OBJECT RECOGNITION IN LAKE AND ESTUARY ENVIRONMENTS

By

A.J. Punch

A Thesis
Submitted to the
Faculty of the Graduate School
of
Western Carolina University
in Partial Fulfillment of
the Requirements for the Degree
of
Master of Science in Technology

Committee:

_____ Director

_____

_____

_____

_____ Dean of the Graduate School

Date: _____

Spring 2011
Western Carolina University
Cullowhee, North Carolina

OBJECT RECOGNITION IN LAKE AND ESTUARY ENVIRONMENTS

A thesis presented to the faculty of the Graduate School of Western Carolina University in partial fulfillment of the requirements for the degree of Master of Science in Technology.

By

A.J. Punch

Director: Dr. Brian Howell
Assistant Professor
Department of Engineering and Technology

Committee Members:
Dr. James Zhang, Department of Engineering and Technology
Dr. Peter Tay, Department of Engineering and Technology
Dr. Erin McNelis, Department of Mathematics and Computer Science

April 2011

ACKNOWLEDGMENTS

I would like to thank Jesus Christ, my Lord and Savor. Without his guidance, nothing I do is possible. He gave me a second chance at life and the strength to overcome all of life's challenges.

I would also like to thank my family for all your support and encouragement, especially my loving wife Katy. She has been so patient and sacrificed so much for my success over the past few years. No matter the situation, you were always willing to listen. You are more than my wife, your my best friend.

To Dr. Brian Howell, thank you so much for your teachings, support and the opportunity to work with you in a professional environment. Thanks to Dr. Peter Tay for all your time and effort to make sure I perform at the highest academic standard. To Dr. James Zhang, thank you for believing in me and giving me an opportunity to further my education at the graduate level. Thank you for all the countless hours of help with any and all issues. I would also like to extend a special thank you to Dr. Erin McNelis for selfishly giving of her time over my entire academic career. I thank the good Lord for making you a part of my life, you are truly a blessing.

Last but certainly not least, I would like to thank all my fellow graduate students and friends. Its been a long road. Without all of you, I don not think I would have made it. I would like to extend a special thanks to Matthew Proffitt for the countless hours of help, support, and being a great friend. Finally, to Kirke Shouse, thank you so much for everything. I will always treasure your friendship. Over the years you have always found a way to keep me grounded and focused through all the hardships. I would not be where I am today without you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

OBJECT RECOGNITION IN LAKE AND ESTUARY ENVIRONMENTS

A.J. Punch, M.S.T.

Western Carolina University (April 2011)

Director: Dr. Brian Howell

Traditionally, autonomous underwater vehicles employ multiple configurations of sensor payloads in order to accomplish a specific mission. Due to advances in imaging technology, imaging sonar arrays and optical imaging devices are among these payloads. Independent of mission specifics, the majority of imaging data is either stored onboard the vehicle or transmitted to a base station for later analysis. In either situation, there is limited local real time analysis and limited mission duration. One focus for increasing real time analysis is the reduction of image information. By using image processing techniques, such as edge detection, less relevant information can be eliminated while preserving important object features. This reduced object information is then used as inputs to a neural network. A neural network is a cognitive algorithm which has the ability to adapt to achieve desired tasks. These networks are able to generalize and make decisions based on partial or limited input information. The goal of this research is to create an autonomous in-situ recognition system for marine environments, specifically the processing and classification of object image data. Image information will be applied to a neural network approach to mimic higher order decision making in an artificial cognitive algorithm.

CHAPTER 1: INTRODUCTION

Autonomous vehicles, specifically autonomous robots, are an on-going field of study. My research focuses on increasing the level of autonomy thereby substantially reducing human interaction. Human interaction for these vehicles is either in terms of navigation, information retrieval, communication and control. How these vehicles are controlled and operated varies, however most are guided by remote or tethered to a control unit. For this reason, all vehicles are not truly autonomous. However, advances in technology have allowed these vehicles to perform research in environments too harsh for humans such as deep ocean, deep space, and high altitude scenarios. Mission specific applications include: underwater mapping of the sea floor, geophysical observation, and exploration of shipwrecks. Therefore, exploration and experimentation are made more sustainable due to continuous communication and control, while mission success is not dependent upon human interaction or environmental conditions. Vehicles can also be configured for specific applications, where the configurations are dependent on sensor payloads. A multi-sensor approach that is used for such mission as target localization and tracking is based on the type and number of sensory devices used. How this sensory information is processed and analyzed varies for each vehicle and mission parameter, however this information is usually processed externally.

Typically, underwater vehicles use a combination of sonar arrays and optical camera payloads for exploration, detection and recognition of underwater environments. In most cases, these sensor payloads are continuously operating with little to no pertinent

information to report. However, the amount of information, specifically imaging data, is too large to analyze locally in real time. Hence the reason for storage devices on board the vehicle or communications links to large storage base stations. Another draw back to this type of design is vehicle resource cost. The more information being actively processed and communicated decreases the vehicle longevity in the search environment.

Recently, autonomous systems for shallow water and coastal regions are being explored. Applications include: harbor patrol, pollution detection, and even military applications for intelligence gathering and ordinance location. These applications reflect changes to sensory input over time. Onboard data reduction and communication only for event triggers can increase autonomy and over all system performance.

The main focus of the following research is to develop a neural network or learning algorithm for the purpose of object recognition in underwater environments, specifically shallow water. Using existing image processing and feature extraction techniques, a significant reduction in computational and resource cost can be obtained to increase vehicle longevity. These methods will be used to recognize five fundamental geometric shapes, where these are: cone, cube, sphere, cylinder, and prism [1].

This thesis will include: a literature survey introducing the concepts for sonar and optical sensor hardware comparison, image processing, feature extraction techniques, and neural network methods. A design methodology section to introduce the feed-forward neural network with backprogation weight update using optical images followed by sections dedicated to experimental results and future work.

CHAPTER 2: LITERATURE SURVEY

As mentioned earlier, the number and type of sensor payloads are completely dependent on the defined mission parameters. Types of sensor payloads can be classified as imaging and non-imaging devices, where the most commonly used imaging devices are optical cameras and acoustic imaging systems. The implementation for each sensor is different since the input parameters and calibration techniques vary due to the range and complexity of the specific environmental conditions being measured. However, acoustic systems have longer range due to the ability of seeing through turbid conditions. All imaging sensors return enormous amounts of information although all of the sensor data may not play a role in navigation and event detection. Either way, the information has to be processed for analysis. In terms of this research, the information being processed is object image data. There are numerous methods for extracting object features from images. Some of these methods include; Canny, Sobel, and Linear Prediction. These methods detect changes in light intensity to determine shape edges in an image, where these edge define shape contours. Given that the object edges can be determined, decisions on object shape are needed. By increasing the degree of autonomy by a neural network approach, the individual vehicle can make determinations on what the object is.

The following sections are dedicated to the theory and background information for the hardware, software, and neural methods used for this design. These sections will give insight into these methods for the purpose of object recognition.

2.1 Imaging Systems

Sound navigation and ranging (Sonar) also known as acoustic ranging is one of the major sensory techniques used for underwater vehicles. Sonar systems use sound propagation for navigation, detection and communication. Sound propagation is based on the speed of sound value through a fluid medium. Temperature, pressure, and salinity are the three major factors that effect sound propagation. These three factors influence the range and return magnitude of the sound wave. Sonar systems can be classified into two categories; active or passive. Of those two classifications, systems can be further categorized into imaging and non-imaging. Distinctions can be made between conventional sonar and acoustic imaging systems. While they share hardware implementation and calibration parameters, they are not one in the same. The differences between the two are in the intended purpose of each. Conventional sonar systems are more concerned with where something is, while acoustic imaging systems is centered toward what it looks like [2].

The return sound of an acoustic imaging system is used to produce images in two-dimensional space. This is important in underwater environments due to optical camera systems being limited. Optical systems will generate higher quality images but their effective range is limited. These limitations are contributed to the difference between sound and optical wavelength characteristics. For example, in deep ocean environments optical system visibility maybe approximately 60m, but in turbid shallow water conditions visibility could be reduced to $6-10\text{m}$ or less. If ever the environment is disturbed during observation, visibility can be further reduced to less than 1m [2].

According to Sutton [2], the basic characteristics of acoustic imaging systems are: frequencies, wavelengths, apertures, resolution, range, and depth of fields. The main difference between these characteristics relative to other imaging techniques is the range variation over which the sound waves propagate. Sutton also states these parameters are

not independent of each other as shown in equations (2.1) and (2.2).

$$\lambda = \frac{c}{f} \tag{2.1}$$

$$\alpha = \frac{\lambda}{D} \tag{2.2}$$

The wavelength of the signal λ is dependent on the speed of sound $c(1500\frac{m}{s})$ divided by the frequency of the signal sent f. The angular resolution α is dependent on λ and the receiving aperture diameter D. Range and resolution are also effected by attenuation coefficients of seawater($0.3\frac{dB}{m}$ at $1MHz$) as well as reverberation effects. [2] Due to sound wave characteristics varying based on water depth, sound is distorted laterally more than longitudinally in shallow water, hence oscillation and vibration effects are more prominent. Therefore, the intensity of acoustic return is subject to change [3]. For this reason acoustic imaging systems are of lesser resolution quality as compared to optical systems.

Another aspect of acoustic systems that makes recognition difficult is object shape and material composition. An experiment performed at the University of Hawaii [3] studied these effects by varying material composition verses object shape. The results from this experiment show dynamic change in acoustic images given material composition for identical object shapes. These effects are attributed to the reflection and absorption characteristics of the acoustic wave for certain materials.

Viewpoint is another major factor in object recognition. Changing the degree of orientation varies the amount of object edges subjected to the acoustic signal. This change affects the amount of acoustic signal return thus producing a extremely different image. Therefore, the intensity of the acoustic signal is dependent on three factors; object shape,

material composition and orientation. Since acoustic reflection is directly related to shape and material composition, and reflection angle is dependent on orientation, the intensity of the produced image is a function of angular reflection shown in equation (2.3) [3]

$$I = f(ma, \alpha) \tag{2.3}$$

where ma is compositional acoustic reflection and α is reflection at angle of incident.

An experiment conducted at Harbin Engineering University [4] focused on feature extraction of sonar images for object classification. By scanning two extremely different objects at varying orientations, it was shown that in some cases scattering of the acoustic signal produced relatively identical images. The amount of shadow present in the images make it extremely difficult to determine object features. However, further experimentation by [5] introduced a partial shape recognition method using contour points. This method was based on a "curvature-based polygonal approximation" which combined corner detection and polygonal approximation. By using these contour points or landmarks, a partial reconstruction of each object was determined.

The focus of both experiments is object recognition and classification, however the purpose of [5] is to demonstrate the low resolution quality of certain imaging sonar systems. Recognition and classification will be discussed in more detail in the image processing section.

When detecting objects underwater particularly for the first time, situations my occur when the object is partially obscured by environmental conditions or by debris. Due to the low resolution of acoustic imaging systems, these conditions could mislead an autonomous processing technique into falsely classifying the object. In these cases a partial recognition or adaptive method that considers the occluded object scenario is needed.

2.2 Feature Extraction and Image Processing Techniques

When performing object detection, the focus is on the extraction of features from the object, where the features are defined as the structural information. The structural information can be further defined as object edges in the image. Therefore, given these edges the object features can be determined [6].

Feature extraction is a process of recognizing and extracting specific shape components of an object from its images. Shape composition is determined by three parameters, object size, orientation, and position, although all three may not influence feature extraction at the same time. These parameters define three specific conditions for all feature extraction. Position invariance, also known as location invariance is important in determining object shape regardless the object position. Orientation invariance for object shapes influence feature extraction due to the shape or camera aperture location being unknown. The most difficult condition is shape size. Shape size or size invariance can be determined either by using the distance from the camera or estimations by known shapes or features in the image background. However, these estimations are still dependent on orientation and position. Therefore position and orientation are important to size determination. The most important condition for feature extraction is illumination. Illumination invariance should have little to no affect on feature extraction [1].

Another aspect of feature extraction is to simplify a complex image by decomposing it into basic fundamental shapes. For example, if an image is of a person, specific body features such as eyes, head, arms, and torso can be viewed as simple shapes. The eyes and head portion can be viewed as spherical shapes, while the arms can be viewed as cylindrical or rectangular in nature. Therefore, more complex image shapes can be decomposed into simple shape structures such as cones, cubes, and prisms [1], similar to the way individuals view real world environments.

The concept of feature extraction was first introduced by Hough [7] to find bubble shapes rather than objects. This method was further expanded by Duda and Hart [8] and focused on the extraction of lines and conical shapes. These early methods focused on cross-sections from each object shape, however, the expanded versions were used to find shape contours. Other methods include template matching and template matching with Fourier transforms. Template matching is more position dependent, hence changes to orientation and scale make it difficult to extract and match due to gaps or spaces appearing in the coordinate system caused by rotation [1]. Computationally, template matching is much larger compared to versions of the Hough transform due to independence from shape parameters. However, the Hough transform requires large amounts of computational storage [1].

A simpler feature extraction technique is thresholding and subtraction. This technique uses known illumination and background values at the time the image is captured to set an illumination threshold. Once this threshold is computed, the background can be subtracted from the foreground leaving only the object space. Therefore, this method is highly dependent on known values of illumination [1]. All images will be contaminated by noise. Even though noise can be filtered out, the degree and computational cost can vary based on specific extraction technique.

Another common approach in image processing is edge detection. As previously mentioned, edges are the structural information of an object shape. Edges can also be viewed as changes of light intensity. An edge map is an image that contains structural information and ignores less important image details [6]. For this reason, edge detections methods are prime candidates for feature extraction. The more prominent edge detection methods are “Sobel”, “Canny”, and “Prewitt”. Canny’s method is regarded as the optimal edge detection method and has been mathematically proven for optimal performance [9].

Developed in 1986, Canny's method focuses on maximizing signal to noise ratio(SNR) in an image. He further defines three performance criteria and optimal filter design in order to maximize this ratio. The three performance criteria are as follows [9]:

1. Good Detection: The probability of falsely marking nonedge points and failing to mark real edge points should be as low as possible.

2. Good Localization: Points marked as edges should be as close to the true edge center.

3. Only one response to an edge: If there are two responses to the same edge, one must be false.

The localization criteria is dependent on maximizing the SNR shown in equation (2.4). The localization criteria can be derived by taking the derivative of the signal component and the noise component, given the noise component is a Gaussian random variable shown in equation (2.5). According to Canny, since these constraints are not independent of each other, maximizing the inner product of these two will maximize both criteria simultaneously. This inner product inherently simplifies the analysis for step edges. A good localization combined with an optimal Gaussian filter will smooth out step edges which in turn smooths out edge contours

$$\mathrm{SNR} = \frac{\left[\int_{-\infty}^{\infty} G(-x)f(x)dx\right]}{n_o\sqrt{\int_{-\infty}^{\infty} f^2(x)dx}} \tag{2.4}$$

$$\text{Localization} = \frac{\left[\int_{-\infty}^{\infty} G'(-x)f'(x)dx\right]}{n_o\sqrt{\int_{-\infty}^{\infty} f'^2(x)dx}} \tag{2.5}$$

where $G(\cdot)$ is a Gaussian filter.

Another form of edge detection is in the use of Linear Prediction (LP). One approach was developed at Western Carolina University by Dr. James Zhang. This method combines a linear prediction detection method with entropy thresholding. Unlike Canny's method which focuses on the maximization of the signal to noise ratio, LP uses a linear function of previous samples to estimate future values, or simply stated, using past values to predict future values. [6] The key to this prediction technique is choosing the appropriate filter coefficients to minimize prediction errors. Given a one-step forward linear predictor of order p, which can be expressed as a convolution of the prediction coefficients $a[k]$, and past values $x[n-k]$ shown in equation (2.6), where $\hat{x}[n]$ is the expected value for each image pixel. The prediction error $f_p[n]$, is a subtraction of the estimated image from the real image shown in equation (2.7), where $a(0) = 1$ by definition [6].

$$\hat{x}[n] = -\sum_{k=1}^{p} a[k]x[n-k] \tag{2.6}$$

$$f_p[n] = x[n] - \hat{x}[n] = \sum_{k=0}^{p} a[k]x[n-k] \tag{2.7}$$

In order to minimize prediction error, the minimum mean square error(MMSE) is calculated using the conjugate of the k^{th} autocorrelation value of $x[n]$ and is shown in equation (2.8) [6].

$$\min\left\{\mathcal{E}_p^f\right\} = \min\left\{E\left[|f_p[n]|^2\right]\right\} = r_{xx}(0) + \sum_{k=1}^{p} a[k] r_{xx}(-k) \tag{2.8}$$

As previously mentioned, edges in an image are abrupt changes in light intensity, therefore edges can be considered discontinuities for a given 2-D signal [6]. These "jumps" in intensity are not predicted well using this method due to the errors at these "jumps" being large. However, LP does predict values over a smooth transition signal state. Applying a threshold to separate smooth states from the "jumps" allows the extraction of values at the "jumps", where these "jumps" represent the edge information. Therefore, this LP detector maximizes the predicted errors used for the extraction of edge information [6].

Each pixel is generally stored as an eight bit unsigned integer, that corresponds to a gray level. LP is applied to the $+x$ and $-x$ directions respectively in order to generate edge information for the x direction. This process is repeated for both $+y$ and $-y$ directions. The edges or errors in each direction can be computed by equations (2.9) and (2.10) with the image edges obtained by equation (2.11) [6]. Due to subtracting information from each direction, this method suppresses less important information while amplifying the errors associated with these abrupt transition states.

$$\triangle I_{x,error} = \hat{I}_{+x} - \hat{I}_{-x} \tag{2.9}$$

$$\triangle I_{y,error} = \hat{I}_{+y} - \hat{I}_{-y} \tag{2.10}$$

$$I_{edge} = \sqrt{\triangle I_{x,error}^2 + \triangle I_{y,error}^2} \tag{2.11}$$

To further separate the edge information from the less important background information, an entropy-based threshold is introduced. Thresholding is a useful technique for

image segmentation. Given a gray scale image of size MxN with scale 0 to K. The total probability for the edge points $P_{\mathcal{E}}$ and the probablity for the background $P_{\mathcal{B}}$ are calculated by equations (2.13) and (2.14) respectively , given equation (2.12) which represents the *a priori* probability p_k for the given gray scale image [6]

$$p_k = \frac{N_k}{M \cdot N} \tag{2.12}$$

where N_k is the pixel number of the edge,

$$P_{\mathcal{E}} = \sum_{k_1=1}^{K_{TH}} p_{k_1}, \tag{2.13}$$

and

$$P_{\mathcal{B}} = \sum_{k_2=K_{TH}+1}^{K} p_{k_2}. \tag{2.14}$$

The term K_{TH} is the separation threshold between the background and the edge information. The entropy for the edges $H(\mathcal{E})$ and entropy for the background $H(\mathcal{B})$ can be expressed in the equations (2.15) and (2.16), where the total entropy histogram can be seen in equation (2.17).

$$H(\mathcal{E}) = -\sum_{k_1=1}^{K_{TH}} \frac{p_{k_1}}{P_e} \ln\left(\frac{p_{k_1}}{P_e}\right) \tag{2.15}$$

$$H(\mathcal{B}) = -\sum_{k_2=K_{TH}+1}^{K} \frac{p_{k_2}}{1-P_e} \ln\left(\frac{p_{k_2}}{1-P_e}\right) \tag{2.16}$$

$$\Phi(K_{TH}) = H(\mathcal{E}) + H(\mathcal{B}) \tag{2.17}$$

Therefore, the K_{TH} value that maximizes the histogram is the desired threshold, thus separating the object from the background. This segmentation of object edges from the background leads to a more pronounced edge map.

2.3 Neural Methods

A neurocomputing approach to information processing first involves a learning process within an artificial neural network architecture that adaptively responds to inputs according to a learning rule [10]. A neural network has the ability to learn by example. These networks also have the ability to learn from their environments and adapt in an interactive manner. Given specific environments neural networks have the ability to generalize and infer given partial or limited input information. Once these networks have learned what they need to know, they can be applied to perform specific tasks. Therefore, a neural network mimics or emulates its biological counterpart in an artificial cognitive algorithm [10].In order to bridge the link between an artificial neural network and it biological counterpart, a brief introduction of biological neural networks is needed.

The nervous system is a complex neural network where the brain is the central element. The brain receives sensory information from receptors, processes this information and delivers action commands to effectors. The human brain consists of approximately 10^{11} neurons, the central processing element. Each neuron is interconnected through subnetworks called nuclei. These subnetworks are collections of neurons designed to process specific information. Therefore, the biological neuron is the basis for the artificial neuron [10]. The nervous system contains a wide variety of neurons, each with specific numbers, sizes, patterns and electrical properties. Each neuron is interconnected to other neurons in each subnetwork by connectors called synapses. The number of synaptic pathways

for each neuron can vary depending on type and purpose. When the nervous system receives some sensory input, this information is transmitted to specific subnetworks through electrical impulses. Once the magnitude of the impulse reaches a certain threshold, information is passed to the effectors to initiate the desired action. Given the complexity of the subnetworks and numbers of synaptic pathways, the human brain is an adaptive, nonlinear, parallel computer. Artificial neural networks model the human brain in order to accomplish specific tasks [10].

Artificial neural networks are defined by how they learn over time (training). The ability for each network to learn is further defined by the complexity of the network design and the type of input information used for training. Thus, artificial neural networks can be categorized into supervised and unsupervised learning networks, where the function of each network can be linear or nonlinear. The most predominant supervised learning networks are; associative memory, feed-forward multilayer perceptrons trained through backpropagation, counterpropagation, and radial basis function. The commonality between all four networks is their capability of performing pattern association, classification, and function approximation. An associative network is a mapping network where given incomplete or partial input information, the network is able to "associate" correct patterns. Counterpropagation networks are known as optimal self-programming lookup tables, while radial basis function networks have the ability to learn more quickly compared to other network paradigms. Feed-forward multilayer perceptrons with variants of backpropagation update are the most well known and more often used. These networks work on the principle of steepest descent(gradient) for learning [10]. Figure (2.1) shows the architecture for a feed-forward neural network minus the backpropagation weight update portion. Similar to its biological counterpart, the amount of neuron or node interconnectivity can be seen for each layer, where a layer can be viewed as a subnetwork of neurons. Each layer consists

of a weight matrix corresponding to specific input values. These weights are multiplied by each input given a specific neuron or node. These resultant scalar values are summed together and used as the input to an activation function. The activation function shown in equation (2.18) also known as a transfer function bounds the input vector of scaler values between a specified range. The output of the activation function is used as the input to the following network layer. The process is repeated for each layer throughout the network [10].

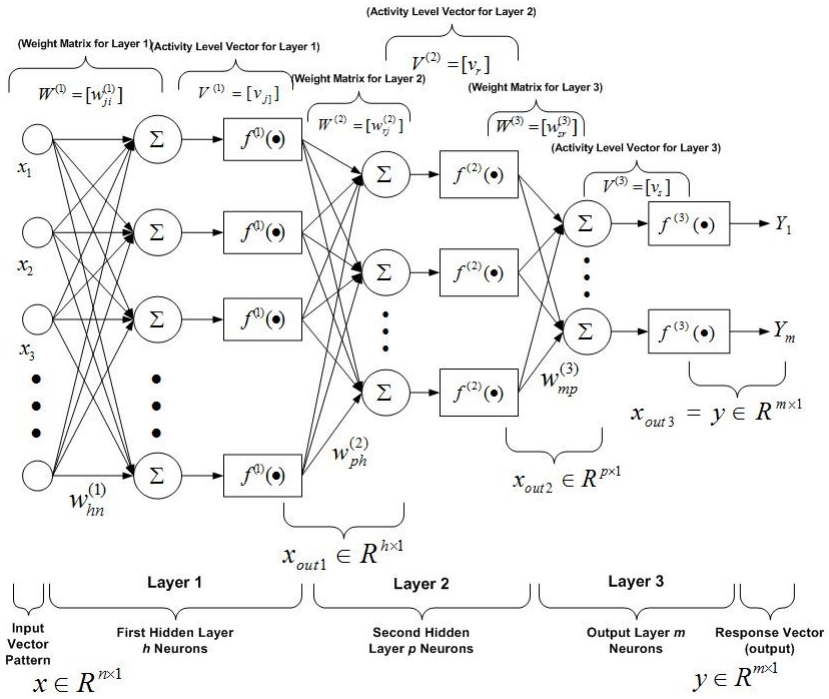


Figure 2.1: Feed-forward Neural Network without Backpropagation

An example of a possible activation function shown in equation (2.18) is of a sigmoid function.

$$f(x) = \frac{1}{1+e^{-x}} \tag{2.18}$$

The characteristics of a sigmoid function allow for the bounding of values between $0-1$. Given specific network applications, the type of function used can change based on a

desired output scale. The choice of activation function can also depend on the neural network paradigm. An example of this is in the use of radial basis function networks. The activation functions for this particular paradigm can take one of three forms, Gaussian, multiquadratic, and inverse multiquadratic.

Figure (2.2) is the feed-forward neural network with backpropagation weight update. In order for a network to be used to accomplish specific tasks, network training is needed. This training involves information being applied to the network relative to the desired task application. Each input is multiplied by a set of weights over each training cycle or epoch. After each input is processed, error is calculated given the estimated output for that given training iteration. This estimated value is subtracted from the nominal or desired output predetermined for this application, where the range of the output is determined by the activation function.

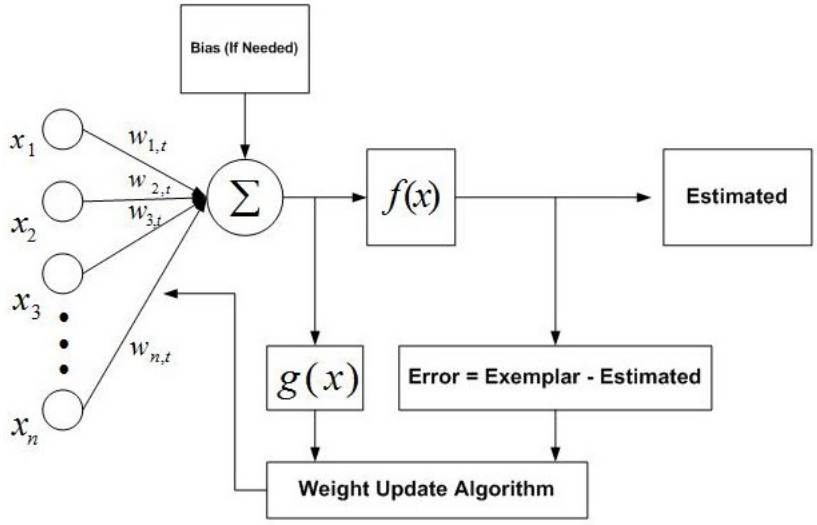


Figure 2.2: Feed-forward Neural Network with Backpropagation Weight Update

Since the feed-forward neural network is based on the steepest descent approach, the first derivative of the activation function shown in equation (2.19) is needed to further

calculate the weight update for each subsequent network layer.

$$g(x) = \frac{e^{-x}}{(1+e^{-x})^2} \tag{2.19}$$

This derivative is calculated using the given sigmoid activation function in equation (2.18). Equation (2.20) is the function for calculating the weight update given the s^{th} layer at the $(k+1)^{th}$ iteration, where $\mu^{(s)}$ is the learning rate assigned to the network during training.

$$w_{j,i}^{(s)}(k+1) = w_{j,i}^{(s)}(k) + \mu^{(s)}\delta_j^{(s)}x_{out,i}^{(s-1)} \tag{2.20}$$

The estimated output given a certain training cycle is denoted by $x_{out,i}^{(s-1)}$. The layer error is denoted by $\delta_j^{(s)}$ and is shown in equation (2.21).

$$\delta_j^{(s)} = \begin{cases} \left(d_{q,j} - x_{out,j}^{(s)}\right) g\left(v_j^{(s)}\right) & \text{if s is the last output layer} \\ \left(\sum_{h=1}^{n_s+1} \delta_j^{(s+1)} \cdot w_{h,j}^{(s+1)}\right) g\left(v_j^{(s)}\right) & \text{otherwise} \end{cases} \tag{2.21}$$

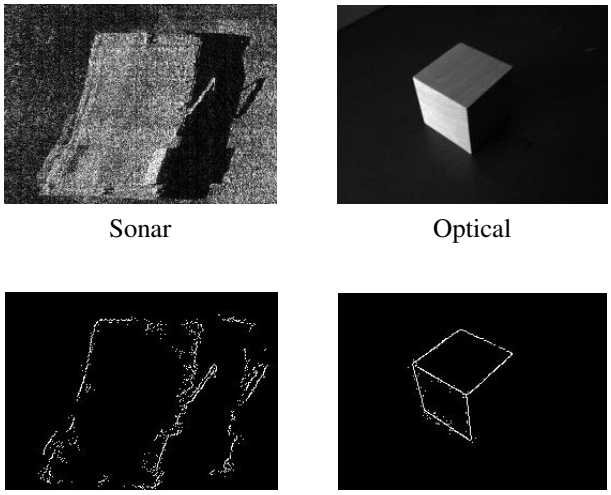
Since, this is a backpropagation algorithm, the last layer in the network is updated first, and each subsequent layer update propagates in reverse order. This is important since the weight update equations vary given the network layer. The difference value determined from the subtraction of the estimated output from the desired is multiplied by the first derivative of the activation function. The previous layers use a summation of error for the previously updated layer multiplied by the weight matrix values multiplied by the first derivative of the activation function.

Research in this field of study has seen increase over the past twenty years. An experiment performed by [11] focus on a Feed-forward neural network with backpropagation weight update for prediction of acoustic signals to model the effects of attenuation through layered media. Given what is know of acoustic reflection characteristic, they used a five layer network architecture to increase accuracy. This network applied an attenuation model from underwater media to this network to simulate the exponential decay of

the signal amplitudes. A follow up experiment by [12] was performed using a parallel architecture to further predict the media parameters given reflection coefficients and travel times of the acoustic signals. However, the training data set used for this experiment is artificially generated based on the previous network experiment. Given this training set, the parallel network was able to accurately detect media parameters and detect the output signal from each media. The values that were returned were scaled for a min-max amplitude range predetermined before testing.

CHAPTER 3: METHODOLOGY

The implementation for this autonomous system is a three step process. The initial step is the creation of an optical image data base. This data base consists of fundamental object shape images captured over varying orientations and illuminations. Due to the lack of sonar image availability for use in the database, optical images were used for all experiments. Since edge maps are used as inputs to the neural network, the results should be applicable to sonar images. Shown below in Figure (3.1) is an image comparison of a sonar image and a gray scaled optical image from this data base with corresponding edge maps.



Sonar Optical

Sonar Edge Map Optical Edge Map

Figure 3.1: Sonar Image / Optical Grey Scale Image

The data base is divided into two image sets, a training and testing set. Each set

consists of equal amounts for each object shape and orientation. Once these sets were determined, each image was processed using the Linear Prediction with Entropy Thresholding edge detection method to produce the corresponding edge maps. Shown in Figure (3.2) is a depiction of these orientations with varying lighting conditions with their corresponding edge maps.

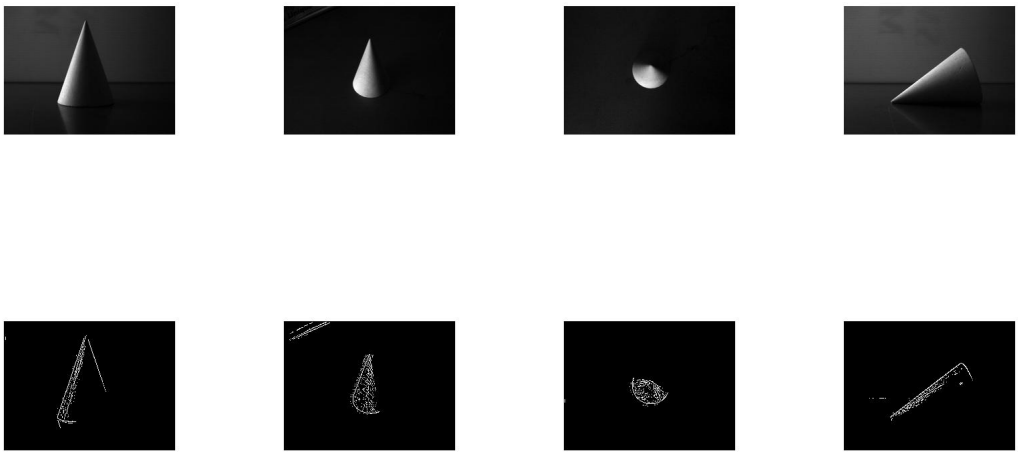


Figure 3.2: Edge Map Generation for Varying Illuminations and Orientations

Shown in Figure (2.2) is the Feed-forward neural network with backpropagation weight update used for this design. A collection of edge maps corresponding to the five fundamental shapes will be used as inputs to this network. Each training cycle will consist of forty image frames. The specific shape image counts vary due to the variation in orientation needed for each shape type. The activation function $f(x)$ used is a sigmoid function in equation (2.18). The first derivative $g(x)$ in equation (2.19) is used to calculate the weight update functions for each network layer. The network will train for 300 iterations (epochs). At the end of the 300 epochs the weight files for the trained network are written

to text files. These files are uploaded back to the network in order to run the network in feed-forward mode. At this point the backpropagation weight update is turned off.

The network process shown in Figure (3.3) begins when an object image is captured. This image is processed using the LPC edge detection method to extract the edge map. This array of pixels is used as the input to the network in feed-forward mode. The output for each image is a 5x1 matrix corresponding to each shape type. Figure (3.3) shows a nominal case in which the input image is of a cube. The output recognition range is on a scale from $0-1$, where a value close to 1 represents high recognition confidence for that particular shape type. A recognition rate of 0 represents confidence that the object is not that shape.

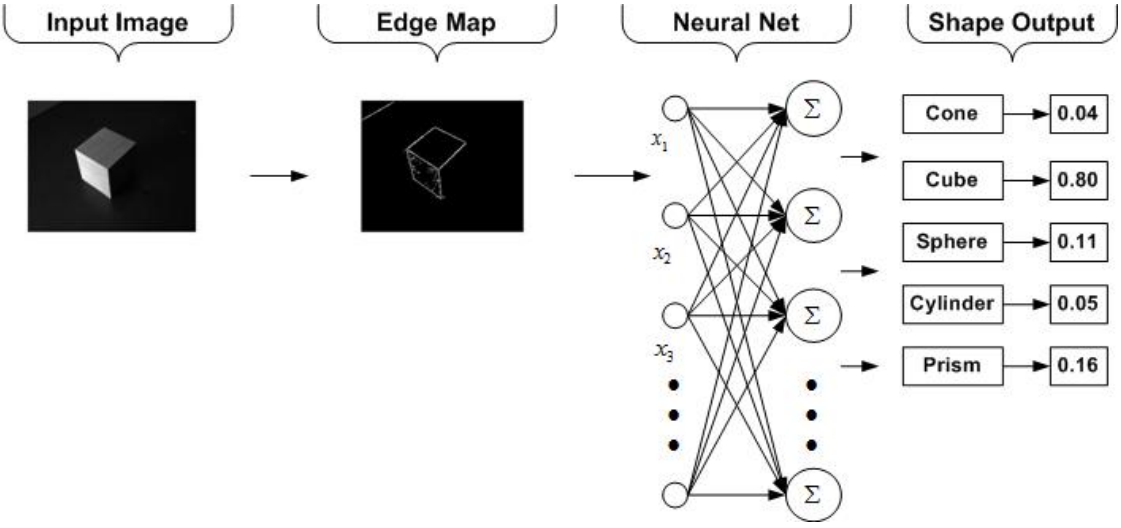


Figure 3.3: Feed-forward Network Input to Output Process

Network testing involves applying each of the five object shapes to the network in four orientations and a noised corrupted image. The four orientations are 0°, 45°, 90°, and random. The random orientation image chosen is also corrupted by added noise and used as the fifth test input image. Each object shape is processed through the network over three

replicates. No two images are duplicated. The results for each replicate will be recorded in spread sheet format for later analysis. The error attributed to each training cycle will also be stored to verify system learning with no over training.

CHAPTER 4: RESULTS AND ANALYSIS

This chapter is dedicated to the analysis of results from the feed-forward neural network algorithm using LP edge detection. These results are from the experiment described in the methodology portion of this thesis. All shape types were tested over defined orientations using this method to verify recognition rate. The purpose for defining the image orientation before testing allows for greater insight into which if any orientation presents recognition problems for a given shape. These recognition rates are dependent on limited training time for this algorithm.

Each test consisted of a five images with one of the five being corrupted by added noise. Shown in Table (4.1), are the recognition rate results using cone images as inputs to the network. Given the recognition rates shown, this network has high confidence that the input image is of a cone, except in the case of noise added images. The low recognition rates for these images are attributed to the fact that during the training phase, no corrupted images were used for network learning. The purpose for this image test was to see if the network given limited training time could generalize this image shape when the input edge map is not ideal. The remaining four image inputs show high confidence for correctly identifying this shape type with the lowest set of recognition rates corresponding to the random images. The second highest recognition rate mean value corresponds to the prism shape. This is due to the cone and prism shapes being approximately identical in specific orientations in a 2-D image. An example of this is the 0° case. From this orientation, the difference between these shape types can be seen at their base. The cone will

have some conical tendency where the prism is linear. Given the pixels that were detected using LPC these images could look identical. Given this relationship, the recognition for the cone at 0° and the random orientation are lower where the prism recognition rates are of higher confidence. Given longer training time this might be eliminated.

Table 4.1: Neural Network Recognition Rates for Cone Image

Cone															
	Cone			Cube			Sphere			Cylinder			Prism		
Input	Rep1	Rep2	Rep3	Rep1	Rep2	Rep3	Rep1	Rep2	Rep3	Rep1	Rep2	Rep3	Rep1	Rep2	Rep3
0°	0.65	0.69	0.68	0.09	0.10	0.12	0.01	0.05	0.04	0.03	0.04	0.05	0.22	0.20	0.26
45°	0.72	0.74	0.70	0.08	0.09	0.03	0.19	0.18	0.15	0.04	0.03	0.10	0.13	0.11	0.11
90°	0.79	0.78	0.75	0.10	0.11	0.08	0.14	0.14	0.17	0.15	0.17	.011	0.17	0.17	0.19
Rand	0.55	0.56	0.54	0.15	0.13	0.12	0.12	0.14	0.12	0.13	0.11	0.15	0.32	0.31	0.28
Noise	0.12	0.20	0.15	0.17	0.21	0.20	0.05	0.09	0.05	0.27	0.21	0.25	0.23	0.25	0.31
Mean	0.58			0.12			0.11			0.13			0.22		
St dev	0.23			0.04			0.06			0.03			0.07		

Table (4.2) below are the recognition rates when the prism shape is the input to the neural algorithm. These results show high confidence for recognizing the input shape is a prism. Similar to the cone input results, the network has trouble recognizing the noise corrupted image. The confidence for this specific input is indeterminate across all shape outputs, although there is a correlation between the prism and cone results for the 0° orientation. Even though the prism is correctly recognized for this input, the confidence is less compared to other orientation inputs. This corresponds to the higher confidence of the cone shape for this input. Given these results, there is a direct recognition issue for this specific orientation. Therefore, a larger image library, and more training time, these values might be more consist with other recognition values.

Table (4.3) and Table (4.4) show the recognition rates when the input to the neural algorithm is a cube and cylinder respectively. In both cases, the recognition confidence is significantly higher for the noise corrupted images. The network was able to generalize given these edge maps that the object shapes were the correct image input. These values are still close to 0.5 in both cases, however these results demonstrate the ability for the

Table 4.2: Neural Network Recognition Rates for Prism Image

Prism															
	Cone			Cube			Sphere			Cylinder			Prism		
Input	Rep1	Rep2	Rep3	Rep1	Rep2	Rep3	Rep1	Rep2	Rep3	Rep1	Rep2	Rep3	Rep1	Rep2	Rep3
0°	0.30	0.31	0.25	0.13	0.13	0.16	0.04	0.05	0.05	0.11	0.12	0.10	0.65	0.60	0.62
45°	0.07	0.07	0.08	0.10	0.09	0.13	0.01	0.02	0.01	0.10	0.10	0.11	0.85	0.82	0.85
90°	0.06	0.09	0.10	0.07	0.06	0.07	0.04	0.06	0.09	0.03	0.06	0.05	0.70	0.77	0.74
Rand	0.11	0.15	0.12	0.08	0.09	0.14	0.05	0.04	0.05	0.07	0.06	0.04	0.84	0.83	0.84
Noise	0.12	0.11	0.11	0.04	0.10	0.10	0.05	0.06	0.07	0.06	0.06	0.05	0.14	0.10	0.11
Mean	0.14			0.10			0.05			0.07			0.63		
St dev	0.08			0.03			0.02			0.03			0.23		

network to generalize given partial information. The recognition rates produced from the cube input verify network learning for this shape type. However, the 90° orientation recognition rate is lower compared to other input orientations. This corresponds to the higher recognition rate of the prism produced from this input. The similarity between these two shape types at this orientation is the result of a four-sided prism being approximately identical to a cube from this angle. Given the input resolution of the edge map, the cross-sections from the prism are non-existent. Therefore, the prism looks like a cube. The cylinder results are especially interesting. Two factors are attributed to the results for a 2-D image of a cylinder in two orientations, 0° and 90°. For the 0° case, there are similarities between a cylinder and 3-sided prism in 2-D space. Both edge maps will look rectangular, thereby attributing to the recognition rates corresponding to that orientation. For the 90° case, it would stand to reason that the recognition rate of a sphere would be higher with respect to the cylinder in this orientation. From a top down view, both images would have a sphere shape in each edge map. The reason for this discrepancy is attributed to the over training of the sphere shape. This will be explained in more detail when the results from the sphere are introduced.

Figure (4.1) shows the epoch(iteration) error curves for each of the previously discussed shape types. The error values for each curve are the average error calculated for each set of training images for each object shape type. The spike at time 0 is attributed

Table 4.3: Neural Network Recognition Rates for Cube Image

Cube															
	Cone			Cube			Sphere			Cylinder			Prism		
Input	Rep1	Rep2	Rep3	Rep1	Rep2	Rep3	Rep1	Rep2	Rep3	Rep1	Rep2	Rep3	Rep1	Rep2	Rep3
0°	0.08	0.09	0.08	0.80	0.78	0.79	0.10	0.10	0.08	0.03	0.05	0.03	0.11	0.12	0.11
45°	0.09	0.11	0.08	0.82	0.80	0.81	0.04	0.05	0.03	0.03	0.03	0.04	0.17	0.17	0.15
90°	0.10	0.09	0.12	0.57	0.68	0.59	0.07	0.10	0.08	0.03	0.04	0.03	0.33	0.30	0.31
Rand	0.04	0.06	0.06	0.80	0.71	0.77	0.11	0.14	0.12	0.05	0.05	0.07	0.16	0.18	0.18
Noise	0.16	0.18	0.12	0.61	0.55	0.59	0.31	0.29	0.30	0.07	0.08	0.08	0.21	0.20	0.20
Mean	0.10			0.71			0.13			0.05			0.19		
St dev	0.04			0.10			0.09			0.02			0.07		

Table 4.4: Neural Network Recognition Rates for Cylinder Image

Cylinder															
	Cone			Cube			Sphere			Cylinder			Prism		
Input	Rep1	Rep2	Rep3	Rep1	Rep2	Rep3	Rep1	Rep2	Rep3	Rep1	Rep2	Rep3	Rep1	Rep2	Rep3
0°	0.05	0.09	0.04	0.06	0.08	0.08	0.09	0.10	0.10	0.73	0.77	0.74	0.14	0.20	0.71
45°	0.16	0.11	0.15	0.11	0.11	0.12	0.05	0.06	0.12	0.76	0.75	0.76	0.09	0.08	0.12
90°	0.08	0.09	0.05	0.08	0.05	0.11	0.13	0.13	0.10	0.63	0.63	0.65	0.31	0.30	0.31
Rand	0.06	0.06	0.11	0.04	0.06	0.09	0.09	0.08	0.05	0.44	0.44	0.52	0.29	0.28	0.29
Noise	0.04	0.08	0.03	0.16	0.12	0.20	0.24	0.24	0.22	0.46	0.40	0.49	0.21	0.20	0.25
Mean	0.08			0.10			0.12			0.61			0.22		
St dev	0.04			0.04			0.06			0.14			0.08		

to the randomization of the weight matrix before training began. Since the feed-forward method works on the principle of steepest descent or gradient method, these values quickly trend to lower and lower values as training iterations increase. The gradient method is focused on determining the global error minimum. As seen in these graphical representations, the error given specific iterations are trending lower and lower. In the case of the cylinder and cube, the error has not reached a global minimum by the 300 epoch. Given longer training cycles, this error would trend lower thus producing higher recognition values for these specific shape types.

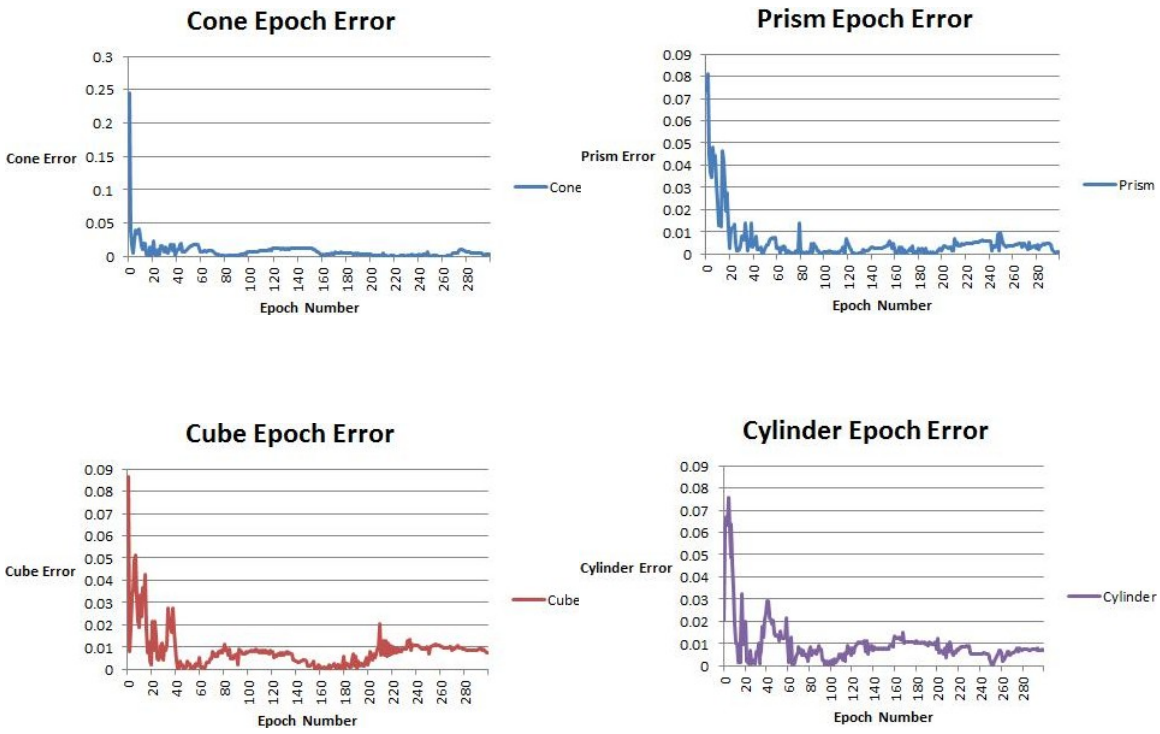


Figure 4.1: Cone, Prism, Cube, Cylinder Epoch Error Rates

Table (4.5) shows the results for the sphere shape input. In all cases except for the 0° orientation, the recognition rates are considerably low. This is due to this shape being over trained during backpropagation weight update portion of this algorithm. The 0° results show a maximum output value given the defined output range being between $0-1$. This is due to the network having the inability to generalize or infer given partial information. The values for this orientation correspond to edge maps that are complete circles with limited to no pixels missing from these shapes. The remaining test orientations are partial spheres or hemisphere like shapes. Given the inability to generalize, the network only recognizes a complete circle as a sphere, and cannot infer that the other images are spherical in nature. For this reason, the network could not recognize the spherical tendency in the cylinder presented earlier.

Table 4.5: Neural Network Recognition Rates for Sphere Image

Sphere															
	Cone			Cube			Sphere			Cylinder			Prism		
Input	Rep1	Rep2	Rep3	Rep1	Rep2	Rep3	Rep1	Rep2	Rep3	Rep1	Rep2	Rep3	Rep1	Rep2	Rep3
0°	0.13	0.11	0.15	0.00	0.01	0.00	1.00	1.00	0.98	0.06	0.05	0.10	0.00	0.00	0.00
45°	0.10	0.08	0.15	0.16	0.20	0.17	0.21	0.28	0.22	0.27	0.22	0.25	0.13	0.14	0.18
90°	0.06	0.09	0.06	0.17	0.21	0.19	0.14	0.20	0.15	0.09	0.05	0.11	0.10	0.07	0.09
Rand	0.06	0.08	0.06	0.17	0.11	0.12	0.14	0.15	0.14	0.09	0.09	0.10	0.10	0.10	0.18
Noise	0.22	0.22	0.20	0.24	0.40	0.27	0.06	0.10	0.09	0.05	0.09	0.06	0.30	0.31	0.30
Mean	0.12			0.16			0.32			0.11			0.13		
St dev	0.06			0.11			0.35			0.07			0.10		

Shown below in Figure (4.2) is the epoch error curve for the sphere shape. It can be seen that at approximately iteration 118 the error starts to trend back up. This is due to the lack of dynamic change in each of the training images. This lack of change means the network will learn more quickly this shape is a sphere. Given this faster training, the network begins to over train this shape type. When over training occurs, the network loses its ability to generalize given partial information. Therefore, only complete spheres or circles in a 2-D image can be recognized. This over training also hinders the network of generalizing other shape types that have spherical tendency. Even though this result is not desired, the information gained can lead to better understanding of optimal training time given shape type as well as insight into frequent network learning between training cycles. Frequent error checks could give insight to network status during the training process rather than waiting until the network has completed all cycles.

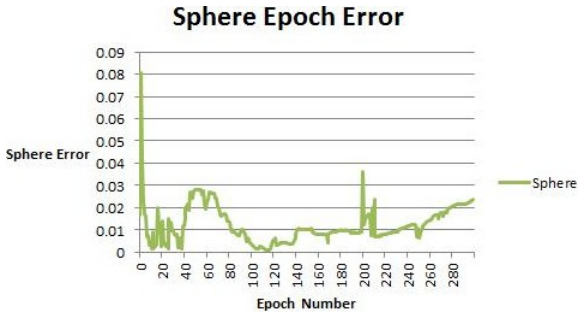


Figure 4.2: Sphere Epoch Error Rates

CHAPTER 5: CONCLUSION AND FUTURE WORK

In Conclusion, based on the results presented from this research primitive shapes can be recognized and determined using a feed-forward with backpropation weight update neural network with optical image input. Given the limited data base and training time, recognition rates showed accurate recognition of each shape. Over training for this network was due to the lack of variation for the sphere shape type. Given less training time for this particular shape, over training could be avoided. This approach also demonstrated the adaptiveness of the LPC edge detection method with regards to processing batch image sets. Given any image, a quality edge map can be generated from this method thus preserving structural features for the purpose of object recognition. The feed-forward mode of this network also demonstrated the ability to perform local real time analysis, where this time scale does not include training time. From the time an image is captured, the shape recognition process takes approximately 3 seconds using a C-based language. Therefore, the experiment in this thesis demonstrated the robustness of the feed-forward neural network with backpropagation to recognize five fundamental objects in optical images.

Given the results from this research, several suggestions are offered for future work. The first of which is increasing the size of the image data base to include noise corrupted images. Due to the less than desirable results produced from adding noise, these images could be used for both training and test sets. These corrupted images coupled with the addition of emergent shape images would produce a more robust data base which could lead to more accurate recognition rates for real world environmental events. Due to the

lack of available sonar images, optical images were implemented to verify network learning and the ability for this network to recognize primitive shapes. Given this information, further testing is needed to include a sonar image library to verify the effectiveness of the LPC edge detector and the feed-forward neural network. Once this library is obtained, improvements to the overall data base could lead to higher recognition rates for all imaging systems.

Additional improvements could be attained by the investigation of alternative neural network paradigms. These networks can be used as stand along networks or used in a parallel fashion. The increased computational power of parallel networks could reduce training time and increase global network recognition. The addition of a linear classifier or associative memory network is one option. The output from this addition could be used as the input for the feed-forward network. Due to the significant training involved for this network, another alternative is the use of radial-basis function or momentum learning algorithm. This would reduce training time and allow for the image size to be increased to produce a more defined edge map. More structural information could lead to more confident recognition rates corresponding to specific inputs. Using these alternative networks would allow for more research in optimizing the training time for specific mission applications. Given a significant reduction in training time, mission preparation time is reduced.

BIBLIOGRAPHY

[1] A. Aguado M. Nixon, *Feature Extraction and Image Processing*, Elsevier Ltd, 2008.

[2] J.L. Sutton, "Underwater acoustic imaging," *Proceedings of the IEEE*, vol. 67, no. 4, pp. 554–566, 1979.

[3] S. C. Yu, T. W. Kim, G. Marani, and S.K. Choi, "Real-time 3d sonar image recognition for underwater vehicles," in *Underwater Technology and Workshop on Scientific Use of Submarine Cables and Related Technologies, 2007. Symposium on*, 2007, pp. 142 –146.

[4] Y. Lu and E. Sang, "Underwater target's size/shape dynamic analysis for fast target recognition using sonar images," in *Underwater Technology, 1998. Proceedings of the 1998 International Symposium on*, 1998, pp. 172 –175.

[5] Daniel S., Guillaudeux S., and Maillard E., "Adaptation of a partial shape recognition approach," in *Systems, Man, and Cybernetics, 1997. 'Computational Cybernetics and Simulation'., 1997 IEEE International Conference on*, 1997, vol. 3, pp. 2157–2162.

[6] J.Z. Zhang, P.C. Tay, and R.D. Adams, "A novel image edge detection method using linear prediction," in *2010 53rd IEEE International Midwest Symposium on, Circuits and Systems (MWSCAS)*, 2010, pp. 620–623.

[7] P.V.C Hough, "Method and means for recognizing complex patterns," *US Patent 39696454*, 1962.

[8] Richard O. Duda and Peter E. Hart, "Use of the hough transformation to detect lines and curves in pictures," *Commun. ACM*, vol. 15, January 1972.

[9] John Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[10] Ivica Kostanic Fredric M. Ham, *Principles of Neurocomputing for Science and Engineering*, McGraw-Hill Higher Education, 2000.

[11] S. Setayeshi and F. El-Hawary, "Neural network based signal prediction and parameter estimation for underwater layered media systems identification," in *Electrical and Computer Engineering, 1994. Conference Proceedings. 1994 Canadian Conference on*, Sept. 1994, pp. 777–780 vol.2.

[12] S. Setayeshi, F. El-Hawary, and M.E. El-Hawary, "Underwater signal prediction and parameter estimation using artificial neural networks," in *Electrical and Computer Engineering, 1995. Canadian Conference on*, Sept. 1995, vol. 2, pp. 621–626 vol.2.

[13] I. T. Ruiz, S. de Ravencourt, Y. Petillot, and D. M. Lane, "Concurrent mapping and localization using sidescan sonar," *IEEE J. Oceanic Eng.*, vol. 29, no. 2, pp. 442–456, April 2004.

[14] X. Tian, D. Zhou, and Z. Liu, "Object recognition algorithm of sonar image," in *Signal Processing, 2006 8th International Conference on*, 2006, vol. 2.

[15] Wright J., Scott K., Tien-Hsin Chao, Lau B., Lathrop J., and McCormick J., "Multisensor data fusion for seafloor mapping and ordnance location," in *Autonomous Un-*

derwater Vehicle Technology, 1996. AUV '96., Proceedings of the 1996 Symposium on, 1996, pp. 167–175.

[16] Te-Chih Liu and Henrik Schmidt, "A framework of real-time seabottom target detection using acoustic sensors on auvs," in *OCEANS '04. MTTS/IEEE TECHNO-OCEAN '04*, 2004, vol. 4, pp. 2019–2023 Vol.4.

[17] C. Liu, Z. Zhang, and E. Sang, "A noval acoustical vision system design for automated underwater vehicles," in *Intelligent Control and Automation, 2008. WCICA 2008. 7th World Congress on*, 2008, pp. 7438–7443.

[18] J. Tian, S. Xue, and H. Huang, "Classification of underwater objects based on probabilistic neural network," in *Natural Computation, 2009. ICNC '09. Fifth International Conference on*, 2009, vol. 2, pp. 38–42.

[19] J.E. McFee, Y. Das, and R.O. Ellingson, "Locating and identifying compact ferrous objects," *Geoscience and Remote Sensing, IEEE Transactions on*, vol. 28, no. 2, pp. 182–193, Mar. 1990.

[20] F. Langner, C. Knauer, W. Jans, and A. Ebert, "Side scan sonar image resolution and automatic object detection, classification and identification," in *OCEANS 2009 - EUROPE*, May 2009, pp. 1–8.

[21] O. Postolache, J.M. Dias Pereira, P.S. Girao, and H. Ramos, "Smart flexible turbidity sensing based on embedded neural network," in *Sensors, 2006. 5th IEEE Conference on*, 2006, pp. 658–661.

[22] V. Murino and A. Trucco, "Three-dimensional image generation and processing in underwater acoustic vision," *Proceedings of the IEEE*, vol. 88, no. 12, pp. 1903–1948, Dec. 2000.

[23] R.N. Bracewell, *The Fourier Transform and its Applications*, Mcgraw-Hill Book Co, 2nd edition, 1986.

[24] S.L. Speidel, "Neural adaptive sensory processing for undersea sonar," *Oceanic Engineering, IEEE Journal of*, vol. 17, no. 4, pp. 341–350, Oct. 1992.

[25] V. Mitra, Chia-Jiu Wang, and S. Banerjee, "Lidar detection of underwater objects using a neuro-svm-based architecture," *Neural Networks, IEEE Transactions on*, vol. 17, no. 3, pp. 717–731, May 2006.