Multiresolution Framework with Neural Network Approach for Automatic Target Recognition

A DISSERTATION
Submitted in partial fulfillment of the requirements for the award of the degree of
MASTER OF TECHNOLOGY
in
INFORMATION TECHNOLOGY
(Specialization: INTELLIGENT SYSTEMS)

By
Dhirendra Pratap Singh

Under the Guidance of:
Dr. Anupam Agrawal
Associate Professor
IIIT-Allahabad

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY
ALLAHABAD – 211 011 (INDIA)
Automatic Target Recognition is an approach by which we identify one or a group of target-objects in a scene. It plays a pivotal role in the challenging fields of defense and civil. Most of the methods in this context are based on fix window-size technique.

In this thesis we proposed a novel approach which gives scale, rotation and translation invariant results for automatic target recognition in high-resolution satellite images which in turn are able to recognize the multiple targets in a scene. We make a system which can predict the possible area of interest in a scene, where target may be present or not. Prediction of areas of interest is based on edge detection and similarity measure of wavelet co-occurrence features of segmented sub-blocks. Proposed method uses a systematic approach for selecting features of area of interest.

Zernike moments, calculated for scale and translation normalized area of interest, are thereby used as the features of the concerned area. Zernike moments are rotation invariant. The extracted features are then fed to trained neural network for recognition. This approach is more suitable for the satellite images because resolution of image and idea about the target are two essential factors by which we can predict the minimum and maximum size of the target. The algorithm takes considerably less time compared to the fix window based approach because the predicted numbers of interest areas to be processed in a scene are very less. The proposed approach has successfully been tested on number of satellite images of different resolutions and their timing analysis has been compared with fix window based approach.
This thesis has been a very satisfying experience for me. I am sure that the knowledge and experience gathered in the course of this work will stand me in good stead in the future.

I am highly grateful to the honorable Director, IIIT Allahabad, Prof. M. D. Tiwari, for his ever helping attitude and encouraging us to excel in studies. Besides, he has been a source of inspiration during my entire period of M.Tech. at, IIITA.

I am thankful to Prof. U. S. Tiwari, Dean Academics, IIIT Allahabad for providing all the necessary requirements and for his moral support for this dissertation work as well during the whole course of M. Tech.

The most notable source of guidance was my advisor, Dr. Anupam Agrawal, Associate Professor, IIIT Allahabad. I owe his a great deal of thanks for taking me under his wing and allowing me to soak up some of his knowledge and insight. He has not only made us to work but guided us to orient towards research.

I am also thankful to my classmates for their cooperation during my work. I am also thankful to them for helping me in my project work and also some kind of discussion regarding my work which helps me to understand the concept regarding my work.

This acknowledgement will not complete until I pay my respectful homage to my family especially my parents, whose enthusiasm to see this work complete was as infectious as their inspiration.

Dhirendra Pratap Singh
# List of Figures and Tables

**Figures:**

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Different stages of target recognition</td>
<td>4</td>
</tr>
<tr>
<td>3.1</td>
<td>(a) Model of ideal digital edge (b) Model of a ramp digital image</td>
<td>9</td>
</tr>
<tr>
<td>3.2</td>
<td>(a) Gray-level profile (b) First derivative (c) Second derivative</td>
<td>10</td>
</tr>
<tr>
<td>3.3</td>
<td>(a) 1-level wavelet transform (b) 2-level wavelet transform</td>
<td>14</td>
</tr>
<tr>
<td>3.4</td>
<td>Neural network</td>
<td>23</td>
</tr>
<tr>
<td>3.5</td>
<td>Simple neuron</td>
<td>24</td>
</tr>
<tr>
<td>3.6</td>
<td>Hard-limit transfer function</td>
<td>25</td>
</tr>
<tr>
<td>3.7</td>
<td>Linear transfer function</td>
<td>25</td>
</tr>
<tr>
<td>3.8</td>
<td>Sigmoid transfer function</td>
<td>26</td>
</tr>
<tr>
<td>3.9</td>
<td>Perceptron architecture</td>
<td>27</td>
</tr>
<tr>
<td>3.10</td>
<td>Multilayer feed forward neural network</td>
<td>28</td>
</tr>
<tr>
<td>4.1</td>
<td>Proposed method</td>
<td>29</td>
</tr>
<tr>
<td>4.2</td>
<td>Original image</td>
<td>30</td>
</tr>
<tr>
<td>4.3</td>
<td>Image after edge detection</td>
<td>30</td>
</tr>
<tr>
<td>4.4</td>
<td>Image after selection of sub-blocks</td>
<td>31</td>
</tr>
<tr>
<td>4.5</td>
<td>Image after selection on AOI</td>
<td>32</td>
</tr>
<tr>
<td>4.6</td>
<td>Final image</td>
<td>33</td>
</tr>
<tr>
<td>4.7</td>
<td>Process flow chart</td>
<td>34</td>
</tr>
<tr>
<td>5.1</td>
<td>(a) Input image 1 (b) Output image 1</td>
<td>36</td>
</tr>
<tr>
<td>5.2</td>
<td>(a) Input image 2 (b) Output image 2</td>
<td>37</td>
</tr>
<tr>
<td>5.3</td>
<td>(a) Input image 3 (b) Output image 3</td>
<td>38</td>
</tr>
</tbody>
</table>
Fig 5.4  Size and response graph  ............................................. 39
Fig 5.5  Limitations  ................................................................. 40

Tables:

Table 3.1  (a) Gray image
          (b) Co-occurrence matrix for gray image  ......................... 17
Table 5.1  Overall recognition  .................................................. 40
Contents

Abstract ............................................................................................................................................. i
Acknowledgments .......................................................................................................................... ii
List of Figures and Tables ............................................................................................................... iii
Contents ............................................................................................................................................... v
1 Introduction ................................................................................................................................. 1
  1.1 Overview ................................................................................................................................... 1
  1.2 Problem Statement ................................................................................................................. 3
  1.3 Thesis Structure ..................................................................................................................... 3
2 Analysis of Past Work .................................................................................................................. 4
  2.1 Overview ................................................................................................................................... 4
  2.2 Approaches for Target Detection and Recognition ............................................................. 5
3 Methodologies Used ..................................................................................................................... 9
  3.1 Edge Detection ....................................................................................................................... 9
  3.2 Wavelet Transforms ............................................................................................................ 11
    3.2.1 Wavelet Series Expansions .......................................................................................... 11
    3.2.2 Discrete Wavelet Transforms ...................................................................................... 12
    3.2.3 Wavelet Transform in two Dimensions ........................................................................ 13
    3.2.4 Features of Wavelet Coefficients ................................................................................. 15
  3.3 Gray-Level Co-Occurrence Matrix ....................................................................................... 15
    3.3.1 Process Used to Create the GLCM ............................................................................. 16
    3.3.2 Texture Feature of Co-occurrence Matrix ................................................................. 17
  3.4 Zernike Moment ..................................................................................................................... 18
    3.4.1 Rotation Invariant Features ......................................................................................... 20
  3.5 Neural Network ..................................................................................................................... 22
    3.5.1 Simple Neuron .............................................................................................................. 24
    3.5.2 Transfer Functions .......................................................................................................... 25
    3.5.3 Perceptron Architecture .............................................................................................. 26
    3.5.4 Back Propagation Neural Network .............................................................................. 27
Multiresolution framework with Neural Network approach for Automatic Target Recognition

4 Proposed Method ................................................................. 29
  4.1 Image Segmentation .......................................................... 29
  4.2 Selection of Area of Interest .............................................. 31
  4.3 Feature Extraction and Target Classification ....................... 32
  4.4 Process Flow Chart ......................................................... 34
5 Simulation and Results .......................................................... 35
  5.1 Input and Output Images .................................................. 36
  5.2 Overall Recognition ....................................................... 40
6 Conclusion and Future Scope .................................................. 41
  6.1 Conclusion ................................................................. 41
  6.2 Future Scope ............................................................. 41
Appendix ‘A’ ........................................................................ 43
Reference ............................................................................. 45
1.1. Overview:

Computer vision is at present an active field of research, with the aim of developing visual sensing and processing algorithms that can see and understand the world around them. The main aim of computer vision is to describe an image in terms of the meaningful objects comprise it. Automatic Target Recognition is one of the most important applications of computer vision. Automatic Target Recognition is an approach by which we identify one or group of target objects in scene. It is very easy for a human to identify different objects in image but it’s difficult for a computer program to identify different objects. The problem is become challenging by the facts that computer; unlike people have not a priori information of the orientation and scale of the object.

Automatic target recognition is a challenging problem having potential application in defense and civil. Defense application includes design of smart weapon system, guiding pilot of high performance fighter aircraft. Civil applications are automated manufacturing industry, remote surgery in medical field.

The Automatic target recognition is used in monitoring disaster management during relief operation after calamity. Various natural disasters, namely drought, flood, cyclone, earth quake, landslide, forest fire, hail storm, locust, volcanic eruption etc, which strike causing a devastating impact on human life, economy and environment. Though it is almost impossible to fully compensate the damage caused by the disasters, but it is possible to (i) minimize the possible risks by developing early warning strategies (ii) prepare and implement developmental plans to provide resilience to such disasters (iii) to help in rehabilitation and post-disaster reconstruction.

Automatic Target Recognition system constitutes an important part of the future strategy reconnaissance system. In the modern war, how to recognize military target such
as airplane, naval ships, missile etc, is significant to win initiative in the battle field and to attack the goal accurately.

Automatic Target Recognition requires processing of two-dimensional images for detecting, classifying, and recognizing targets. The first stage in Automatic Target Recognition is the detection process. The detection step acts as a filter that focuses attention only certain regions of image. Detection process finds all possible areas of interest (AOIs) in image. In most of the images numbers of objects are connected to each other, these situations make the main problem to select appropriate AOIs, when the sizes of AOIs are not fixed.

Object recognition is often posed as a pattern recognition problem in supervised learning approaches. Training the object recognition system with images of the known objects and classifying the newly coming AOI into one of the classes is the main aspect of the target recognition system. Object recognition is hard job because there are wide ranges of conditions (including varying position, orientation, scale and illumination) under which an object may need to be classified.

Illumination refers to the amount of light falling on the scene. Variations in illumination can produce drastic changes in the grey levels or the color values of objects in the image. These changes can change the appearance of the object and thus making recognition difficult. Complex backgrounds in images can cause a lot of misclassifications while recognition an object.

An Automatic Target Recognition system must be invariant toward vantage point differences. These differences include illumination changes, shadowing, noise, and occlusion. Normally, the target recognition process is highly data dependent. Most systems are only able to recognize a pre-specified number of targets and are unable to expand their object database. In addition, many Automatic Target Recognition systems are encoded with predetermined tolerances in that they tend to be very sensitive to scale and orientation changes.

Target recognition problems are classified according to three different criteria.

1. The number of classes of target.
2. The input of the classifier.
3. The number of independent stages used in recognition problem.
1.2. Problem Statement:

The main goal of this thesis is to develop an approach for automatic target recognition which is scale, rotation and translation invariant and able to recognize the multiple targets in any scene. For which:

Make a system which can predict the possible areas of interest in scene, where target may be present or not. There is no need to search the complete image to find the target because we know that targets are present only some parts of image.

Propose a systematic approach for selecting features of AOI which are used for classification, by which we can achieve high classification performance in any given image and which is also able to classify objects in a new scene without requiring further supervised training.

Demonstrate an object recognition system based on supervised learning.

1.3. Thesis Structure:

This thesis is divided into six chapters (including this introduction), appendix and references. Chapter 2 analyses the past work in the area of target detection and target recognition. Chapter 3 gives the basic idea about the methodologies which are used to implement the proposed approach of automatic target recognition. Chapter 4 introduces the method proposed for Automatic target recognition in high resolution satellite images, along with the development of the work. Chapter 5 shows the results obtained after simulation of this approach. Chapter 6 contains the conclusion of the work presented and the direction that the future work will take.
2.1. Overview:

There are two approaches which are mainly used in automatic target recognition, one is pixel based approach and second is feature based approach. In pixel based approach pixels intensities are used for target recognition and in feature based approach first we calculate the features of area of interest and then these features are used for target recognition.

Most work in target recognition involves four stages: preprocessing, segmentation, feature extraction, and classification, as shown in figure 2.1.

![Diagram](image)

Figure 2.1 Different stages of Target recognition [3]

The preprocessing stage aims to remove noise or enhance edges. The segmentation stage is useful for dividing the image space into separate regions of interest. The feature extraction system extracts key features of area of interest. The result of features extraction is normally a feature vector. Classification involves classification and identifying the target.
The segmentation process generates separate area of interest. These areas of interest represent a portion of the image space and are used to search for targets in a specified location. Generally an area of interest is created though utilizing a moving window. This allows us to systematically search an image in an incremental fashion and allow multi target detection.

In most of the image-processing applications, a classical technique used to find the similarity between a pair of images is to evaluate correlation measure between the images [4]. A pattern that is close to the reference image pattern produces a higher signal in the correlation plane than the patterns that is far. Based on this fact, we can utilize a correlation scheme which tightly clusters together related targets while excluding non-target objects. We can accomplish this task by implementing a procedure which cleverly extracts the features needed for characterizing an object as a target.

2.2. Approaches for Target Detection and Recognition:

ANVIL [1] is a 3-D multiple object detection system which uses a neural network to detect, locate, and identify multiple targets in any scene. The neural network in ANVIL is used to separate the features of targets and background images. The neural network is trained on 11x11 pixel-sized features. After this learning phase, a set of features is selected and an associative mapping phase associates features with images.

ANVIL is able to provide robust detection, high classification, identification and location accuracy as well as a low false alarm rate. The main limitation with this approach is that the system is not scale and rotation invariant. The system does not produce the same response to an identical target if there are changes in its size or orientation. Along with that, the assumed feature size of 11x11 pixels does not ensure that all relevant and necessary features can be extracted.

Daniell [2] performs automatic target recognition with the help of three-stage process. It initially using an edge detector algorithm to binarizes target boundary information and separates it from background by. The object image is then presented into a multilayer feed forward neural network that associates the input image with a cluster of feature characteristics. The final stage then utilizes a neural network to classify an object based on these associated features.
The main problem associated with Daniell’s approach is that it is assumed a simple edge detector algorithm is capable for segmenting objects from background. Based on this assumption, not only is the system not robust when faced with changes in lighting conditions, but objects which are not targets, such as rocks and trees, would also be processed as possible targets. In addition, the Neocognitron architecture used to associate input images with features requires additional layers if more information is required for classification.

In the method proposed by Howard [4], the input is an image acquired by video camera. Here first partition the input image into smaller areas of interest (AOIs). These AOIs give them the ability to systematically search for targets in an incremental fashion. Once these AOIs are extracted, they look for relevant features in the segmented image space which will signify that an object is located in the region. They accomplish this task by projecting the extracted image onto an object eigen-space. The resultant vector is then fed into a multi-stage neural network which allows the detection of targets/non-targets. The multi-stage neural network contains two neural networks which train in serial, but run in parallel. The use of a dual neural network allows for a very high target recognition rate while still maintaining an almost zero false positive alarm rate.

Neural network (NN) possesses advantages as adaptivity, parallelism, robustness, optimality, so that it has applied widely on challenging Problems arising in image processing and pattern recognition. However, as the same as other methods, NN is also short of solving the uncertainty in image processing and pattern recognition. To make up this shortage, NN integrating with fuzzy set theory methods are proposed [4], so as to depict the uncertainty by the ability of fuzzy set theory. It uses fuzzy measure as objective function to correct NN’s weights adaptively. By which the method can capably consider uncertainty in the procedure of target detection and classification and optimize it. SCBFNN [3] possesses good ability to automatic target detection, as well possesses valid capabilities to eliminating uncertainty and retaining target shape compared with conventional neural network methods. This method integrating Neural Network with Fuzzy Set theory uses fuzzy measure as objective function. While weighting processing in the neighborhood targets, structure context information is uses to restrict the weighting, so that targets shape and outline features can be reserved better.
The main problem with these approaches is image Segmentation, i.e. how we find the areas of interest in the scene. If we select an initial window of fixed size and move it from one end to another to find areas of interest, it will be time consuming process and object size will fix.

In the method proposed by Ganesan [5] target detection is achieved by calculating co-occurrence matrix features from detail sub-bands of discrete wavelet transformed, nonoverlapping but adjacent sub-blocks of different sizes, depending upon the target image. After features calculation, the sub-block with the maximum of combined wavelet co-occurrence feature values is selected as a seed window. Then, by applying a region growing algorithm, the sub-blocks or regions are grouped into a larger block or region based on some predefined criteria. Then, the target is identified by a bounded rectangle. The proposed algorithm is applied on both man-made and non-man made single image.

The success of most computer vision problems depends on how effectively the texture is quantitatively represented [7, 8]. Regardless of whether the application is target detection, object recognition, texture segmentation, or edge detection, one must be able to recognize and label homogeneous texture regions within an image and differentiate between distinct regions. Thus, texture analysis is one of the most important techniques used in the analysis and interpretation of images, consisting of repetition of some fundamental image elements.

Analysis of textures requires the identification of proper attributes or features that differentiate the textures in the image for segmentation, classification, and recognition. The features are assumed to be uniform within the regions containing the same textures. Initially, texture analysis was based on the first-order or second-order statistics of textures. New methods [9, 10] are based on multiresolution or multichannel analysis, such as Gabor filters and wavelet transform, have received a lot of attention but, the outputs of Gabor filter banks are not mutually orthogonal, which may result in a significant correlation between texture features. Finally, these transformations are usually not reversible, which limits their applicability for texture synthesis. Most of these problems can be avoided if one uses the wavelet transform, which provides a precise and unifying framework for the analysis and characterization of a signal at different scales.
To find the high recognition rate in system, the choice of feature extraction is very crucial. Hse and Neuton [17] use the Zernike moment feature for handwritten symbol recognition which gives scale, rotation and translation invariant results. Hong [16] uses the Zernike moment feature for invariant object recognition.
Chapter 3

Methodologies Used

3.1. Edge Detection:

An edge is a set of connected pixels that lie on the boundary between two regions. Fundamentally an edge is a local concept whereas a region boundary, owing to the way it is defined, is a more global idea. A reasonable definition of edge [22] requires the ability to measure gray level transition in a meaningful way. Most likely an edge has the properties of the model shown below in figure 3.1. An ideal edge according to this model is a set of connected pixels, each of which is located is an orthogonal step transition in gray level.

![Figure 3.1](image)

(a) Model of ideal digital edge                                (b) Model of a ramp digital edge

The slope of the ramp is inversely proportional to the degree of blurring in the image. The thickness of the edge is determined by the length of the ramp.
Figure 3.2(a) shows a horizontal gray level profile of the edge between two regions and first and second derivatives of the gray level profile. The first derivative is positive at the points of transition into and out of the ramp as we move from left to right along the profile; it is constant for points in the ramp; and zero in areas of constant gray level. The second derivative is positive at the transition associated with the dark side of the edge, negative at the transition associated with the light size of the edge, and zero along the ramp and in areas of constant gray level. The signs of the derivatives would be reserved for an edge that transitions from light to dark.

![Figure 3.2](image)

We find from these observations that the magnitude of the first derivative can be used to detect the presence of an edge at a point in an image, similarly, the sign of the second
derivative can be used to determine whether an edge pixel lies on the dark or light side of an edge.

### 3.2. Wavelets Transform:

Multiresolution [24] is used to define the human ability to view and comprehend phenomena on different scales. The first component to multiresolution analysis is vector spaces. For each vector space, there is another vector space of higher resolution until you get to the final image. Also, each vector space contains all vector spaces that are of lower resolution. The basis of each of these vector spaces is the scale function for the wavelet. For practical purposes one can think of an image as a vector space such as $V^0$ would be the perfectly normal image, and $V^{j-1}$ would be that image at a lower resolution until you get to $V^0$ where you just have one pixel in the entire image.

The wavelet transform is a multiresolution technique, which can be implemented as a pyramid or tree structure and is similar to sub-band decomposition. The discrete wavelet transform (DWT) [5] has properties that make it an ideal transform for the processing of images encountered in target recognition applications, including rapid processing, a natural ability to adapt to changing local image statistics, efficient representation of abrupt changes and precise position information, ability to adapt to high background noise and uncertainty about target properties, and a relative independence to target-to-sensor distance.

#### 3.2.1. Wavelet Series Expansions [22]:

The wavelet series expansion of function $f(x) \in L^2(R)$ relative to wavelet $\psi(x)$ and scaling function $\phi(x)$ can write as:

$$f(x)=\sum_{k} c_{j_0}(k) \phi_{j_0,k}(x) + \sum_{j=j_0}^{\infty} \sum_{k} d_{j}(k) \psi_{j,k}(x)$$

(3.1)

Where $j_0$ is an arbitrary starting scale and the $c_{j_0}(k)$’s are called the approximation or scaling coefficients; the $d_j(k)$’s are referred to the detail or wavelet coefficients. This is
because the first sum in Eq.(3.1) uses scaling function to provide an approximation of 
\( f(x) \) at scale \( j_0 \). For each higher scale \( j \geq j_0 \) in the second sum, a finer resolution function –a 
sum of wavelets is added to the approximation to provide increasing detail. If the 
expansion functions form an orthonormal basis frame, which is often the case, the 
expansion coefficients are calculated as:

\[
C_{j_0}(k) = \left[ f(x), \varphi_{j_0,k}(x) \right] = \int f(x) \varphi_{j_0,k}(x) \, dx \tag{3.2}
\]

And

\[
d_j(k) = \left[ f(x), \psi_{j,k}(x) \right] = \int f(x) \psi_{j,k}(x) \, dx \tag{3.3}
\]

### 3.2.2. Discrete Wavelet Transforms [22]:

Wavelet series expansion maps a function of a continuous variable in a sequence of 
coefficients. If the function being expanded is a sequence of numbers, like sample of a 
continuous function \( f(x) \), the resulting coefficients are called discrete wavelet transform 
(DWT) of \( f(x) \). For this case, the series expansion defined in Eqs.(3.1) through (3.3) 
become the DWT transform pair:

\[
W_\varphi(j_0,k) = \frac{1}{\sqrt{M}} \sum_x f(x) \varphi_{j_0,k}(x) \tag{3.4}
\]

\[
W_\psi(j,k) = \frac{1}{\sqrt{M}} \sum_x f(x) \psi_{j,k}(x) \tag{3.5}
\]

Here, \( f(x) \), \( \varphi_{j_0,k}(x) \) and \( \psi_{j,k}(x) \) are function of the discrete variable \( x=0,1,2,\ldots,M-1. \) The 
\( W_\varphi(j_0,k) \)'s and \( W_\psi(j,k) \)'s are correspond to the \( C_{j_0}(k) \)'s and \( d_j(k) \)'s of the wavelet series 
expansion.
3.2.3. Wavelet Transform in Two Dimensions:

The one dimensional transforms of the previous section are easily extended to two dimensional functions like images [22]. In two dimensions, a two dimensional scaling function, \( \phi(x,y) \) and three two dimensional wavelets, \( \psi^H(x,y), \psi^V(x,y) \) and \( \psi^D(x,y) \) are required. Each is the product of a one dimensional scaling function \( \phi \) and corresponding wavelet \( \psi \). Excluding products that produce one dimensional result, like \( \phi(x) \psi(x) \), the four remaining products produce the separable scaling function

\[
\phi(x,y) = \phi(x)\phi(y)
\]  

and separable, “directionally sensitive” wavelets

\[
\psi^H(x,y) = \psi(x)\phi(y) \]  
\[
\psi^V(x,y) = \phi(x)\psi(y) \]  
\[
\psi^D(x,y) = \psi(x)\phi(y).
\]  

These wavelets measure functional variations –intensity or gray-level variations for image –along different directions: \( \psi^H \) measures variations along columns, \( \psi^V \) measures variations along rows, and \( \psi^D \) measures variations along diagonals. The scale and transform basic functions in two dimensions are define as below.

\[
\phi_{j,m,n}(x,y) = 2^{j/2} \phi(2^jx-m,2^jy-n)
\]  
\[
\psi^i_{j,m,n}(x,y) = 2^{j/2} \psi^i(2^jx-m,2^jy-n)
\]  

\( i = \{H, V, D\} \).

Where index \( i \) identifies the directional wavelets in equation. The discrete wavelet transform of function \( f(x, y) \) of size \( M \times N \) is then

\[
W_{\phi}(j_0,m,n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \phi_{j_0,m,n}(x,y)
\]
Multiresolution framework with Neural Network approach for Automatic Target Recognition

\[
W^i_{\psi}(j,m,n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \psi^i_{j,m,n}(x,y)
\]

(3.13)

As in one dimensional case, \(j_0\) is an arbitrary starting scale and the \(W_{\psi}(j_0,m,n)\) coefficients define an approximation of \(f(x, y)\) at scale \(j_0\). The \(W^i_{\psi}(j, m, n)\) coefficients add horizontal, vertical, and diagonal details for scales \(j \geq j_0\).

In DWT [5, 8, 9] the image is actually decomposed, i.e., divided into four sub bands and critically sub sampled by applying DWT as shown in Figure 3.3(a). These sub-bands labeled LH1, HL1 and HH1 represent the finest scale wavelet coefficients, i.e., detail images while the sub band LL1 corresponds to coarse level coefficients, i.e., approximation image. To obtain the next coarse level of wavelet coefficients, the sub band LL1 alone is further decomposed and critically sampled. These results in two-level wavelet decomposition as shown in Figure 3.3(b) similarly, to obtain further decomposition, LL2 will be used. This process continues until some final scale is reached. The values or transformed coefficients in approximation and detail images (sub-band images) are the essential features, which are shown here as useful for texture analysis and discrimination. As micro textures or macro textures have non-uniform pixel value variations, they are statistically characterized by the features in approximation and detail images or in other words, the values in the sub-band images or their combinations or the derived features from these bands uniquely characterize a texture. The features obtained from this wavelet transformed images to be used for texture analysis, classification.

(a) 1-level wavelet transform,     (b) 2-level wavelet transform [8]

Figure 3.3[5]
3.2.4. Features of Wavelet Coefficients:

The wavelet coefficients are highly correlated with each other [11, 12, 13]. This correlation, mainly caused by features such as lines, edges, and corners, arises between neighboring coefficients in a given sub band as well as between coefficients corresponding to different scales and orientations. Work has been done to include the inter-sale and intra-scale dependencies between the wavelet coefficients. It is shown that intra-scale models capture most of the dependencies between the wavelet coefficients, and the gains obtained by including the inter-scale dependencies are marginal. These dependencies can be calculated with the help of statistical features of wavelet coefficients like mean and standard deviation.

Mean:

\[
\bar{b} = \sum_{b=0}^{L-1} b P(b)
\]  

(3.14)

Standard deviation:

\[
S_b = \left[ \sum_{b=0}^{L-1} (b - \bar{b})^2 P(b) \right]^{1/2}
\]  

(3.15)

\(b\) represent the gray level of image.

The two-dimensional joint probability density function known as the co-occurrence matrix is used to represent the spatial correlation between the wavelet coefficients. The co-occurrence matrix is widely used in texture and image classification and segmentation. Several parameters associated with the co-occurrence matrix [13, 14] are inertia, total energy, entropy, cluster prominence, max. Probability, cluster Shade, cluster shade, local homogeneity.

3.3. Gray-Level Co-Occurrence Matrix:

A statistical method that considers the spatial relationship of pixels is the gray-level co-occurrence matrix (GLCM), also known as the gray-level spatial dependence matrix.

Indian Institute Of Information Technology, Allahabad
The co-occurrence [14, 15] method of texture description is based on the repeated occurrence of some gray-level configuration in the texture and this configuration varies rapidly with distance in fine textures and slowly in coarse textures.

GLCM considers the relation between two pixels at a time, called the reference and the neighbor pixel. Consider the part of textured image to be analyzed is of size N*N. In the co-occurrence matrix, an occurrence of some gray-level configuration is described by a matrix of relative frequencies $C_{\theta,x}(i,j)$ describing how frequently two pixels with gray levels i, j appear in the window separated by a displacement vector d in direction $\theta$.

For example [5], if the displacement vector is specified as (1, 1), it has the interpretation of one pixel below and one pixel to the right, in the direction of 45 and if it is specified as (1, −1), it has the interpretation of one pixel below and one pixel to the left, in the direction of 135. Similarly, the displacement vector (0, 1) has the interpretation of zero pixel below and one pixel to the right, that is, in the direction of 0 and the displacement vector (1, 0) has the interpretation of one pixel below and zero pixel to the left, that is, in the direction of 90. These co-occurrence matrices are symmetric if defined as given below. However, an asymmetric definition may be used, where matrix values are also dependent on the direction of co-occurrence. Normalized frequencies of co-occurrence as functions of angle and distance can be represented as:

$$C_{\theta,x}(i,j) = \text{card}\{((s,t),(u,v)) : I(s,t) = i, I(u,v) = j, d((s,t),(u,v)) = x, \tan^{-1} ((s,t) - (u,v)) = q \}$$

(3.16)

### 3.3.1. Process Used to Create the GLCM:

The table (3.4) [Matlab help] shows how gray level co-occurrence matrix (GLCM) calculates the values from the gray image. This GLCM is calculated for zero pixels below and one pixel to the right of input image. In the output GLCM, element (1, 1) contains the value 1 because there is only one instance in the input image where two horizontally adjacent pixels have the values 1 and 1, respectively. GLCM (1, 2) contains the value 2 because there are two instances where two horizontally adjacent pixels have the values 1 and 2.
Multiresolution framework with Neural Network approach for Automatic Target Recognition

(a) Gray image

(b) Co-occurrence matrix for gray image

Table 3.1

3.3.2. Texture Feature of Co-occurrence Matrix:

These features [5] which are shown below are subjected to either linear or logarithmic normalization, depending on their dynamic ranges. The contrast features have moderate values and hence they are subjected to linear normalization, while cluster shade and
cluster prominence are subjected to logarithmic normalization, since they have very large
dynamic range of values.

\[
\text{Contrast} = \sum_{i,j=1}^{N} (i-j)^2 c(i,j)
\]
(3.17)

\[
\text{Cluster Shade} = \sum_{i,j=1}^{N} (i-M_x + j-M_y)^3 c(i,j)
\]
(3.18)

\[
\text{Cluster Prominence} = \sum_{i,j=1}^{N} (i-M_x + j-M_y)^3 c(i,j)
\]
(3.19)

\[
M_x = \sum_{i,j=1}^{N} i c(i,j) \quad M_y = \sum_{i,j=1}^{N} j c(i,j)
\]
(3.20)

3.4. Zernike Moment:

Moment features captures global information about the image [16, 17]. Regular
moments are defined as:

\[
m_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x, y) \, dx \, dy
\]
(3.21)

Where \( m_{pq} \) is the \((p + q)\) th order moment of the continuous image function \( f(x, y) \).
For digital images the integrals are replaced by summations and \( m_{pq} \) becomes

\[
m_{pq} = \sum \sum x^p y^q f(x, y) \, dx \, dy
\]
(3.22)
Hu [17] introduced seven nonlinear functions defined on regular moments which are translation, scale, and rotation invariant. These seven so-called moment invariants were used in a number of pattern recognition problems.

Zernike moments used in this study are a class of such orthogonal moments. The reason for selecting them from among the other orthogonal moments is that they possess a useful rotation invariance property. Rotating the image does not change the magnitudes of its Zernike moments. Hence, they could be used as rotation invariant features for image representation. These features could easily be constructed to an arbitrary high order.

The Zernike moments are only rotation invariant. To obtain scale and translation invariance, the image is first subjected to a normalization process using its regular moments. The rotation invariant Zernike features are then extracted from the scale and translation normalized image.

The Zernike polynomials were first proposed in 1934 by Zernike. Complex Zernike moments are constructed using a set of complex polynomials which form a complete orthogonal basis set defined on the unit disc \( (x^2 + y^2) \leq 1 \). They are expressed as \( A_{mn} \) two-dimensional Zernike moments:

\[
A_{mn} = \frac{m+1}{\pi} \int x y f(x,y)[V_{mn}(x,y)]^* \, dx \, dy 
\]  

(3.23)

Where:

\[
x^2 + y^2 \leq 1
\]

Where \( m = 0, 1, 2, \ldots, \infty \) and defines the order, \( f(x, y) \) is the function being described and * denotes the complex conjugate. While \( n \) is an integer (that can be positive or negative) depicting the angular dependence, or rotation, subject to the conditions:

\[
m - |n| = \text{even}, \ |n| \leq m
\]

(3.24)

The Zernike polynomials expressed in polar coordinates are
Multiresolution framework with Neural Network approach for Automatic Target Recognition

\[ V_{mn}(r, \theta) = R_{mn}(r) \exp(jn\theta) \]  

(3.25)

Where \((r, \theta)\) are defined over the unit disc, \(j = \sqrt{-1}\) and \(R_{mn}(r)\) is the orthogonal radial polynomial, defined as:

\[
R_{mn} = \sum_{s=0}^{m-|n|/2} (-1)^s F(m,n,s,r)
\]

(3.26)

Where:

\[
F(m,n,s,r) = \frac{(m-s)!}{s!(m-|n|/2-s)!(m-|n|/2-s)!} r^{m-2s}
\]

(3.27)

To compute the Zernike moments of a given image, the center of the image is taken as the origin and pixel coordinates are mapped to the range of unit circle, i.e., \(x^2 + y^2 \leq 1\). Those pixels falling outside the unit circle are not used in the computation.

3.4.1. Rotation Invariant Features [17]:

Consider a rotation of the image through angle \(\alpha\). If the rotated image is denoted by \(f'\), the relationship between the original and rotated images in the same polar coordinates is

\[
f'(r, \theta) = f(r, \theta - \alpha)
\]

(3.28)

The Zernike moment expression can be mapped from the xy-plane into the polar coordinates by changing the variables in double integral form of Eq.3.23. This can be seen from Eq. 3.28.
\[
\int_A \int \phi(x,y) \, dx \, dy = \int_{G'} \int \phi[p(r,\theta),q(r,\theta)] \frac{\delta(x,y)}{\delta(r,\theta)} \, dx \, dy = \int_{G'} \int \phi[p(r,\theta),q(r,\theta)] \delta(x,y) \, dx \, dy
\]

(3.29)

Where \( \delta(x,y)/\delta(p,\theta) \) denote the Jacobian of the transformation and are the determinant of the matrix:

\[
\frac{\delta(x,y)}{\delta(r,\theta)} = \begin{bmatrix}
\frac{\delta x}{\delta r} & \frac{\delta x}{\delta \theta} \\
\frac{\delta y}{\delta r} & \frac{\delta y}{\delta \theta}
\end{bmatrix}
\]

(3.30)

For this case where \( x = r \cos \phi \) and \( y = r \sin \phi \), the Jacobian becomes \( r \). Hence

\[
A_{mn} = \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 f(r,\theta) V_{mn}^*(r,\theta) r \, dr \, d\theta
\]

(3.31)

\[
= \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 f(r,\theta) R_{mn}(r) \exp(-jm\theta) r \, dr
\]

(3.32)

The Zernike moment of the rotated image in the same coordinate is:

\[
A'_{mn} = \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 f(r,\theta-\alpha) R_{mn}(r) \exp(-jm\theta) r \, dr \, d\theta
\]

(3.33)

By a change of variable \( \phi_1 = \phi - \alpha \).
\[ A_{mn}' = \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 f(r, \theta_1) R_{mn}(r) \exp(-jm(\theta_1 + \pi)) r \, dr \, d\theta \]

\[ = \frac{n+1}{\pi} \int_0^{2\pi} \int_0^1 f(r, \theta_1) R_{mn}(r) \exp(-jm\theta_1) r \, dr \, d\theta \exp(-jm\pi) \]

\[ = A_{mn} \exp(-jm\pi) \]

Equation 3.36 shows that Zernike moments have simple rotational transformation properties; each Zernike moment merely acquires a phase shift on rotation. This simple property leads to the conclusion that the magnitudes of the Zernike moments of a rotated image function remain identical to those before rotation. Thus the magnitude of the Zernike moment can be taken as a rotation invariant feature of the underlying image function.

### 3.5. Neural Network:

A neural network is an information-processing system that has been developed as generalizations of mathematical models matching human cognition. They are composed of a large number of highly-interconnected processing units (neurons) that work together to perform a specific task. According to Haykin [18], a neural network is a massively parallel-distributed processor that has a natural prosperity for storing experimental knowledge. It resembles the brain in two respects:

- Knowledge is acquired by the network through a learning process;
- Inter-connected connection strengths known as synaptic weights are used to store the knowledge.
A neuron is considered to be an adaptive element. Its weights are modified during the process of learning. This is influenced by the input signal it receives, its desired output, and its associated response to the input. Depending on the nature of a task, the user can choose between two learning techniques: supervised and unsupervised learning.

Supervised learning involves a ‘teacher’ which provides the desired response or data for the network to train. The distance between the actual and the desired output vectors serves as an error measure that is used to correct adaptation parameters. The network’s weights adapt to decrease this error.

In an unsupervised learning environment, the desired response is not known. Learning is based on observations of responses to inputs. The network discovers for itself any existing irregularities or properties leading to changes in its parameters.

Commonly neural networks are adjusted, or trained, so that a particular input leads to a specific target output. Such a situation is shown in figure 3.4. There, the network is adjusted, based on a comparison of the output and the target, until the network output matches the target. Typically many such input/target pairs are needed to train a network.

Neural networks have been trained to perform complex functions in various fields, including pattern recognition, identification, classification, speech, vision, and control systems.
3.5.1. Simple Neuron [19]:

A neuron with a single scalar input and bias appears on the below.

\[ a = f(wp + b) \]

The scalar input \( p \) is transmitted through a connection that multiplies its strength by the scalar weight \( w \) to form the product \( wp \), again a scalar. Here the weighted input \( wp \) is the only argument of the transfer function \( f \), which produces the scalar output \( a \). The neuron on the right has a scalar bias \( b \). You can view the bias as simply being added to the product \( wp \) as shown by the summing junction or as shifting the function \( f \) to the left by an amount \( b \). The bias is much like a weight, except that it has a constant input of 1.

The transfer function net input \( n \), again a scalar, is the sum of the weighted input \( wp \) and the bias \( b \). This sum is the argument of the transfer function \( f \). Here \( f \) is a transfer function, typically a step function or a sigmoid function, that takes the argument \( n \) and produces the output \( a \).

Note that \( w \) and \( b \) are both adjustable scalar parameters of the neuron. The central idea of neural networks is that such parameters can be adjusted so that the network exhibits some desired or interesting behavior. Thus, you can train the network to do a particular job by adjusting the weight or bias parameters, or perhaps the network itself will adjust these parameters to achieve some desired end.
3.5.2. Transfer Functions:

Three of the most commonly used functions are shown below.

The hard-limit transfer function shown above limits the output of the neuron to either 0, if the net input argument \( n \) is less than 0, or 1, if \( n \) is greater than or equal to 0. This function is used in Perceptrons, to create neurons that make classification decisions.

The linear transfer function is shown below.
Neurons of this type are used as linear approximators in Linear Filters. The sigmoid transfer function shown below takes the input, which can have any value between plus and minus infinity, and squashes the output into the range 0 to 1.

\[ a = \text{logsig}(n) \]

This transfer function is commonly used in back-propagation networks, in part because it is differentiable.

**3.5.3. Perceptron Architecture:**

The perceptron network consists of a single layer of S perceptron neurons connected to R inputs through a set of weights \( w_{ij} \), as shown below. As before, the network indices i and j indicate that \( w_{ij} \) is the strength of the connection from the jth input to the ith neuron.
3.4.5. Back Propagation Neural Network:

The feed forward, back-propagation architecture was developed in the early 1970’s by several researchers independently. Currently this is the most popular Neural Network architecture being used.

A typical Back-Propagation Neural Network will have an input layer, an output layer, and at least one hidden layer. There is no theoretical limit on the number of hidden layers. It depends on the purpose for which is being constructed. But it has been proved that a maximum of 3 to 4 layers is enough even the most complicated pattern recognition tasks. Each layer of this network is fully connected to the next layer. There are no
interconnections in neurons of the same layer which makes it a Feed Forward model (Figure 3.10).

![Multilayer Feed Forward Neural Network](image)

Figure 3.10 Multilayer Feed Forward Neural Network

There are many forms and variations of the Back Propagation Algorithm but the most common form is given by the equations 3.37 through 3.39.

\[
\begin{align*}
h_j &= \frac{1}{1 + e^{-\sum_{i=0}^{A} W(n-1)_{ij} X_i}} \\
\delta_j(n) &= h_j(1-h_j) \sum_{i=0}^{C} \delta_j(n+1)W(n+1)_{ji} \\
\Delta W(n)(t+1)_{ij} &= \eta\delta_j(n)h_j + \alpha \Delta W(n)(t)_{ij}
\end{align*}
\]

(3.37)  
(3.38)  
(3.39)

The algorithm can be summarized as given below.

1. Find the Layer Activations $h_j$.
2. Calculate Error between input and target.
4. Update Weights of each layer using Eq 3.39.
Chapter 4

Proposed Method

In the proposed approach the solution to the Automatic Target Recognition problem is divided into 4 parts as shown in the figure 4.1.

1. Image segmentation - To divide the image into small sub-parts.
2. Selection of AOI - By grouping sub-parts.
3. Feature extraction - By using the Zernike moment.
4. Recognition of Targets - With the help of trained Neural Network.

4.1. Image Segmentation:

We take an input image and change its size to a number which is a factor of 8 because in the next step we will divide the image into 8*8 or 4*4 pixels size sub-images. If the image is colored change it to gray-scale image. Now we apply the canny edge detection algorithm to find the edges in image and then divide the edge-detected image in 8*8 or 4*4 pixels size non-overlapping sub-blocks, according to the resolution of the image. From these sub image blocks select those blocks which are having number of edge pixels more than this limit - for 8*8 pixels sub-images limit is 7 pixels and for 4*4 pixels sub-
images limit is 3, as shown in figure 4.4. Initially assign the group number 0 and find all the adjacent blocks of each selected block. Find the row and column number of each selected block.

Figure 4.2 (Original image)

Figure 4.3 (Image after edge detection)
4.2. Selection of Area of Interest:

In this method we find those areas which may have target object, because all the image parts are not having objects. So we are going to search only those areas, which may have any object. These areas are called area of interest (AOI). Here selection of AOI is based on the grouping of selected sub-blocks. Make groups of selected sub-blocks which are adjacent to each other. From these groups selection of appropriate groups depending on size of object in any particular resolution. Avoid those groups which are created due to big noisy edges, as shown in the figure 4.3 where number of big noisy edges is selected in edge detection algorithm. These edges are selected on the basis of change in reflectance from background. Also avoid those groups, where the number of selected sub-blocks is very less but the size of AOI is comparatively large.

Select only those groups which are bigger than the minimum size of object and make AOI around those groups which satisfy the size criteria of object as in figure 4.2 (minimum size is 32*32 pixels and maximum is 72*72 pixels).
Take those groups where AOI is bigger than maximum size of object and again do the regrouping of blocks of these groups, on the basis of its texture features. To calculate the texture feature first find 1 level DWT of each block by using DB2 function and make wavelet co-occurrence matrices that are derived for $\sigma=135$ and $d= (1, 1)$ (i.e. one pixel below and one pixel to right) for detail sub bands (i.e. LH, HL, HH). Then from these co-occurrence matrices compute texture features such as contrast, cluster shade and cluster prominence and normalize these features. On the basis of these texture features, regroup those blocks which are adjacent to each other and having same combined texture features value. Then from these new groups select those groups which satisfy the size criteria of object and make AOI around them.

In figure 4.5 white boxes are areas after simple grouping of sub-blocks and black boxes are AOIs after regrouping.

4.3. Feature Extraction and Target Classification:

After finding the area of interest, we are calculating the Zernike moment of selected area of interest. For the selection of appropriate order of Zernike moment we tested
Multiresolution framework with Neural Network approach for Automatic Target Recognition

different orders and try to find the order for which we get exact reconstruction of image. On the basis of this test order 1 to 30 is selected. So calculate the Zernike moment of order 1 to 30 of each selected area of interest and normalize these features. Zernike moments are not invariant to scale and translation, therefore AOI is first scaled and translation normalized such that they are of the same dimension and their centroids are positioned at the origin. Now these moment features are scale, rotation and translation invariant, which are used as a feature vector for the target image.

For the classification we created a 3 layer neural network. This neural network has 255 neurons in input layer, 30 neurons in middle layer and one neuron in output layer. It is using “tansig” function. Normalized moment features are applied to the input of the neural network. In figure 4.7 red color boxes are showing the recognize target objects

![Figure 4.7(Final image)](image)

Indian Institute Of Information Technology, Allahabad
4.4. Process Flow Chart:

This flow chart shows the all steps in this proposed approach.

Input image

Apply edge detection

Divide new image

Select the blocks

Make group of blocks

Make AOI

Extract features of AOI

Apply train NN for recognition

Presence/Absence Of target

Figure 4.6
This approach is implemented in Matlab 7.2. The implemented approach is tested on satellite images of different resolutions, which are having airplanes and airplanes are selected as target. These images can have multiple target and non-target objects. This approach is appropriate for the satellite images because on the basis of the resolution of image and basic idea about the target, we can predict the minimum and maximum size of targets. Zernike moment is implemented using “LANS Pattern Recognition Toolbox [23]”. For the selection of appropriate order of Zernike moment we tested different orders. Order range 1 to 30 has been selected that yields exact reconstruction of image. To find scale and translation invariant Zernike moments, first we made AOIs scale and translation normalized. Here all AOIs are resized and become equal in size for scale normalization and for translation normalization, it calculate the centroid of AOI and make this as origin of AOI. For the training of neural network we are using 60 AOIs of different size which are containing possible target or non-target objects. Test images are “EROS” satellite images [25]. Obtained sample of the resultant images, are as shown in figure 3.1 through 3.3 where red boxes show the AOIs which is recognized as target.

Our approach is able to recognize the multiple objects in any scene. It gives scale, rotation and translation invariant target recognition results. Output images show that all the selected AOIs are of different size and our approach is able to recognize the target accurately. This confirms that it is a scale invariant approach. Objects that are present in selected AOIs are having different rotation and translation and those are recognized by the system which manifests rotation and translation invariant properties of the system. This approach gives more appropriate results for high resolution images and for the images where the targets are not overlapping to each other.
5.1. Input and Output Images:

(a) Input image 1

(b) Output image 1

Figure 5.1
Multiresolution framework with Neural Network approach for Automatic Target Recognition

Figure 5.2

(a) Input image 2

(b) Output image 2

Figure 5.2
Multiresolution framework with Neural Network approach for Automatic Target Recognition

(a) Input image 3

(b) Output image 3

Figure 5.3
This graph (figure 5.4) shows the comparison between response time of proposed approach and the fix size window-based approach for square images. Green line represents the response time of proposed approach while blue line represents the response time of later one. It demonstrates that our method takes very less response time as compare to fixed window based method.

In our approach only meaningful possible AOIs are selected and then only for these selected areas we calculate the Zernike moments, which lastly used as input for classifier. So the whole process takes very less time to response. But in window based approach this response time is comparatively large as if we have an input image of size 256*256 pixels; this gives a total number of \((256\text{-size of window})^2\) AOIs because AOIs are constructed by incremental traversing through the image, pixel by pixel. All the empirical values in graph are the results of simulation of both the approaches that are performed on Desktop system with 3.0 GHz processor and 512 MB RAM hardware specification.
5.2. Overall Recognition:

Although the overall recognition result is not 100%, Table 5.1 shows the overall recognition result. This table is created on the basis of the result in three images. In the table column named as “Successfully classified” shows the number of AOI which are correctly classified, column “False positive” shows the number of AOI which are incorrectly classified and “Missed AOI” shows the number of areas which should be selected as an AOI but not selected in this approach. In some cases after applying edge detection algorithm, we do not get all the edges of any object due to the similarity in reflectance from foreground and background. Similarly in the cases where objects having same texture feature values are overlapped or connected then selection of AOIs is not accurate. Due to both of above conditions this approach produces erroneous results. As shown in figure 5.6 object is not recognized because AOI is not created around that particular object due to faulty edge detection result. In this image yellow box shows the area around which AOI is not created.

<table>
<thead>
<tr>
<th>Image</th>
<th>Possible AOI</th>
<th>Successfully classified</th>
<th>False positive</th>
<th>Missed AOI</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9</td>
<td>9</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>6</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>27</td>
<td>24</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 5.1

Figure 5.5
6.1. Conclusion:

We have proposed a novel approach for solving the automatic target recognition problem for high resolution satellite images. This approach commits scale, rotation and translation invariant recognition and it is able to recognize the multiple targets in any scene. In this thesis we have solved one of the main problems of automatic target recognition that is the selection of appropriate AOI. Where other approaches take most of the time to process the areas where objects are not present, here we are predicting the possible AOIs and only these areas are checked for target or non-target. Sizes of AOIs vary according to the size of objects, around which they are present. This property makes the system capable to detect the objects of different size.

We are calculating the Zernike moments of scale and translation normalized AOIs, which are used as feature vector for neural network, so it always adheres to invariant properties of these Zernike moments. This approach takes very less time as compare to fixed window-size approach to process any image. It gives good recognition result for the images of different resolution and as the resolution of image increases, rate of recognition also increases.

6.2. Future Scope:

In future, the proposed recognition system may be further extended for different type of targets. We can make a system which will work on color images because the texture features are more accurate for such images.

We can use the Fourier descriptors [20, 21] in place of Zernike moment, which will add color invariant recognition property to present system because Fourier descriptors
work on basic shape of object. As Fourier descriptors calculation requires complete closed curve edges of objects, it is not applicable in my system.
Appendix ‘A’

[1] **LANS Pattern Recognition Toolbox:**

(MATLAB 5.0 or higher required)

(C) 2000.08.13 Kui-yu Chang

http://lans.ece.utexas.edu/~kuiyu

This program is free software; you can redistribute it and/or modify it. It provides number of image processing functions.

- **lans-patrec** - Pattern recognition routines
  - **knn1** - Find nearest k neighbours in 1 dim, sorted indices
  - **lans_centroid** - Compute centroid of a 2-D image
  - **lans_gmoment** - Compute geometric moments of a 2-D image
  - **lans_invariant** - Center and scale image to fixed mass
  - **lans_imgscale** - Scale and center a binary image w.r.t. bounding shape
  - **lans_zi2nm** - Convert index to order n and repetition m
  - **lans_zmoment** - Compute Zernike moments of a 2-D image
  - **lans_zmrecon** - Reconstruct image from Zernike Moments
  - **lans_znm2i** - Compute cumulative count of order n and repetition m
  - **lans_zpoly** - Compute Zernike polynomial over unit circle
  - **lans_zpossible** - Compute repetitions for Zernike moments of order n
  - **lans_imgcrop** - Crop/Fill image
  - **lans_classifier** - General LANS classifier framework
  - **lans_cidx2class** - Recover class name from unique integer label
  - **lans_class2cidx** - Assign unique integer label to class (non-numeric)
  - **lans_confuse** - Compute confusion matrix from output/desired classes
[2] EROS satellite:

EROS satellites are remote-sensing satellites including the EROS B satellite with 70 cm resolution since April, 2006 and, EROS A, with a 1.9 meter resolution since December, 2000. These satellites are giving “Panchromatic” images.
References


[23] LANS Pattern Recognition Toolbox.

    URL: http://davis.wpi.edu/~matt/courses/wavelets.

    URL: http://www.globalsecurity.org/military/facility/diego-garcia-imagery-2.htm