object detection from videos can be classified in the following way [10]:

- **Background subtraction (BS)** is commonly used in static scenes. It uses a modelled background image and subtract the

current image pixel-by-pixel from it to detect regions of motion. With each new image a background model is updated to adapt to scene changes.

- **Temporal differencing** is usually applied in cases where the camera is moving and the background is not stable. Moving regions are detected by taking pixel-by-pixel difference of consecutive frames in a video sequence.
- **Statistical approaches** are a modification of BS methods, that use statistical information of each pixel and compare pixels' statistics to identify foreground.
- **Optical flow** can be used with the moving camera and moving background, but are computationally complex and time consuming. The basic idea is to compute the apparent velocity and direction of every pixel to find moving objects.

Foreground pixels from the background by using the defined background model. During the following step pixel level processing, including, for example, morphological operations and low pass filter, is performed to remove the noise. Then the foreground pixels are grouped into connected regions and the bounded boxes of the regions are calculated. Finally, after a set of individual regions is obtained, all false regions, which do not correspond to objects, are removed based on, for example, size.

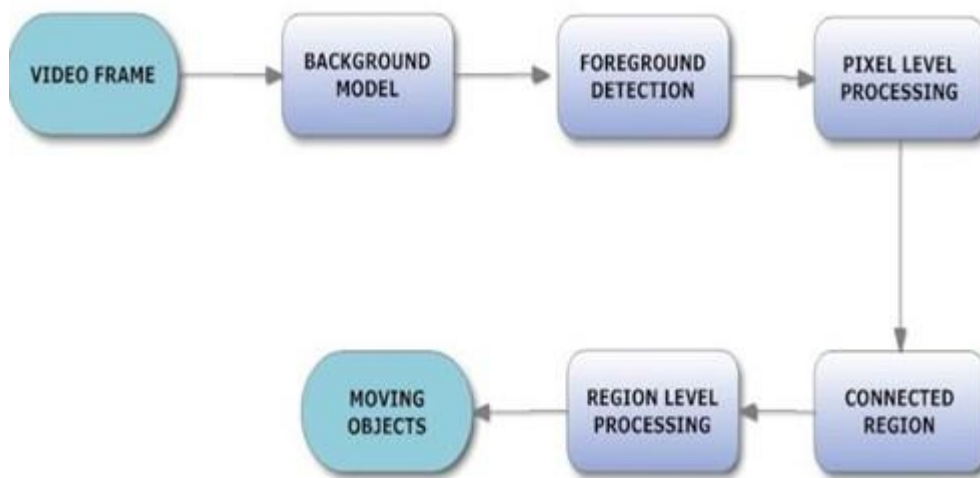


Fig 1.2. Block Diagram

1.1 Background Subtraction

BS methods are widely used for solving the task of moving object detection. These algorithms are relatively fast, which makes it possible to detect objects in real time. Moreover, they do not require a lot of memory and show robust results. BS approach is mostly applicable when both the camera and the background are stable, but can be easily modified for the cases with moving camera or background [12,13]. Due to its benefits, BS methods have gained popularity among researches, which led to the emergence of a large number of different algorithms [14–16].

Most of the BS methods are implemented by the same scheme, which consists of three main steps (Figure 3):

• Background modeling

This ⁴ is the first step of every BS algorithm. For detecting foreground objects, a background image that does not contain any moving objects needs to be estimated. For that purpose, a background model is built using a several number of successive frames.

- **Foreground detection.** After the background image is obtained, all subsequent frames are extracted from that model in order to find pixels that do not belong to the background and therefore can be considered as a foreground.
- **Background update.** During the detection process a background model must be updated with each new frame, because in real cases a background does not remain still throughout the whole video. The model must be adaptive for various changes

(e.g. some moving object does not move for a long time) so that they do not affect the result of detection.

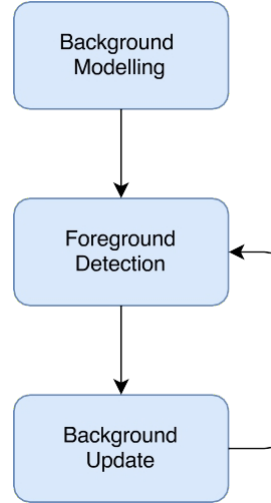


Fig 1.3. BS process

Different BS methods are usually classified by the way they initialize a background model and how they compute the distance between background and foreground pixels [17, 18]. In accordance with this type of classification, the most commonly used BS techniques are the following:

- **Basic Motion Detection.** A background is represented by a single image taken when there is no motion or estimated by a use of temporal median filter [19, 20]. Difference between foreground and background pixels is calculated on the basis of pixels' colors.
- **One Gaussian.** The idea of this approach is that background pixels are modelled with a probability density function (PDF) and the distribution is considered to be normal [21]. The parameters of PDF are estimated with a number of training frames. In this case, a pixel is assumed to be a foreground if it has low probability.
- **Gaussian Mixture Model (GMM).** In contrast to the previous method, GMM based methods make use of a several Gaussians in modeling process [22].
- **Kernel Density Estimation (KDE).** KDE is a non-parametric statistical approach, in which the PDF is estimated directly from the data [23].
- **Other methods.** There is a vast variety of different techniques that are based on, for example, fuzzy logic [24], neural networks [25], codebook [26] and eigenvalues and eigenvectors [27].

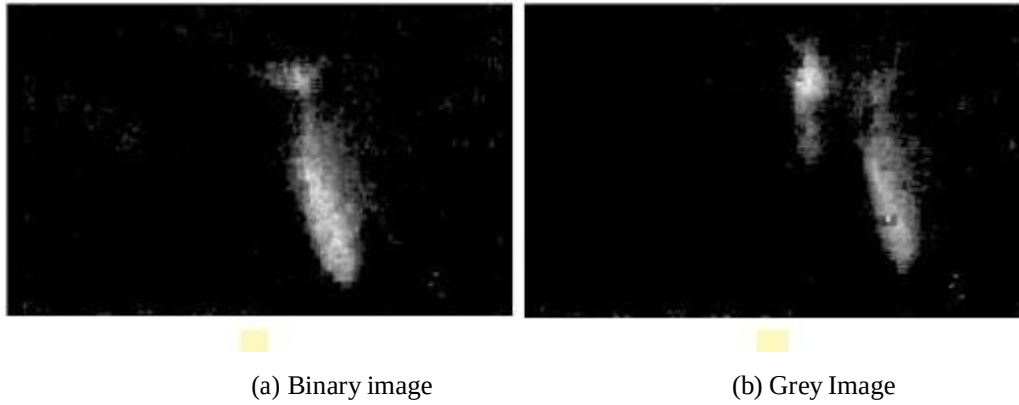
1.2 Fish Detection

Object detection algorithms have various applications. For instance, computer vision techniques provide a great opportunity to make animal monitoring more accurate, less time consuming and fully automated.

In particular, different approaches and methods of moving object detection have been used for the task of fish detection, which is the fundamental step in building a fish observation system. Palazzo et al. [28] proposed an approach for object detection that imply explicit modelling both the background and the foreground for each frame. This allows to avoid misdetections when the background is not stationary or when target objects have the same color as the background. The latter problem is also solved by introducing texture information in models as well as the color. The algorithm was evaluated on real underwater videos and showed high and stable performance.

In [29], an Expectation-maximization (EM) algorithm was proposed for fish detection. Given intensity images of fish (Figure 4), a certain threshold is selected, such that pixels with higher intensities are assumed to belong to the fish. Then the sets of x and y locations corresponding to these pixels are defined (Figure 5). **The shape of each fish is assumed to be a multivariate Gaussian, and then an image is modeled as GMM.**

And finally, the parameters of GMM, including the number of fish, are estimated using an EM algorithm. The method was tested on southern bluefin tuna fish, an individual fish was modelled by using a two-dimensional Gaussian distribution with x and y pixel locations as an input. Spampinato et al. [2] presented an automated video processing system for underwater video surveillance. The system is able to solve the tasks of texture and color analysis,



fish detection and fish tracking by using three image processing subsystems. The first one calculates the average texture and color properties of the frame. It considers the brightness, smoothness and identifies the green tone. Fish detection subsystem is based on a BS approach. It uses the combination of moving average algorithm with the Adaptive GMM [30] which shows high performance on both stable and dynamic background scenes. The last subsystem tracks the fish throughout the video. The proposed system was tested on 20 underwater video sequences and the results were considered to be excellent when compared to other similar methods. An example of the system outcome is shown in Figure 6.



Fig 1.4 Tracking example with two frames grabbed at two consecutive times [2].

Challenges in Fish Detection Despite the variety of the proposed approaches, it is a challenging task to find a robust and high performance method for fish detection. Usually video sequences may contain various attributes that make the task

more difficult. Some of the possible problems connected with the detection are listed below [10]:

- **Illumination changes.** During the video capturing the illumination may change gradually (e.g. intensity of the sun changes throughout the day) or suddenly (e.g. the lamp has been turned off), so the background model must take the light into account.
- **Dynamic background.** Often the background is not a stable scene and the algorithm should distinguish moving objects of interest from those that can be regarded as background.
- **Occlusion.** Object of interest cannot be fully seen and detected on some frames if it is occluded by other objects.
- **Clutter.** It hinders the process of background modelling and complicates the segmentation.
- **Camouflage.** Target objects may have the similar color and pattern as a background, which makes them hardly distinguished even by the human.
- **Camera motions.** If the camera is unstable it may affect the quality of the captured video.
- **Noise.** Video may contain noise which also affects the results.
- **Environmental issues.** The quality of water, underwater garbage, presence of moving plants, etc. may also hamper the fish detection.

In the case of fish detection, different types of videos are used for experiments (Figure 7). In low quality video, there is noise and fish do not differ sufficiently from the background in the terms of color. High quality video on the opposite allows all fish to be distinguished easily, but that may cause clutter and occlusions. Illumination changes and sudden appearance of glares are common attributes of underwater videos as well. This kind of troubles should be considered while choosing an appropriate detection method



Fig 1.5 Difficult cases for fish detection.

CHAPTER 2

LITERATURE SURVEY

2.1 Detecting, tracking and counting fish in low quality unconstrained underwater videos.

Concetto Spampinato, Yun-Heh Chen-Burger, Gayathri Nadarajan, and Robert B Fisher. *Detecting, tracking and counting fish in low quality unconstrained underwater videos. In Proceedings of the Third International Conference on Computer Vision Theory and Applications, volume 2, pages 514–519. Citeseer, 2008.*

In this work a machine vision system capable of analysing underwater videos for detecting, tracking and counting fish is presented. The real-time videos, collected near the Ken-Ding sub-tropical coral reef waters are managed by EcoGrid, Taiwan and are barely analysed by marine biologists. The video processing system consists of three subsystems: the video texture analysis, fish detection and tracking modules. Fish detection is based on two algorithms computed independently, whose results are combined in order to obtain a more accurate outcome. The tracking was carried out by the application of the CamShift algorithm that enables the tracking of objects whose numbers may vary over time. Unlike existing fish-counting methods, our approach provides a reliable method in which the fish number is computed in unconstrained environments and under several scenarios (murky water, algae on camera lens, moving plants, low contrast, etc.). The proposed approach was tested with 20 underwater videos (More)

2.2 Congruence between automatic fish counter data and rod catches of atlantic salmon Thorley, DMR Eatherley, AB Stephen, I Simpson, JC MacLean, and AF Youngson. *Congruence between automatic fish counter data and rod catches of atlantic salmon (salmo salar) in scottish rivers. ICES Journal of Marine Science: Journal du Conseil, 62(4):808–817, 2005.*

We used over 154 000 scale samples collected from salmon fisheries in the large River Teno system over a 40-year period to quantify life history diversity and long-term trends. We identified 120 different life history strategies, including combinations of smolt (2–8) and sea ages (1–5) and previous spawning events. Most strategies were rare; 60% of individuals matured after 1 year at sea following 3–5 years in fresh water. Age at maturity changed with an increase in two-sea-winter salmon and previous spawners and a decline in three-sea-winter fish. Smolt age distribution showed a decreasing proportion of age-3 smolts, while that of age-5 smolts increased. Fishing gear and fishing season times selected for fish differing in life history strategies. Temporal variation in life histories reflected changes in both fisheries and the changing environment. There was an inverse relationship between years spent in fresh water and sea age. Bio complexity was manifested by the multiple year classes (6–11) present in annual runs, which increased with years, reflecting an increase both in previous spawners and sampling effort. The high number of cohorts spawning simultaneously each year indicates strong generational overlap, which has been suggested to maintain genetic diversity and thereby resilience via the portfolio effect.

Bio-complexity in fish populations, i.e. their diversity and variation in life-history and genetic characteristics, provides resilience to disturbances by spreading risk temporally and spatially and thus contributes to long-term sustainability (Stearns 1992; Hilborn et al. 2003; Schindler et al. 2010). Sufficient diversity in populations is also a prerequisite for sustainable fisheries (Hilborn et al. 2003; Hutchinson 2008). Typically, diversity comprises variation in the age at maturity

Downloaded from www.nrcresearchpress.com by University of Chicago Library on 04/15/18 For personal use only.

This Just-IN manuscript is the accepted manuscript prior to copy editing and page composition. It may differ from the final official version of record. Individuals from the same cohort reproducing in different years – or iteroparity with the possibility of repeated spawning events for individual fish; both allocating the reproductive effort across multiple years and stabilizing populations under environmental variability (Schindler et al. 2010; Moore et al. 2014). It is therefore important to characterize the levels of, and changes in, diversity in exploited populations so that the potential importance of bio-complexity for sustaining productivity can be assessed. Such data also provide a baseline for assessing the potential effects of climate change. Anadromous salmonid fish exhibit especially wide variation in life histories among fish species, both in time spent in freshwater before the oceanic migration and in duration of the marine feeding migration and connected timing of return migration to freshwater for spawning (e.g. Klemetsen et al. 2003). The simplest set of different salmonid life histories is that of pink salmon (*Oncorhynchus gorbuscha*) and the most complex those of steelhead trout (*O. mykiss*; Thorpe 1998; Moore et al. 2014) and Atlantic salmon (*Salmo salar*; Klemetsen et al. 2003; Niemelä et al. 2006a).

2.3 Development of Hydroacoustic Methods for Fish Detection in Shallow Water.

Helge Balk. Development of Hydroacoustic Methods for Fish Detection in Shallow Water. PhD thesis, Faculty of Mathematics and Natural Science, University of Oslo, 2001.

The main objective of this thesis was measurement of fish side-aspect target strength, which was determined with simulations and under experimental conditions with both immobilized and swimming fish. In addition, target strength was measured in the natural environment of the fish. Furthermore, horizontal hydroacoustics was utilized to establish the effects of environmental factors on fish migration. Target-strength simulations with a swimbladder model demonstrated that the relationship between target strength and swimbladder length was not linear. Furthermore, with the concave bending of the swimbladder model, the interference between wavelets caused fluctuation in target strength, which could be over 15 dB. A single log-linear regression for target strength as a function of fish length was fitted.

The ex situ measurements of immobilized Atlantic salmon (*Salmo salar* L.) and other species showed that the linear relationship between mean target strength and the logarithm length of fish increases with a slope larger than 20. In addition, whitefish (*Coregonus lavaretus* (L.) s.l.) has a target strength that is about 5 dB higher than salmon of the same size. The variation in target strength with the side-aspect angle was very large. The best fit of the all-side-aspect models between mean target strength and side-aspect angle (α) was obtained using the $\cos^3(2\alpha)$ function. A statistical approach was utilized to predict target-strength distributions of salmon as they ascended a river. With a large number of fish, the large value for standard error of the estimate (S.E.E) in a log-linear regression for target strength and fish length smoothed out these distributions. The probability of detection of salmon with threshold target strength (-29 dB) increased as length of the fish increased. Few statistically significant correlations were observed between entry of salmon into the river and six environmental factors measured in the Tornionjoki River. In Äijälänsalmi Channel, migration of cyprinids and percids through the channel was triggered by the temperature difference between lakes, and migration occurred around dawn and dusk.

In fisheries research, one purpose of echo sounding is to estimate the density or number of aquatic organisms, especially fish (MacLennan & Simmonds 1992). The classification and identification of these organisms generally combines knowledge of the distribution and behavior patterns of constituent species with collection and analysis of acoustic and catch

data (Horne 2000). In echo-sounding systems, acoustic energy is transmitted in pulses at a particular frequency (usually 38 – 420 kHz) by means of a transducer producing a directional sound beam. When a sound wave encounters a density difference, e.g. a fish, an echo propagates outward from this target back to the receiver. Echoes returning to the transducer are said to be backscattered sound (Midttun 1984). Accurate interpretation of the acoustic data from fisheries surveys requires an understanding of the scattering properties of fish. Traditionally in fisheries acoustics, the acoustic beam has been aimed vertically.

However, the echo sounder can also be used to transmit the sound beam horizontally across water. During the past decade, a side-looking (horizontal), fixed location split-beam or dual-beam echo sounder has been used for assessment of fish stock in rivers and shallow channels (Kubecka 1996, Ransom et al. 1998). Use of an acoustic system in a river offers the advantage that it is made up of noninvasive and nondestructive devices for counting number of fish and observing the behavior of fish. In a fixed location, horizontal echo sounding has basically four different sources of echoes. These are echoes (a) from passing fish, (b) from drifting debris, (c) from stationary targets like bottom and stones, and (d) from noise or reverberation. Echoes from fish are naturally the main objective in fisheries acoustics; the rest are usually unwanted sources of echoes (Balk 2001). In rivers, the downstream migration of fish may be passive or active, but upstream migration must be active. Therefore, it can be assumed that all targets that are moving upstream are fish. Selection of sampling site is the most important factor for successful hydroacoustic monitoring in rivers.

2.4 A system for video surveillance and monitoring.

Robert T Collins, Alan J Lipton, Takeo Kanade, Hironobu Fujiyoshi, David Duggins, Yanghai Tsin, David Tolliver, Nobuyoshi Enomoto, Osamu Hasegawa, Peter Burt, et al. A system for video surveillance and monitoring. 2000.

The thrust of CMU research under the DARPA Video Surveillance and Monitoring (VSAM) project is cooperative multi-sensor surveillance to support battlefield awareness [17]. Under our VSAM Integrated Feasibility Demonstration (IFD) contract, we have developed automated video understanding technology that enables a single human operator to monitor activities over a complex area using a distributed network of active video sensors. The goal is to automatically collect and disseminate real-time information from the battlefield to improve the situational awareness of commanders and staff. Other military and federal law enforcement applications include providing perimeter security for troops, monitoring peace treaties or refugee movements from unmanned air vehicles, providing security for embassies or airports, and staking out suspected drug or terrorist hide-outs by collecting time-stamped pictures of everyone entering and exiting the building. Automated video surveillance is an important research area in the commercial sector as well.

Technology has reached a stage where mounting cameras to capture video imagery is cheap, but finding available human resources to sit and watch that imagery is expensive. Surveillance cameras are already prevalent in commercial establishments, with camera output being recorded to tapes that are either rewritten periodically or stored in video archives. After a crime occurs – a store is robbed or a car is stolen – investigators can go back after the fact to see what happened, but of course by then it is too late. What is needed is continuous 24-hour monitoring and analysis of video surveillance data to alert security officers to a burglary in progress, or to a suspicious individual loitering in the parking lot, while options are still open for avoiding the crime. Keeping track of people, vehicles, and their interactions in an urban or battlefield environment is a difficult task.

The role of VSAM video understanding technology in achieving this goal is to automatically “parse” people and vehicles from raw video, determine their geolocations, and insert them into a dynamic scene visualization. We have developed robust routines for detecting and tracking moving objects.

Detected objects are classified into semantic categories such as human, human group, car, and truck using shape and color analysis, and these labels are used to improve tracking using temporal consistency constraints. Further classification of

human activity, such as walking and running, has also been achieved. Geolocations of labeled entities are determined from their image coordinates using either wide-baseline stereo from two or more overlapping camera views, or intersection of viewing rays with a terrain model from monocular views. These computed locations feed into a higher level tracking module that tasks multiple sensors with variable pan, tilt and zoom to cooperatively and continuously track an object through the scene. All resulting object hypotheses from all sensors are transmitted as symbolic data packets back to a central operator control unit, where they are displayed on a graphical user interface to give a broad overview of scene activities. These technologies have been demonstrated through a series of yearly demos, using a testbed system developed on the urban campus of CMU.

2.5 Airborne traffic surveillance systems: video surveillance of highway traffic.

Suman Srinivasan, Haniph Latchman, John Shea, Tan Wong, and Janice McNair. Airborne traffic surveillance systems: video surveillance of highway traffic. In Proceedings of the ACM 2nd International Workshop on Video Surveillance & Sensor Networks, pages 131–135. ACM, 2004.

Timely information about highway traffic conditions is very important for the Department of Transportation (DOT) and other relevant agencies. Such live information would be very important when traffic incidents or accidents occur. An aerial view is the best for traffic situations, particularly over highways. Unmanned aircraft equipped with video cameras and/or other sensors may be able to deliver the necessary information through video images with relatively low operational costs and risks to human life. ATSS (Airborne Traffic Surveillance System), a project funded by the Florida Department of Transportation, attempts to make this vision a reality. This paper describes how the University of Florida research team implemented a system for ATSS from the ground up, using unmanned aerial vehicles, digital video encoding, and transmission of data and multimedia video streams over FDOT's microwave IP networks.

This paper provided a brief overview of the ATSS project funded by the FDOT and implemented by the UFL research team. Other than providing details on how video surveillance of state highways can be implemented using unmanned aerial vehicles and the existing microwave network, the paper also describes how the UFL research team developed in-house software that enabled the transmission of signal strengths over the network, and implemented automatic switching of highway traffic video using the signal strengths obtained from the video signal. We believe that the ATSS project is a ground-breaking project and could revolutionize highway traffic surveillance in the near future. This would enable the Department of Transportation and other emergency services to respond more quickly when accidents occur and for other operations that require urgent feedback. Some of the documents referenced in this paper are available for download from [17].

2.6 Automated detection and recognition of wildlife using thermal cameras.

Peter Christiansen, Kim Arild Steen, Rasmus Nyholm Jørgensen, and Henrik Karstoft. Automated detection and recognition of wildlife using thermal cameras. Sensors, 14(8):13778–13793, 2014.

In agricultural mowing operations, thousands of animals are injured or killed each year, due to the increased working widths and speeds of agricultural machinery. Detection and recognition of wildlife within the agricultural fields is important to reduce wildlife mortality and, thereby, promote wildlife-friendly farming. The work presented in this paper contributes to the automated detection and classification of animals in thermal imaging. The methods and results are based on top-view images taken manually from a lift to motivate work towards unmanned aerial vehicle-based detection and recognition. Hot objects are detected based on a threshold dynamically adjusted to each frame. For the classification of animals, we propose a novel thermal feature extraction algorithm. For each detected object, a thermal signature is calculated using morphological operations.

The thermal signature describes heat characteristics of objects and is partly invariant to translation, rotation, scale

and posture. The discrete cosine transform (DCT) is used to parameterize the thermal signature and, thereby, calculate a feature vector, which is used for subsequent classification. Using a k-nearest-neighbor (kNN) classifier, animals are discriminated from non-animals with a balanced classification accuracy of 84.7% in an altitude range of 3–10 m and an accuracy of 75.2% for an altitude range of 10–20 m. To incorporate temporal information in the classification, a tracking algorithm is proposed. Using temporal information improves the balanced classification accuracy to 93.3% in an altitude range 3–10 of meters and 77.7% in an altitude range of 10–20 m

We have introduced a method for the automatic detection and recognition of wildlife using thermal cameras for UAV technology. Based on a dynamic threshold, hot objects are detected and subsequent feature extraction is performed. The novel feature extraction method, presented in this paper, consist of an extraction of thermal signatures for each detected object and a parameterization of this based on DCT. Methods for classification using measurements from both single and multiple frames is presented. Combining measurements from multiple frames achieves the best performance, with a balanced classification accuracy of 93.5% in the altitude range of 3–10 m and 77.7% in the altitude range of 10–20 m, thus demonstrating a clear relationship between the performance of detection and classification relative to altitude. The simulated and limited dataset is favorable in terms of performance for the given algorithms. The actual applicability of the system should therefore be determined using footage from an actual UAV. The proposed detection and classification scheme is based on top-view images of wildlife, as seen by a UAV. The use of UAV-technology for automatic detection and recognition of wildlife is currently part of ongoing research towards wildlife-friendly agriculture.

2.7 Moving object detection and tracking in video surveillance system.

Kinjal A Joshi and Darshak G Thakore. A survey on moving object detection and tracking in video surveillance system. International Journal of Soft Computing and Engineering, 2(3):44–48, 2012.

Moving object detection and tracking are the more important and challenging task in video surveillance and computer vision applications. Object detection is the procedure of finding the non-stationary entities in the image sequences. Detection is the first step towards tracking the moving object in the video. Object representation is the next important step to track. Tracking is the method of identifying, the position of the moving object in the video. Identifying the position is much more challenging task then detecting the moving object in a video. Object tracking is applied in numerous applications like in robot vision, monitoring the traffic, Video surveillance, Video in-painting and Simulation. Here we are going to present a brief review of numerous object detection, object classification and object tracking algorithms available.

Video scrutiny is a more challenging task in today's environment. Tracking the moving object has attracted many researchers in the field of computer vision and image processing. Video surveillance is the process of observing the behaviour, events and other necessary evidence, typically of the people for supervision and protecting them. Surveillance is mainly used by governments for gathering information, for investigating and preventing the crime. Video surveillance activity can be divided into three types namely. Manual video surveillance, Semi-autonomous surveillance and Fully- autonomous surveillance. In manual surveillance human is responsible for analysing the contents of the video. Semi-autonomous surveillance system involves video processing by system with the interaction of the human when necessary. Fully-autonomous surveillance system, the system will perform every task such as detection of motion, tracking etc., without the need of human interaction. Automated monitoring procedure also known as the Intelligence Visual Surveillance (IVS) involves the scrutiny and interpretation of objects activities, in addition to object detection and tracking to recognize the visual actions of the scene. The main task of IVS consists of wide area surveillance control and scene interpretation. Object tracking must deal with several illumination changes and well-known challenges. Mainly video analysis is categorized into three basic phases: moving entity detecting, finding the trajectory of object from one frame to another frame and scrutiny of entity tracks to identify their performance. Tracking objects in a

static environment is much easier than tracking objects in dynamic environment. In common, in a dynamic ecological system together background and entity varies [1]. In principle, to resolve this common unconstrained problem is rigid. One can place a group. The rest of the paper is systematized as follows. Section II presents methodology in which object detection, representation and tracking Techniques is illustrated. Performance analysis is described in Section III. Conclusion about the paper is presented in Section IV.

2.8 Moving Object Detection Using Background Subtraction

Soharab Hossain Shaikh, Khalid Saeed, and Nabendu Chaki. *Moving Object Detection Using Background Subtraction*. Springer, 2014.

Background subtraction is a widely used approach for detecting moving objects from videos captured with static a camera. This chapter introduces the basic concept behind this approach using a simple frame differencing method. A survey on existing literature on this topic is also reported in this chapter.

CHAPTER 3

PROPOSED SYSTEM

The proposed system is to recognize an object in a noisy and dense condition. Live fish and water bottle is recognized and image count is also given. It has a better approach and morphology techniques done in Matlab to give accurate and efficient results.



Fig.3.1 Block Diagram for fish detection and counting from Image

Here we detected fish and counted from underwater images. Same procedure can be applied for detection and counting of fishes from video. Video is nothing but frames in sequence.

3.1 Underwater Images

The underwater image processing area has received considerable attention within the last decades, showing important achievements. In this paper we review some of the most recent methods that have been specifically developed for the underwater environment. These techniques are capable of extending the range of underwater imaging, improving image contrast and resolution. After considering the basic physics of the light propagation in the water medium, we focus on the different algorithms available in the literature.

3.2 Gray scale conversion

An RGB image can be viewed as three images(a red scale image, a green scale image and a blue scale image) stacked on top of each other. In MATLAB, an RGB image is basically a $M \times N \times 3$ array of colour pixel, where each colour pixel is a triplet which corresponds to red, blue and green colour component of RGB image at a specified spatial location. Similarly, A Grayscale image can be viewed as a single layered image. In MATLAB, a grayscale image is basically $M \times N$ array whose values have been scaled to represent intensities.

3.3 Edge detection

Edge detection includes a variety of mathematical methods that aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed edges. The same problem of finding discontinuities in one-dimensional signals is known as step detection and the problem of finding signal discontinuities over time is known as change detection. Edge detection is a fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction.

3.4 Morphological Operations

Morphological image processing is a collection of non-linear operations related to the shape or morphology of features in an image. According to Wikipedia, morphological operations rely only on the relative ordering of pixel values, not on their numerical values, and therefore are especially suited to the processing of binary images. Morphological operations can also be applied to greyscale images such that their light transfer functions are unknown and therefore their absolute pixel values are of no or minor interest. Morphological techniques probe an image with a small shape or template called a structuring element. The structuring element is positioned at all possible locations in the image and it is compared with the corresponding neighbourhood of pixels.

3.5 Applying Bounding Boxes to Image

In object detection, we usually use a bounding box to describe the target location. The bounding box is a rectangular box that can be determined by the x and y axis coordinates in the upper-left corner and the x and y axis coordinates in the lower-right corner of the rectangle. ²Another commonly used bounding box representation is the x and y axis coordinates of the bounding box center, and its width and height. Here we define functions to convert between these two representations, `box_corner_to_center` converts from the two-corner representation to the center-width-height presentation, and `box_center_to_corner` vice versa.

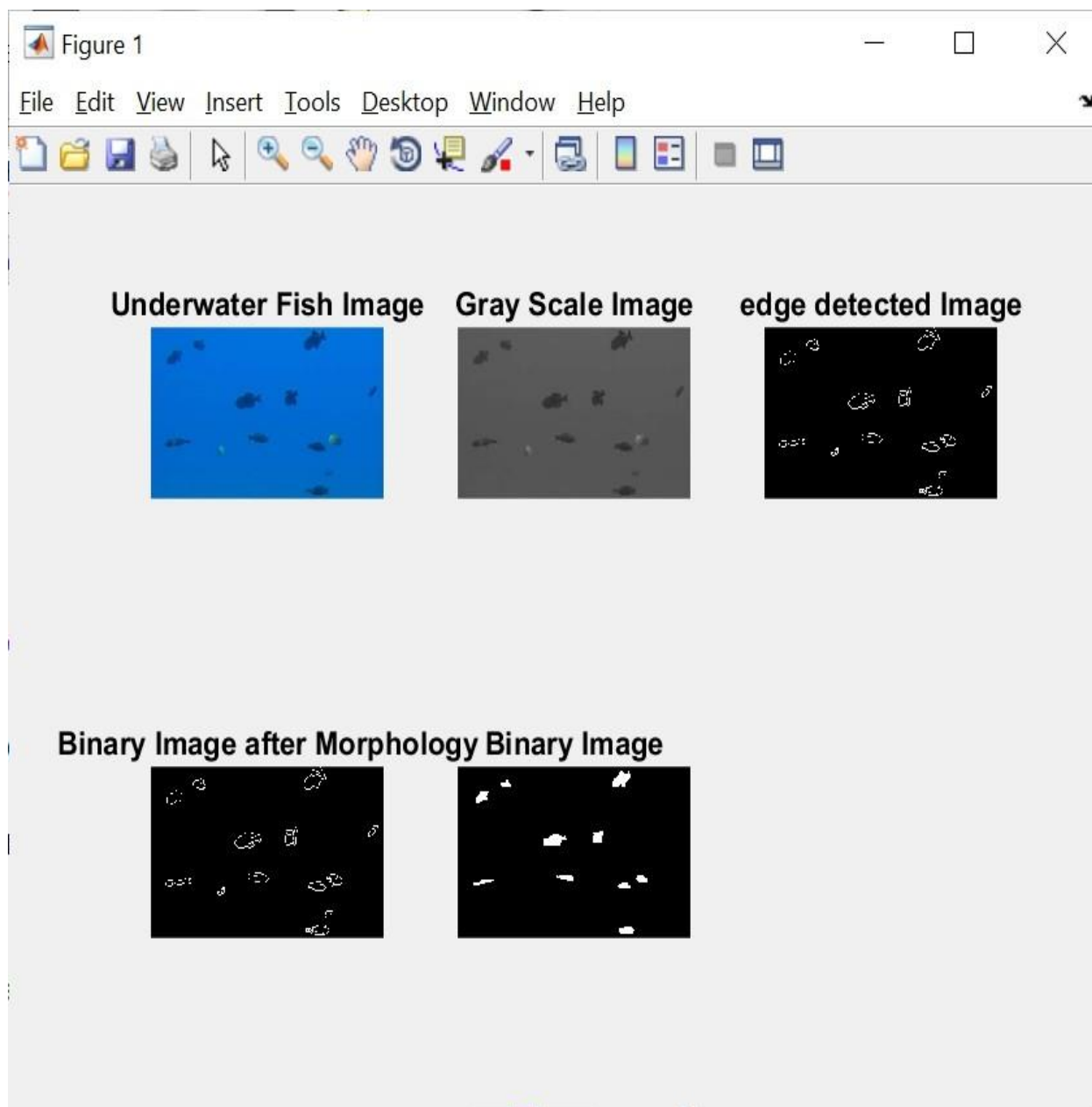


Fig.4.1 Fish Detection from underwater images

CHAPTER 4
Result
Analysis

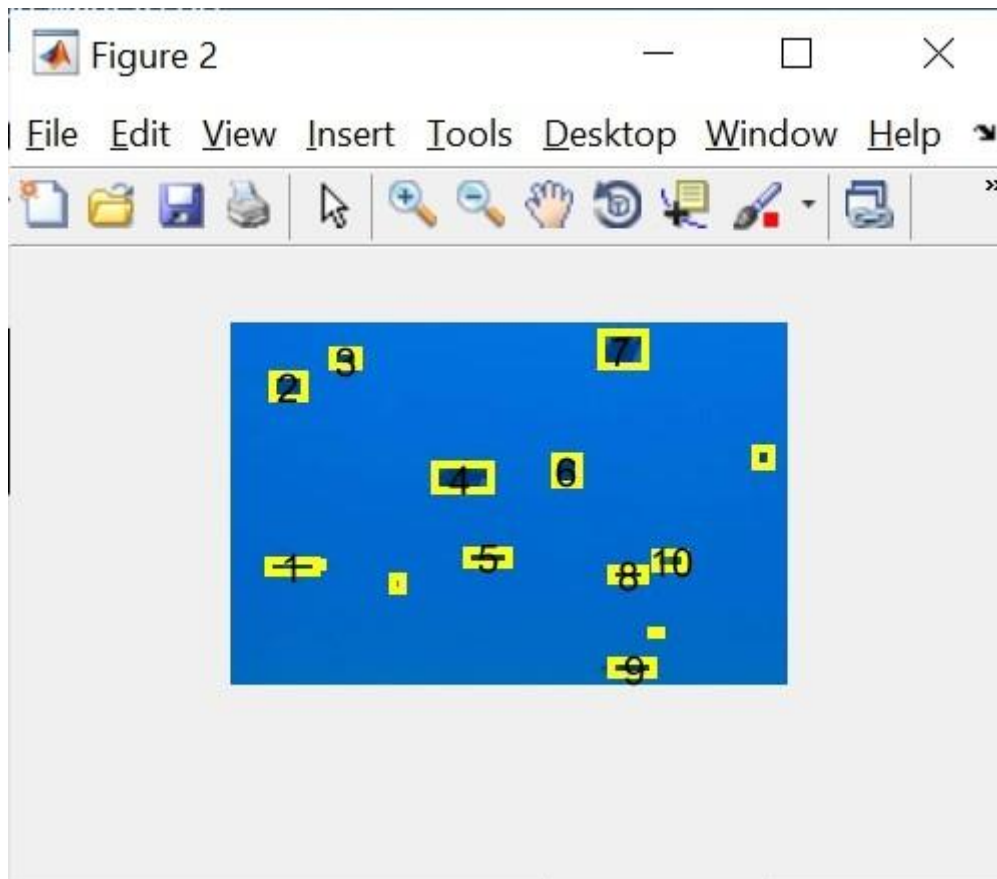


Fig.4.2 Detection and counting Number of Fishes (Applied Bounding Box)

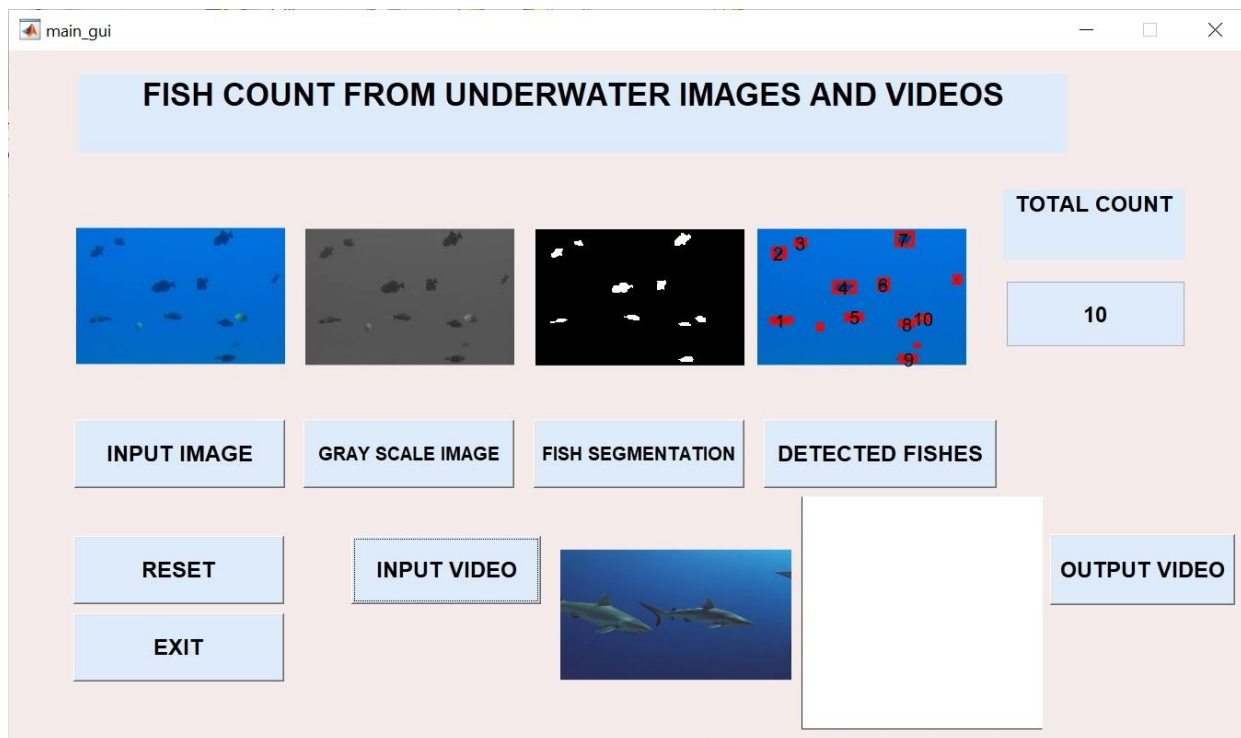


Fig.4.3 GUI selected input video

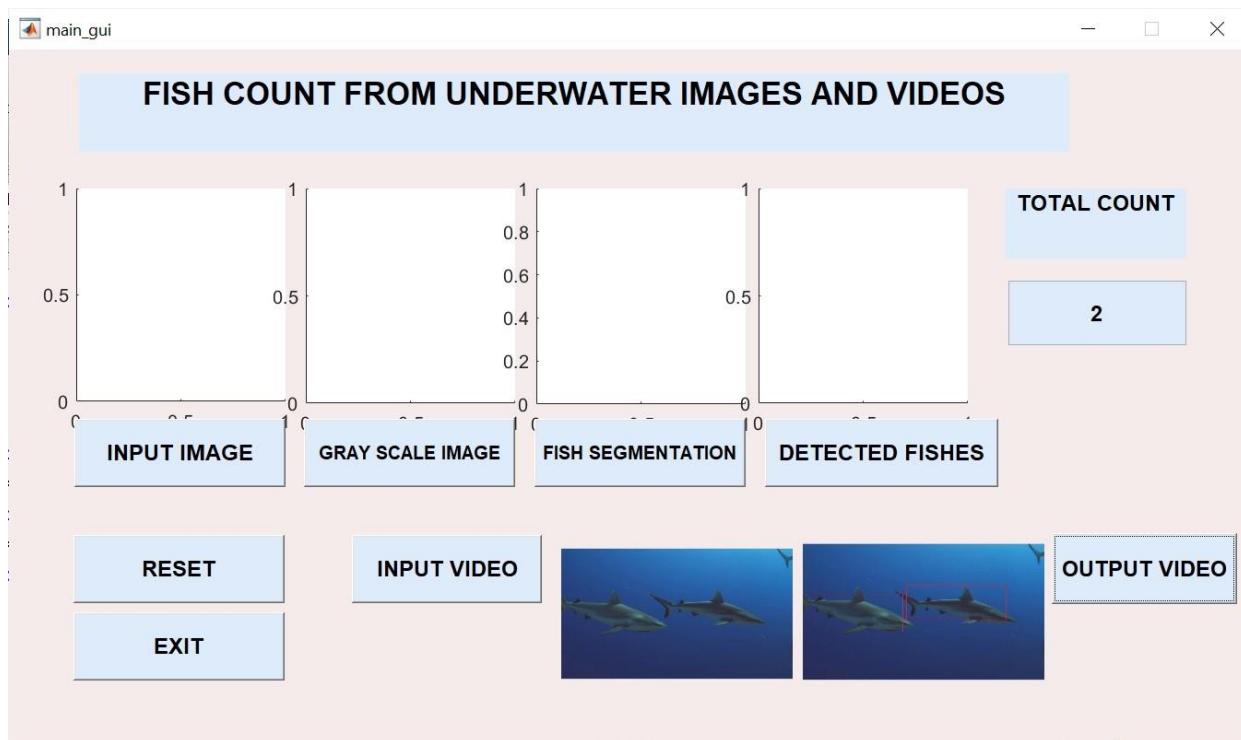


Fig.4.4 Detected Fishes from underwater video

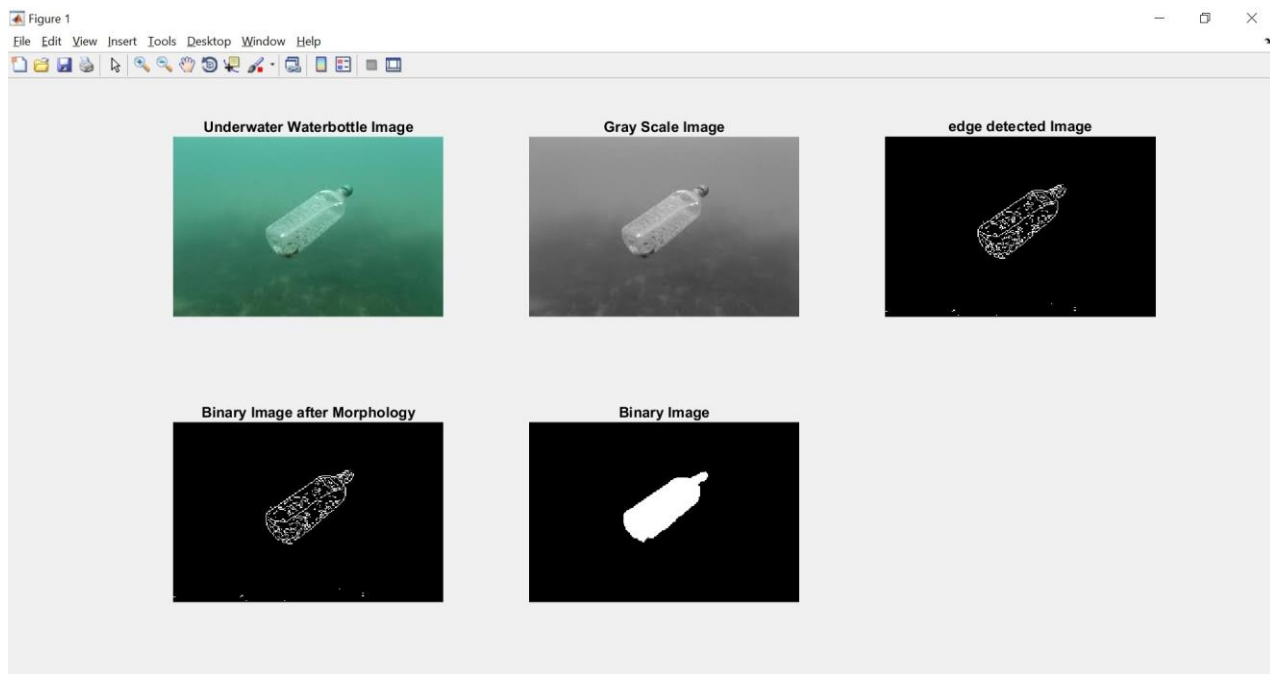


Fig.4.5 Underwater water bottle Image

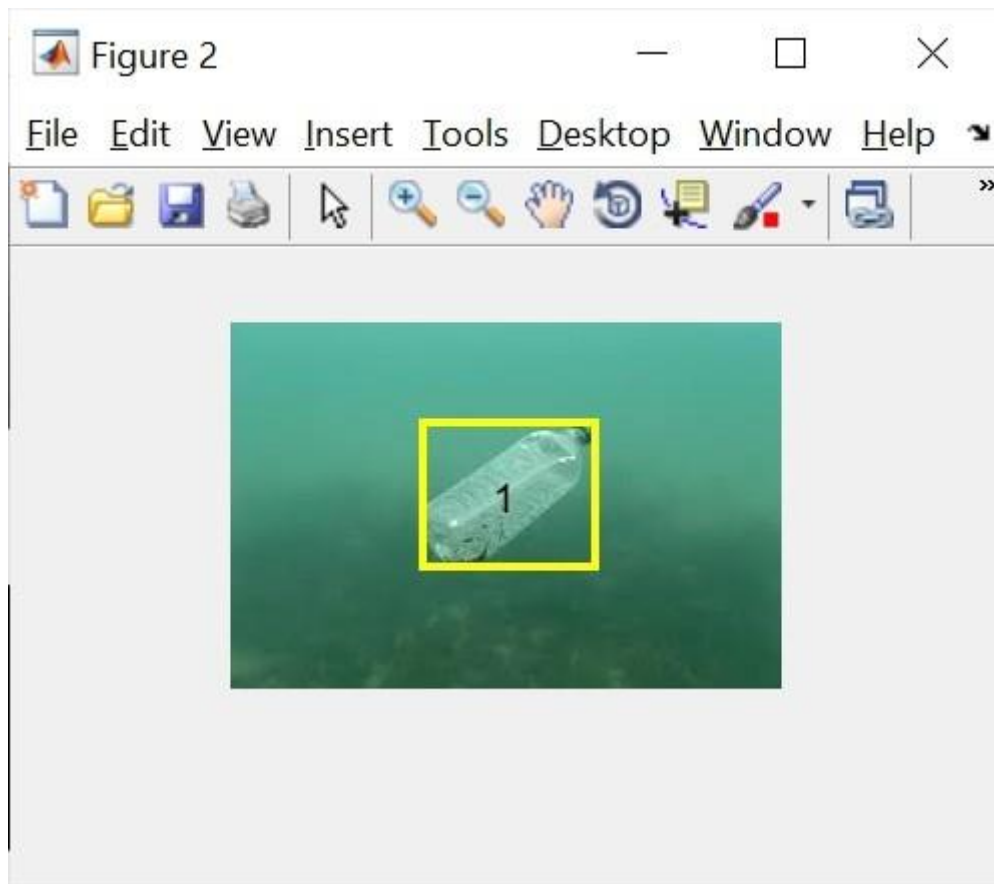


Fig.4.6 Detected water bottle from underwater image

5.1 Introduction

MATLAB is a excessive-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment wherein issues and answers are expressed in familiar mathematical notation. Typical uses encompass

5.2 Applications of MATLAB:

- Math and computation
- Algorithm development
- Data acquisition
- Modeling, simulation, and prototyping
- Data evaluation, exploration, and visualization
- Scientific and engineering pictures
- Application improvement, along with graphical consumer interface constructing.

5.3 MATLAB

MATLAB is a immoderate-overall performance language for technical computing. It integrates computation, visualization, and programming in an clean-to-use surroundings wherein troubles and answers are expressed in acquainted mathematical notation. MATLAB stands for matrix laboratory, and become written first off to offer smooth get entry to to matrix software application advanced by using LINPACK (linear system bundle) and EISPACK (Eigen system package deal deal deal) responsibilities.

MATLAB is consequently constructed on a basis of sophisticated matrix software program application in which the important element is array that does not require pre dimensioning which to resolve many technical computing issues, especially people with matrix and vector formulations, in a fragment of time. MATLAB talents a own family of programs specific answers referred to as toolboxes. Very important to most customers of MATLAB, toolboxes permit analyzing and utilizing specialised era. These are complete collections of MATLAB features (M-documents) that growth the MATLAB environment to remedy unique commands of problems. Areas in which toolboxes are available embody sign processing, manipulate device, neural networks, fuzzy appropriate judgment, wavelets, simulation and plenty of others.

Typical uses of MATLAB encompass: Math and computation, Algorithm improvement, Data acquisition, Modeling, simulation, prototyping, Data evaluation, exploration, visualization, Scientific and engineering pics, Application development, along with graphical character interface constructing. Boom the MATLAB environment to remedy unique commands of problems. Areas wherein toolboxes are to be had embody signal processing, control device, neural networks, fuzzy appropriate judgment, wavelets, simulation and lots of others.

Typical uses of MATLAB include: Math and computation, Algorithm improvement, Data acquisition, Modeling, simulation, prototyping, Data analysis, exploration, visualization, Scientific and engineering photos, Application development, along side graphical character interface constructing.

5.4 Basic Building Blocks of MATLAB

The fundamental building block of MATLAB is MATRIX. The essential information type is the array. Vectors, scalars, real matrices and complicated matrix are dealt with as particular magnificence of this fundamental information kind. The built in

functions are optimized for vector operations. No size statements are required for vectors or arrays.

5.4.1 MATLAB Window

The MATLAB works based totally on five windows: Command window, Workspace window, Current directory window, Command history window, Editor Window, Graphics window and Online-help window.

5.4.2 Command Window

The command window is in which the person sorts MATLAB instructions and expressions on the spark off (>>) and in which the output of those commands is displayed. It is opened when the utility program is released. All instructions inclusive of user-written applications are typed in this window at MATLAB activate for execution.

5.4.3 Work Space Window

MATLAB defines the workspace as the set of variables that the person creates in a piece consultation. The workspace browser indicates these variables and a few records about them. Double clicking on a variable inside the workspace browser launches the Array Editor, which can be used to gain facts.

5.4.4 Three Current Directory Window

The present day Directory tab suggests the contents of the present day listing, whose route is hooked up inside the present day listing window. For example, within the domestic windows working tool the path is probably as follows: C:\MATLABWork, indicating that directory “paintings” is a subdirectory of the precept list “MATLAB”; it's far mounted in pressure C. Clicking at the arrow in the contemporary listing window shows a listing of currently used paths. MATLAB uses a are in search of for route to discover M-files and different MATLAB related documents. Any document run in MATLAB need to live inside the modern listing or in a list that is on trying to find path.

5.4.5 Four Command History Window

The Command History Window consists of a file of the instructions someone has entered within the command window, which includes each modern and previous MATLAB periods. Previously entered MATLAB instructions may be selected and re-accomplished from the command records window through right clicking on a command or collection of commands. This is beneficial to pick out various alternatives further to executing the instructions and is useful characteristic while experimenting with numerous commands in a piece consultation.

5.4.6 Editor Window

The MATLAB editor is every a text editor specialized for growing M-documents and a graphical MATLAB debugger. The editor can seem in a window through the use of itself, or it may be a sub window inside the desktop. In this window you can clearly write, edit, create and preserve applications in documents called M-files.

MATLAB editor window has numerous pull-down menus for responsibilities collectively with saving, viewing, and debugging documents. Because it performs a few smooth assessments and additionally makes use of shade to differentiate among severa factors of code, this newsletter editor is recommended because the device of desire for writing and improving M-functions.

5.4.7 Graphics or Figure Window

The output of all photo instructions typed inside the command window is seen in this window.

5.4.8 Online Help Window

MATLAB presents online assist for all its constructed in features and programming language constructs. The primary manner to get help on-line is to apply the MATLAB assist browser, opened as a separate window each with the useful resource of clicking on the question mark image (?) at the desktop toolbar, or with the aid of typing help browser at the activate inside the command window. The assist Browser is a web browser blanketed into the MATLAB computing tool that shows a Hypertext Markup Language (HTML) documents. The Help Browser includes planes, the assist navigator pane, used to find out statistics, and the display pane, used to view the statistics. Self-explanatory tabs other than navigator pane are used to carry out a are seeking.

5.5 MATLAB Files

MATLAB has 3 varieties of files for storing facts. They are: M-documents and MAT-files.

A.Three.1 M-Files

These are widespread ASCII text record with 'm' extension to the document call and growing own matrices using M-files, which are text documents containing MATLAB code. MATLAB editor or any other text editor is used to create a file containing the same statements which are typed at the MATLAB command line and save the record underneath a call that leads to .M. There are sorts of M-documents:

1. Script Files

It is an M-file with a hard and fast of MATLAB commands in it and is achieved by way of typing call of record on the command line. These documents paintings on worldwide variables presently found in that surroundings.

2. Function Files

A characteristic report is also an M-file besides that the variables in a feature record are all local. This type of files starts with a characteristic definition line.

5.6 MAT-Files

These are binary information documents with .Mat extension to the record which can be created through MATLAB when the records is saved. The information written in a unique format that only MATLAB can examine. These are located into MATLAB with 'load' command.

5.7 The MATLAB System:

The MATLAB machine consists of five important components:

5.8 Development Environment:

This is the set of tools and centers that assist you use MATLAB functions and files. Many of those gear are graphical person interfaces. It consists of the MATLAB computer and Command Window, a command history, an editor and debugger, and browsers for viewing assist, the workspace, files, and the quest path.

5.9 MATLAB Mathematical Function:

This is a large series of computational algorithms ranging from elementary features like sum, sine, cosine, and complex mathematics, to greater sophisticated features like matrix inverse, matrix eigen values, Bessel functions, and rapid Fourier transforms.

5.10 MATLAB Language:

This is a excessive-stage matrix/array language with manage glide statements, capabilities, statistics systems,

input/output, and item-orientated programming features. It lets in both "programming in the small" to rapidly create quick and dirty throw-away packages, and "programming within the massive" to create complete big and complicated software applications.

5.11 Four Graphics:

MATLAB has large centers for displaying vectors and matrices as graphs, as well as annotating and printing these graphs. It consists of high-degree features for two-dimensional and 3-dimensional statistics visualization, picture processing, animation, and presentation pics. It also includes low-level functions that let you fully customize the appearance of photographs as well as to build complete graphical consumer interfaces to your MATLAB packages.

5.12 MATLAB Application Program Interface (API):

This is a library that allows you to put in writing C and FORTRAN packages that have interaction with MATLAB. It consists of centers for calling exercises from MATLAB (dynamic linking), calling MATLAB as a computational engine, and for reading and writing MAT-files.

Demo demonstrates what is possible in **Mat lab**

Who lists all of the variables in your Mat lab workspace?

Whose list the variables and describes their matrix size

clear erases variables and functions from memory

clear x erases the matrix 'x' from your workspace

close by itself, closes the current figure window

figure creates an empty figure window

hold on holds the current plot and all axis properties so that subsequent graphing

commands add to the existing graph

hold off sets the next plot property of the current axes to "replace"

findfind indices of nonzero elements e.g.:

d = find(x>100) returns the indices of the vector x that are greater than 100

break terminate execution of m-file or WHILE or FOR loop

5.13 BASIC PLOT COMMANDS:

Kinds of plots:

plot(x,y) creates a Cartesian plot of the vectors x & y

plot(y) creates a plot of y vs. The numerical values of the factors inside the y-vector

semilogx(x,y) plots log(x) vs y

semilogy(x,y) plots x vs log(y)

loglog(x,y) plots log(x) vs log(y)

polar(theta,r) creates a polar plot of the vectors r & theta wherein theta is in radians

bar(x) creates a bar graph of the vector x. (Note also the command stairs(x))

`bar(x, y)` creates a bar-graph of the elements of the vector `y`, finding the bars

consistent with the vector factors of 'x'

Plot description:`grid` creates a grid at the images plot

5.14 ALGEBRIC OPERATIONS IN MATLAB:

Scalar Calculations:

`+` Addition

`-` Subtraction

`*` Multiplication

`/` Right department (a/b approach $a \div b$)

left division (a manner $b \div a$)

`^` Exponentiation

For instance `three*four` carried out in 'matlab' gives `ans=12`

`4/five` offers `ans=0.Eight`

Array products: Recall that addition and subtraction of matrices involved addition or subtraction of the character factors of the matrices. Sometimes it's far favored to definitely multiply or divide every element of an matrix by way of the corresponding element of any other matrix 'array operations'.

Array or detail-by way of-element operations are accomplished when the operator is preceded via a '.' (Period):

`a.*b` multiplies each detail of `a` by using the respective element of `b`

`a./b` divides each element of `a` by using the respective detail of `b`

`a.B` divides every detail of `b` by using the respective detail of `a`

`a.^b` raise each detail of `a` with the aid of the respective `b` element

5.15 MATLAB WORKING ENVIRONMENT:

5.15.1 MATLAB DESKTOP

Matlab Desktop is the principle Matlab software program window. The laptop includes 5 sub domestic home windows, the

command window, the workspace browser, the current list window, the command history window, and one or greater determine home windows, which can be tested most effective when the individual displays a photo.

The command window is in which the person kinds MATLAB instructions and expressions at the spark off (>>) and wherein the output of those commands is displayed. MATLAB defines the workspace as the set of variables that the patron creates in a work session. The cutting-edge Directory tab above the workspace tab suggests the contents of the present day listing, whose path is shown in the current list window.

For example, in the domestic home windows running device the course is probably as follows: C:\MATLAB\Work, indicating that listing "art work" is a subdirectory of the primary listing "MATLAB"; WHICH IS INSTALLED IN DRIVE C. Clicking at the arrow within the modern list window indicates a list of these days used paths. Clicking at the button to the proper of the window permits the user to change the modern-day listing. MATLAB uses a are seeking for route to locate M-files and other MATLAB related files, which is probably arrange in directories within the laptop report system. Any document run in MATLAB want to are living inside the cutting-edge list or in a list this is on are seeking route. By default, the documents supplied with MATLAB and math works toolboxes are protected within the are looking for direction.

The simplest manner to look which directories are soon the quest path, or to characteristic or alter a seek direction, is to choose set path from the File menu the laptop, after which use the set direction communication box. It is right exercise to feature any normally used directories to the search course to avoid time and again having the change the cutting-edge list.

The Command History Window includes a file of the commands a consumer has entered in the command window, along with every modern and previous MATLAB periods. Previously entered MATLAB commands can be determined on and re-accomplished from the command history window by the use of right clicking on a command or sequence of commands.

This motion launches a menu from which to choose out numerous options further to executing the instructions. This is useful to choose out severa options in addition to executing the instructions. This is a beneficial feature at the same time as experimenting with various commands in a piece consultation.

5.15.2 MATLAB Editor to create M-File:

The MATLAB editor is each a textual content editor specialized for growing M-files and a graphical MATLAB debugger. The editor can appear in a window via itself, or it could be a sub window within the laptop. M-files are denoted by using the usage of the extension .M, as in pixelThe MATLAB editor window has numerous pull-down menus for tasks together with saving, viewing, and debugging files. Because it plays a few simple assessments and also makes use of coloration to distinguish among severa elements of code, this newsletter editor is recommended due to the fact the tool of desire for writing and modifying M-abilities.

To open the editor , kind edit at the prompt opens the M-record filename.M in an editor window, equipped for enhancing. As stated earlier, the report must be inside the modern-day listing, or in a listing within the are looking for route.

5.16 Three Getting Help:

The most critical manner to get assist online is to use the MATLAB help browser, opened as a separate window either

through clicking on the question mark symbol (?) at the computer toolbar, or through typing help browser on the set off in the command window. The help Browser is an internet browser incorporated into the MATLAB laptop that presentations a Hypertext Markup Language(HTML) files. The Help Browser includes two panes, the help navigator pane, used to find records, and the display pane, used to view the statistics. Self-explanatory tabs apart from navigator pane are used to carry out a search.

CHAPTER 6

PROCESSING

INTRODUCTION TO DIGITAL IMAGE

6.1 Digital Image Processing

An image may be defined as a two-dimensional function $f(x, y)$, where x & y are spatial coordinates, & the amplitude of at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y & the amplitude values of f are all finite discrete quantities, we call the image a digital image. The field of DIP refers to processing digital image by means of digital computer. Digital image is composed of a finite number of elements, each of which has a particular location & value. The elements are called pixels.

Vision is the most advanced of our sensor, so it is not surprising that image play the single most important role in human perception. However, unlike humans, who are limited to the visual band of the EM spectrum imaging machines cover almost the entire EM spectrum, ranging from gamma to radio waves. They can operate also on images generated by sources that humans are not accustomed to associating with image.

There is no general agreement among authors regarding where image processing stops & other related areas such as image analysis & computer vision start. Sometimes a distinction is made by defining image processing as a discipline in which both the input & output at a process are images. This is limiting & somewhat artificial boundary. The area of image analysis (image understanding) is in between image processing & computer vision.

There are no clear-cut boundaries in the continuum from image processing at one end to complete vision at the other. However, one useful paradigm is to consider three types of computerized processes in this continuum: low-, mid-, & high-level processes. Low-level process involves primitive operations such as image processing to reduce noise, contrast

enhancement & image sharpening. A low-level process is characterized by the fact that both its inputs & outputs are images. Mid-level process on images involves tasks such as segmentation, description of that object to reduce them to a form suitable for computer processing & classification of individual objects. A mid-level process is characterized by the fact that its inputs generally are images but its outputs are attributes extracted from those images. Finally higher-level processing involves “Making sense” of an ensemble of recognized objects, as in image analysis & at the far end of the continuum performing the cognitive functions normally associated with human vision.

Digital image processing, as already defined is used successfully in a broad range of areas of exceptional social & economic value.

6.2 Types of Images

An image is represented as a two dimensional function $f(x, y)$ where x and y are spatial co-ordinates and the amplitude of ‘ f ’ at any pair of coordinates (x, y) is called the intensity of the image at that point.

Gray scale image:

A grayscale image is a function $I(x, y)$ of the two spatial coordinates of the image plane.

$I(x, y)$ is the intensity of the image at the point (x, y) on the image plane.

$I(x, y)$ takes non-negative values assume the image is bounded by a rectangle $[0, a] \times [0, b]$: $[0, a] \times [0, b] \rightarrow [0, \text{info})$

Color image:

It can be represented by three functions, $R(x, y)$ for red, $G(x, y)$ for green and $B(x, y)$ for blue.

An image may be continuous with respect to the x and y coordinates and also in amplitude. Converting such an image to digital form requires that the coordinates as well as the amplitude to be digitized. Digitizing the coordinate’s values is called sampling. Digitizing the amplitude values is called quantization.

6.3 Coordinate convention:

The result of sampling and quantization is a matrix of real numbers. We use two principal ways to represent digital images. Assume that an image $f(x, y)$ is sampled so that the resulting image has M rows and N columns. We say that the image is of size $M \times N$. The values of the coordinates (x, y) are discrete quantities. For notational clarity and convenience, we use integer values for these discrete coordinates. In many image processing books, the image origin is defined to be at $(x, y) = (0, 0)$. The next coordinate values along the first row of the image are $(x, y) = (0, 1)$. It is important to keep in mind that the notation $(0, 1)$ is used to signify the second sample along the first row. It does not mean that these are the actual values of physical coordinates when the image was sampled. Following figure shows the coordinate convention. Note that x ranges from 0 to $M-1$ and y from 0 to $N-1$ in integer increments.

The coordinate convention used in the toolbox to denote arrays is different from the preceding paragraph in two minor ways. First, instead of using (x, y) the toolbox uses the notation (r, c) to indicate rows and columns. Note, however, that the order of coordinates is the same as the order discussed in the previous paragraph, in the sense that the first element of a coordinate tuple, (r, c) , refers to a row and the second to a column. The other difference is that the origin of the coordinate system is at $(r, c) = (1, 1)$; thus, r ranges from 1 to M and c from 1 to N in integer increments. IPT documentation refers to the coordinates. Less frequently the toolbox also employs another coordinate convention called spatial coordinates which uses x to refer to columns and y to refer to rows. This is the opposite of our use of variables x and y .

6.4 Image as Matrices:

The preceding discussion leads to the following representation for a digitized image function:

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix}$$

The right side of this equation is a digital image by definition. Each element of this array is called an image element, picture element, pixel or pel. The terms image and pixel are used throughout the rest of our discussions to denote a digital image and its elements.

A digital image can be represented naturally as a MATLAB matrix:

$$\begin{bmatrix} f(1,1) & f(1,2) & \dots & f(1,N) \\ f(2,1) & f(2,2) & \dots & f(2,N) \\ \vdots & \vdots & \ddots & \vdots \\ f(M,1) & f(M,2) & \dots & f(M,N) \end{bmatrix}$$

Where $f(1,1) = f(0,0)$ (note the use of a monospace font to denote MATLAB quantities). Clearly the two representations are identical, except for the shift in origin. The notation $f(p,q)$ denotes the element located in row p and the column q . For example $f(6,2)$ is the element in the sixth row and second column of the matrix f ; typically we use the letters M and N respectively to denote the number of rows and columns in a matrix. A $1 \times N$ matrix is called a row vector whereas an $M \times 1$ matrix is called a column vector. A 1×1 matrix is a scalar.

Matrices in MATLAB are stored in variables with names such as A , a , RGB , $real$ array and so on. Variables must begin with a letter and contain only letters, numerals and underscores. As noted in the previous paragraph, all MATLAB quantities are written using monospace characters. We use conventional Roman, italic notation such as $f(x,y)$, for mathematical expressions

6.5 Reading Images:

Images are read into the MATLAB environment using function `imread` whose syntax is

`imread('filename')`

Format name	Description	recognized extension
-------------	-------------	----------------------

TIFF	Tagged Image File Format	.tif, .tiff
JPEG	Joint Photograph Experts Group	.jpg, .jpeg
GIF	Graphics Interchange Format	.gif
BMP	Windows Bitmap	.bmp
PNG	Portable Network Graphics	.png
XWD	X Window Dump	.xwd

Here filename is a string containing the complete of the image file(including any applicable extension).For example the command line

```
>> f = imread ('8.jpg');
```

reads the JPEG (above table) image chestxray into image array f. Note the use of single quotes (') to delimit the string filename. The semicolon at the end of a command line is used by MATLAB for suppressing output. If a semicolon is not included. MATLAB displays the results of the operation(s) specified in that line. The prompt symbol(>>) designates the beginning of a command line, as it appears in the MATLAB command window.

When as in the preceding command line no path is included in filename, imread reads the file from the current directory and if that fails it tries to find the file in the MATLAB search path. The simplest way to read an image from a specified directory is to include a full or relative path to that directory in filename.

For example,

```
>> f = imread ( 'D:\myimages\chestxray.jpg');
```

reads the image from a folder called my images on the D: drive, whereas

```
>> f = imread(' . \ myimages\chestxray .jpg');
```

reads the image from the my images subdirectory of the current of the current working directory. The current directory window on the MATLAB desktop toolbar displays MATLAB's current working directory and provides a simple, manual way to change it. Above table lists some of the most of the popular image/graphics formats supported by imread and imwrite.

Function size gives the row and column dimensions of an image:

```
>> size (f)
```

```
ans = 1024 * 1024
```

This function is particularly useful in programming when used in the following form to determine automatically the size of an image:

```
>>[M,N]=size(f);
```

This syntax returns the number of rows(M) and columns(N) in the image.

The whole function displays additional information about an array. For instance ,the statement

```
>> whos f
```

gives

Name	size	Bytes	Class
F	1024*1024	1048576	unit8 array

Grand total is 1048576 elements using 1048576 bytes

The unit8 entry shown refers to one of several MATLAB data classes. A semicolon at the end of a whose line has no effect ,so normally one is not used.

6.6 Displaying Images:

Images are displayed on the MATLAB desktop using function imshow, which has the basic syntax:

```
imshow(f,g)
```

Where f is an image array, and g is the number of intensity levels used to display it. If g is omitted ,it defaults to 256 levels .using the syntax

```
Imshow (f, {low high})
```

Displays as black all values less than or equal to low and as white all values greater than or equal to high. The values in between are displayed as intermediate intensity values using the default number of levels .Finally the syntax

```
Imshow(f,[ ]) )
```

Sets variable low to the minimum value of array f and high to its maximum value. This form of imshow is useful for displaying images that have a low dynamic range or that have positive and negative values.

Function pixval is used frequently to display the intensity values of individual pixels interactively. This function displays a cursor overlaid on an image. As the cursor is moved over the image with the mouse the coordinates of the cursor position and the corresponding intensity values are shown on a display that appears below the figure window .When working

with color images, the coordinates as well as the red, green and blue components are displayed. If the left button on the mouse is clicked and then held pressed, pixval displays the Euclidean distance between the initial and current cursor locations.

The syntax form of interest here is Pixval which shows the cursor on the last image displayed. Clicking the X button on the cursor window turns it off.

The following statements read from disk an image called rose_512.tif extract basic information about the image and display it using imshow :

```
>>f=imread('rose_512.tif');
```

```
>>whos f
```

Name	Size	Bytes	Class
F	512*512	262144	unit8 array

Grand total is 262144 elements using 262144 bytes

```
>>imshow(f)
```

A semicolon at the end of an imshow line has no effect, so normally one is not used. If another image,g, is displayed using imshow, MATLAB replaces the image in the screen with the new image. To keep the first image and output a second image, we use function figure as follows:

```
>>figure ,imshow(g)
```

Using the statement

```
>>imshow(f),figure ,imshow(g) displays both images.
```

Note that more than one command can be written on a line ,as long as different commands are properly delimited by commas or semicolons. As mentioned earlier, a semicolon is used whenever it is desired to suppress screen outputs from a command line.

Suppose that we have just read an image h and find that using imshow produces the image. It is clear that this image has a low dynamic range, which can be remedied for display purposes by using the statement.

```
>>imshow(h,[ ])
```

6.7 WRITING IMAGES:

Images are written to disk using function imwrite, which has the following basic syntax:

`Imwrite (f,'filename')`

With this syntax, the string contained in filename must include a recognized file format extension .Alternatively, the desired format can be specified explicitly with a third input argument. `>>imwrite(f,'patient10_run1','tif')`

Or alternatively

For example the following command writes f to a TIFF file named patient10_run1:

```
>>imwrite(f,'patient10_run1.tif')
```

If filename contains no path information, then imwrite saves the file in the current working directory.

The imwrite function can have other parameters depending on the file format selected. Most of the work in the following deals either with JPEG or TIFF images ,so we focus attention here on these two formats.

More general imwrite syntax applicable only to JPEG images is

```
imwrite(f,'filename.jpg','quality',q)
```

where q is an integer between 0 and 100(the lower number the higher the degradation due to JPEG compression).

For example, for q=25 the applicable syntax is

```
>> imwrite(f,'bubbles25.jpg','quality',25)
```

The image for q=15 has false contouring that is barely visible, but this effect becomes quite pronounced for q=5 and q=0.Thus, an expectable solution with some margin for error is to compress the images with q=25.In order to get an idea of the compression achieved and to obtain other image file details, we can use function imfinfo which has syntax.

`Imfinfo filename`

Here filename is the complete file name of the image stored in disk.

For example,

```
>> imfinfo bubbles25.jpg
```

outputs the following information(note that some fields contain no information in this case):

Filename: 'bubbles25.jpg'

FileModDate: '04-jan-2003 12:31:26'

FileSize: 13849

Format: 'jpg'

Format Version: ' '

Width: 714

Height: 682

Bit Depth: 8

Color Depth:

'graysca

le'Format Signature:

' '

Comment: { }

Where file size is in bytes. The number of bytes in the original image is corrupted simply by multiplying width by height by bit depth and dividing the result by 8. The result is 486948. Dividing this file size gives the compression ratio: $(486948/13849)=35.16$. This compression ratio was achieved. While maintaining image quality consistent with the requirements of the appearance. In addition to the obvious advantages in storage space, this reduction allows the transmission of approximately 35 times the amount of un compressed data per unit time.

The information fields displayed by `imfinfo` can be captured in to a so called structure variable that can be for subsequent computations. Using the receding an example and assigning the name `K` to the structure variable.

We use the syntax `>>K=imfinfo('bubbles25.jpg');`

To store in to variable `K` all the information generated by command `imfinfo`, the information generated by `imfinfo` is appended to the structure variable by means of fields, separated from `K` by a dot. For example, the image height and width are now stored in structure fields `K.Height` and `K.width`.

As an illustration, consider the following use of structure variable `K` to commute the compression ratio for `bubbles25.jpg`:

```
>> K=imfinfo('bubbles25.jpg');
```

```
>> image_bytes =K.Width* K.Height* K.Bit Depth /8;
```

```
>> Compressed_bytes = K.FileSize;
```

```
>> Compression_ratio=35.162
```

Note that `iminfo` was used in two different ways. The first was to type `iminfo bubbles25.jpg` at the prompt, which

resulted in the information being displayed on the screen. The second was to type `K=imfinfo('bubbles25.jpg')`, which resulted in the information generated by `imfinfo` being stored in `K`. These two different ways of calling `imfinfo` are an example of command_ function duality, an important concept that is explained in more detail in the MATLAB online documentation.

More general `imwrite` syntax applicable only to `tif` images has the form

```
Imwrite(g,'filename.tif','compression','parameter',....'resloution',[colres rows] )
```

Where 'parameter' can have one of the following principal values: 'none' indicates no compression, 'pack bits' indicates pack bits compression (the default for non 'binary images') and 'ccitt' indicates ccitt compression. (the default for binary images). The 1*2 array [colres rows]

Contains two integers that give the column resolution and row resolution in dot per_ unit (the default values). For example, if the image dimensions are in inches, colres is in the number of dots(pixels)per inch (dpi) in the vertical direction and similarly for rows in the horizontal direction. Specifying the resolution by single scalar, res is equivalent to writing [res res].

```
>>imwrite(f,'sf.tif','compression','none','resolution',.....[300 300])
```

the values of the vector[colures rows] were determined by multiplying 200 dpi by the ratio 2.25/1.5, which gives 30 dpi. Rather than do the computation manually, we could write

```
>> res=round(200*2.25/1.5);
```

```
>>imwrite(f,'sf.tif','compression','none','resolution',res)
```

where its argument to the nearest integer. It function round rounds is important to note that the number of pixels was not changed by these commands. Only the scale of the image changed. The original 450*450 image at 200 dpi is of size 2.25*2.25 inches. The new 300_dpi image is identical, except that is 450*450 pixels are distributed over a 1.5*1.5_inch area. Processes such as this are useful for controlling the size of an image in a printed document with out sacrificing resolution.

Often it is necessary to export images to disk the way they appear on the MATLAB desktop. This is especially true with plots. The contents of a figure window can be exported to disk in two ways. **The first is to use the file pull-down menu in the figure window and then choose export.** With this option the user can select a location, filename, and format. More control over export parameters is obtained by using `print` command:

```
Print-fno-dfileformat-rresno filename
```

Where no refers to the figure number in the figure window interest, file format refers one of the file formats in table

above. 8 'resno' is the resolution in dpi, and filename is the name we wish to assign the file.

If we simply type print at the prompt, MATLAB prints (to the default printer) the contents of the last figure window displayed. It is possible also to specify other options with print, such as specific printing device.

CHAPTER 7

Conclusion and Future Work

We successfully designed Intelligent Object Recognition In Underwater Images Using Image Processing. It is used for fish detection and counting of underwater images as well as underwater videos. Fish detection and counting is emerging application for underwater fish-based applications. In this project we detected underwater fishes and detected count of fishes in a given image as well as video. We converted image into gray scale and applied edge detection technique to find fishes. The detected fishes are applied with bounding boxes and count is obtained. Similar application can be used for detection of water bottle. We applied image morphology technique for detection of objects (fish or water bottle) correctly. An outcome of the fish detection system is the fish characterization, which can be used for fish species recognition. Therefore, future work can be concentrated mainly on the classification task. Given the sufficient number of training results, a classification system based on fish size and texture can be build. Also, the proposed process of size estimation can be improved by introducing the information about camera parameters, which allows to perform the camera calibration.

REFERENCES

- [1] Andrews Sobral and Antoine Vacavant. A comprehensive review of background subtraction algorithms evaluated with synthetic and real videos. *Computer Vision and Image Understanding*, 122:4–21, 2014.
- [2] Carl Schlieper. *Research methods in marine biology*. Sidgwick & Jackson, 1972.
- [3] Concetto Spampinato, Yun-Heh Chen-Burger, Gayathri Nadarajan, and Robert B Fisher. Detecting, tracking and counting fish in low quality unconstrained underwater videos. In *Proceedings of the Third International Conference on Computer Vision Theory and Applications*, volume 2, pages 514–519. Citeseer, 2008.
- [4] Helge Balk. *Development of Hydroacoustic Methods for Fish Detection in Shallow Water*. PhD thesis, Faculty of Mathematics and Natural Science, University of Oslo, 2001.
- [5] Janne Heikkilä and Olli Silvén. A real-time system for monitoring of cyclists and pedestrians. *Image and Vision Computing*, 22(7):563–570, 2004.
- [6] JL Thorley, DMR Eatherley, AB Stephen, I Simpson, JC MacLean, and AF Youngson. Congruence between automatic fish counter data and rod catches of atlantic salmon (*salmo salar*) in scottish rivers. *ICES Journal of Marine Science: Journal du Conseil*, 62(4):808–817, 2005.
- [7] Kinjal A Joshi and Darshak G Thakore. A survey on moving object detection and tracking in video surveillance system. *International Journal of Soft Computing and Engineering*, 2(3):44–48, 2012.
- [8] Massimo Piccardi. Background subtraction techniques: a review. In *Proceedings of the 2004 IEEE international conference on Systems, man and cybernetics*, volume 4, pages 3099–3104. IEEE, 2004.
- [9] Nils T Siebel and Steve Maybank. The advisor visual surveillance system. In *Proceedings of the 8th European Conference on Computer Vision Workshop Applications of Computer Vision*, volume 1, 2004.
- [10] Peter Christiansen, Kim Arild Steen, Rasmus Nyholm Jørgensen, and Henrik Karstoft. Automated detection and recognition of wildlife using thermal cameras. *Sensors*, 14(8):13778–13793, 2014.
- [11] Rita Cucchiara, Costantino Grana, Massimo Piccardi, and Andrea Prati. Detecting moving objects, ghosts, and shadows in video streams. *IEEE transactions on pattern analysis and machine intelligence*, 25(10):1337–1342, 2003.
- [12] Robert T Collins, Alan J Lipton, Takeo Kanade, Hironobu Fujiyoshi, David Duggins, Yanghai Tsin, David Tolliver, Nobuyoshi Enomoto, Osamu Hasegawa, Peter Burt, et al. *A system for video surveillance and monitoring*. 2000.
- [13] Shireen Y Elhabian, Khaled M El-Sayed, and Sumaya H Ahmed. Moving object detection in spatial domain using background removal techniques-state-of-art. *Recent patents on computer science*, 1(1):32–54, 2008.
- [14] Soharab Hossain Shaikh, Khalid Saeed, and Nabendu Chaki. *Moving Object Detection Using Background Subtraction*. Springer, 2014.
- [15] Suman Srinivasan, Haniph Latchman, John Shea, Tan Wong, and Janice McNair. *Airborne traffic surveillance systems*:

video surveillance of highway traffic. In Proceedings of the ACM 2nd International Workshop on Video Surveillance & Sensor Networks, pages 131–135. ACM, 2004.

ORIGINALITY REPORT

8%

SIMILARITY INDEX

5%

INTERNET SOURCES

2%

PUBLICATIONS

1%

STUDENT PAPERS

PRIMARY SOURCES

<https://www.geeksforgeeks.org/matlab-edge-detection-of-an-image-without-using-in-built-function/>

Internet Source

1%

2

Internet Source

3%

Bounding boxes - Computer Science Wiki

3

" Development of Hydroacoustic Methods for Fish Detection in Shallow Water. PhD thesis, Faculty of Mathematics and Natural Science, University of Oslo, 2001.

Publication

1%

4

Fish Detection from Low Visibility Underwater Videos | IEEE ...

<https://ieeexplore.ieee.org/document/8211111>
by V Shevchenko · 2018 ·

Publication

1%

5

Submitted to ICPR

Student Paper

1%

6

Yaser Sheikh, Omar Javed, and Takeo Kanade. Background subtraction for freely moving cameras. In Proceedings of the 2009 IEEE 12th

1%

7

ipar.org

Internet Source

1%

8

icomia.com

Internet Source

<1%

Exclude quotes On

Exclude bibliography On

Exclude matches Off