# Target Recognition Using Neural Networks for Model Deformation Measurements

Richard W. Ross
NASA Langley Research Center
r.w.ross@larc.nasa.gov


Dr. David L. Hibler
Christopher Newport University
dhibler@pcs.cnu.edu

## Abstract

*Optical measurements provide a non-invasive method for measuring deformation of wind tunnel models. Model deformation systems use targets mounted or painted on the surface of the model to identify known positions, and photogrammetric methods are used to calculate 3-D positions of the targets on the model from digital 2-D images. Under ideal conditions, the reflective targets are placed against a dark background and provide high-contrast images, aiding in target recognition. However, glints of light reflecting from the model surface, or reduced contrast caused by light source or model smoothness constraints, can compromise accurate target determination using current algorithmic methods. This paper describes a technique using a neural network and image processing technologies which increases the reliability of target recognition systems. Unlike algorithmic methods, the neural network can be trained to identify the characteristic patterns that distinguish targets from other objects of similar size and appearance and can adapt to changes in lighting and environmental conditions.*

## Introduction

At the NASA Langley Research Center, airfoil models are placed in environmentally controlled wind tunnels for aerodynamic testing and are subjected to pressure and temperature extremes. Aerodynamic loads are applied to the test specimen, and forces, moments, and other model parameters are measured directly. However, some properties, such as model twisting and bending, must be measured indirectly to avoid interactions with the test article. Optical measurement systems provide a non-intrusive solution for these applications [1].

Model deformation data can be acquired using a photogrammetric determination of 2-D coordinates of reflective targets from digital images. A frame grabber reads the digital image data from a camera and stores the digitized image in the frame grabber memory. The software acquires the image data from the frame grabber and a target recognition system analyzes the image to determine the location of the targets on the two-dimensional image. From these 2-D coordinates, the system transforms the coordinates into a 3-D coordinate space. These coordinates are compared to the known positions previously measured under steady-state conditions to determine the amount of twisting and bending under load [2, 3].

Traditional target recognition systems use algorithmic methods for identifying the position of the targets. Systems at NASA Langley Research Center compare the intensity of each pixel to a threshold; values below the threshold are eliminated by setting the intensity to zero. Groups of adjacent pixels that exceed the threshold are identified, and a region enclosing the adjacent pixels is defined. The size of this region is computed, and if the calculated size matches the expected size, the software concludes that a target has been found. When the software has located the expected number of targets, no further searching is performed; if it finds fewer, the search terminates unsuccessfully.

The algorithmic method described above has problems in many situations. Normal changes in test conditions can make accurate identification of targets difficult. Variations in model attitude alter target reflectivity levels, requiring the intensity thresholds to be constantly adjusted. Worse, using a threshold cutoff value can cause strong glints of light to be retained, while dim reflections from targets may be discarded. Physical constraints may often inhibit placing the light source near the camera, which reduces the amount of light reflected back at the camera and reduces

the contrast of the image. To compensate for poor lighting conditions, the test cell walls are painted black to increase the contrast. However, the paint may peel during a test run due to harsh environmental conditions.

On the other hand, targets have a distinct round shape and a characteristic light intensity pattern, allowing them to be distinguished from glints of light by a neural net. Unlike algorithmic methods, neural networks can identify patterns and adapt to changing conditions without reprogramming, resulting in a more flexible and robust method for target recognition.

This paper describes the development and evaluation of a neural net based target recognition system. The next section describes the methodology used, including the segmentation method and neural net design and training. This is followed by an implementation section that gives a brief description of the actual software used. In the final section the neural net target recognition system is evaluated and future directions are considered.

# Methodology

The target recognition system acquires image data from a frame grabber and analyzes the image to determine the location of targets on the two-dimensional image. This involves six stages.

1. Preprocessing
2. Segmentation
3. Normalization
4. Feature extraction
5. Classification
6. Postprocessing

The Preprocessing stage consists of two optional transformations that can be applied to make the targets easier to recognize. One transformation consists of a uniform contrast enhancement technique. The other involves wavelet filtering. For most images they are not required and will not be described further.

The Segmentation stage splits the image into smaller overlapping segments to simplify the processing by the neural network. The Normalization stage prepares the input data for the neural network by scaling the range of pixel intensities from 0 through 255 to 0 through 1. The Feature Extraction stage uses the trained neural network to determine which segments might contain a target. The Classification stage examines the results from the previous stage to determine which segments contain targets.

The final stage, Post-Processing, determines the coordinates of each target by computing the centroid of each segment using the weighted average of the intensity of each pixel. The x-coordinate of the target center is computed using each pixel's x-coordinate as a weighting factor; the y-coordinate is calculated in a similar manner.

The two most important areas for the design of the system involve segmentation and the neural net.

## Segmentation

Each image is segmented into a series of rectangular regions. To simplify the identification of targets, it would be desirable to impose constraints on the size of the segment and the degree of overlap between segments. Since the neural net will indicate which segment contains a target we require:

1. A segment contains either no targets or a single target.
2. Each target has one or more segments that contain it.

The first condition limits the size of the segment to be no greater than the size of two of the smallest targets, $T_{min}$, separated by the minimum separation distance, $D_{min}$, plus a one-pixel border. The second condition dictates the segment size must accommodate the largest target size, $T_{max}$, plus a one-pixel border on all sides. The second condition also places a lower bound on the overlap between adjacent segments as described previously. Thus, the segment size, $S$, and overlap, $O$, should be selected such that:

$$T_{\max} \leq S \leq 2T_{\min} + D_{\min} + 1$$

$$O \geq T_{\min} + 1$$

Based on the sizes of the targets on the images used, a segment size of 23 pixels per side, with an overlap of 16 pixels per side, was selected. These choices were selected so that each target would be contained in at least one region and that no region would contain more than one target. Based on these parameters, a single image with 752 x 240 pixels can be divided into 3360 segments.

## Neural Net Design and Training

The selected design approach for the neural network topology used a feed-forward network with a single hidden layer. The training method selected was backpropagation using a variable learning rate. These choices were selected because of their simplicity, robustness, and minimal memory requirements. (Second derivative methods turned out to require too much memory.) An adaptive learning rate approach was selected to compensate for the slow training times associated with backpropagation networks.

The adaptive learning rate approach was used by employing a small learning rate initially, then dynamically increasing the learning rate each training epoch as long as the error rate did not increase significantly. This approach achieved high accuracy due to the low learning rate, but avoided lengthy training times by increasing the rate where appropriate. If the error rate increased significantly, the newly calculated weights were discarded, the learning rate was rapidly decreased, and the training process continued at a more conservative pace.

The number of input neurons was determined by the choice of segmentation parameters, as described previously. However, the training time required depends on the number of neurons, and compressing the image size can dramatically reduce the training time. The image was compressed by averaging two pixels in each dimension, which reduced four pixels to one with relatively little loss of information. Thus the actual input size was $11^2$, or 121, instead of $23^2$.

The determination of the optimal number of neurons in the hidden layer was achieved by starting at a very low value and increasing until the network could be trained to an acceptable error level. The optimal number was 25.

For training purposes twelve sample images of a model with painted targets were used. These images were acquired under a variety of lighting and environmental conditions. Some of the images represented ideal conditions: bright reflections from the targets against a dark background. Other images taken at oblique angles produced darker reflections from the targets. Some images had a bright, uneven background caused by paint peeling from the chamber walls, and a few images were obtained when fog was present in the test section.

One problem encountered during the training of the neural network was premature saturation. The network would often reach a level of saturation early in the training cycle, requiring the training to be frequently restarted using a new set of random weights. This problem was overcome by limiting the initial range of values for the random weights and altering the training process. With this process, the network was initially trained on a set of patterns primarily containing targets. Patterns that did not contain targets were gradually added to the training set until the network was successfully trained.

## Implementation

The target recognition system uses a neural network for pattern recognition by analyzing images acquired by cameras from an external video measurement system. The system provides the capability to preprocess the images, train and test the neural network, and to identify targets from the images. The system was implemented using MATLAB® Version 5.1 by The MathWorks, Inc. MATLAB is a mathematical environment with support for matrix operations, file input and output, and image processing. It features a powerful programming language with built-in graphical user interface (GUI) and plotting capabilities. MATLAB has a large user base and has many add-on "toolboxes" which may be purchased from the vendor. Similar freeware packages are available on the Internet.

The target recognition system uses the Neural Network Toolbox from The MathWorks, Inc. for implementation of the neural network, and WaveLab, a freeware package for wavelet analysis for image filtering. WaveLab was developed through a collaboration between Stanford University and NASA Ames Research Center. Source code for WaveLab is available for UNIX, Windows, and Macintosh systems via the Internet at http://playfair.stanford.edu/~wavelab.

The software consists of five major functional components.
1. Manual Target Identification
2. Training Set Definition
3. Neural Network Training
4. Neural Network Testing
5. Target Recognition

The target recognition system provides a graphical user interface to allow the user to manage data files, specify operating parameters, apply image processing transformations, manually identify targets, divide the image into segments, and operate the neural network. These operations are performed through a menu structure.

## Results and Conclusions

The neural network proved to be a valuable tool for target recognition. Unlike algorithmic methods, the neural network can be trained to perform under different conditions without reprogramming or adjusting tuning parameters and thresholds. The neural network was successful in distinguishing targets from glints of light and background noise because of its ability to recognize the distinctive shape and intensity distribution of targets, compared to the less symmetric patterns of other objects.

The neural network successfully classified nearly all of the segments in the image as containing or not containing a target. Table 1 shows the data obtained by this implementation using six test images. The number of segments correctly classified and the number of targets

accurately identified are given for each image. Based on the average number of targets correctly found, the overall percent of segments accurately identified is 99.7%. The overall percent of targets properly identified is 92%.

| Img # | # Seg Right | # Seg Missed | # Seg Xtra | # Tgt Right | # Tgt Missed | # Tgt Xtra |
|---|---|---|---|---|---|---|
| 1 | 3348 | 5 | 7 | 8 | 0 | 0 |
| 2 | 3340 | 19 | 1 | 7 | 0 | 1 |
| 3 | 3346 | 10 | 4 | 8 | 0 | 0 |
| 4 | 3360 | 0 | 0 | 8 | 0 | 0 |
| 5 | 3341 | 13 | 6 | 7 | 1 | 1 |
| 6 | 3350 | 6 | 4 | 6 | 1 | 1 |

Table 1. Neural Network Training Results.

The reason the percent of targets identified is lower than the overall percent of segments correctly identified is due to the mixture of training data provided to the neural network. Since the network was trained on complete images, most of the training patterns did not contain targets. Out of the 3360 training patterns obtained per image, only an average of 29 of these segments contained targets, or 0.86%. More realistic results would be expected if the proportion of patterns containing targets to those not containing targets was more balanced.

Overall, the neural network approach, combined with preprocessing techniques where appropriate, was successful at identifying targets. Of the 13,440 training patterns, the neural network classified which of these patterns contained targets with 99.7% accuracy. Unlike algorithmic methods, which simply identify objects of particular sizes and intensities, the neural network was able to identify the distinctive characteristics of the targets and to successfully discriminate between targets and other objects.

An alternative training method using auxiliary information was also explored to determine if this would decrease training time. The *x* and *y* coordinates of each segment were included with the intensity data. This approach did result in faster training times, but adversely affected the accuracy of the network. By adding the coordinates, the number of "false positives" was greatly increased. The segments that were falsely identified as containing targets were most often located in straight lines radiating from actual targets. There was a distinct pattern to these regions: they followed a path parallel to the edge of the model wing. It appeared that the network learned to expect where the targets should be, based on the coordinates, instead of identifying where targets are actually located. The problem was easily corrected by only training the network on the image data without coordinates.

## Future Directions

Neural networks, combined with the appropriate use of preprocessing methods, have proved to be effective at identifying targets in digital images. However, additional research may be required to implement this technique in a production environment.

For full-scale production use of this system, additional images would need to be acquired to provide more thorough training under a greater variety of conditions. In particular, more training patterns containing targets are required. Further research into other preprocessing and encoding techniques may prove beneficial in improving accuracy or reducing training time.

Identification of the coordinates of the targets relies on the centroid computations, which determine the center of an object based on the center of mass of the image segment. The "mass" of each pixel is the intensity level. This method works well for identifying the center of a target against a uniform, dark background, but is affected by the presence of background noise and glints of light. Further research areas may include the development of better techniques for computing the center of a target.

## References

[1] A. W. Burner, "Model Deformation Measurements at NASA Langley Research Center," in proceedings of *AGARD 81st Fluid Dynamics Panel Symposium on Advanced Aerodynamic Measurement Technology* (Seattle, WA, 1997).

[2] A. W. Burner, R. H. Radeztsky, and Tianshu Liu, "Videometric Applications in Wind Tunnels," in Videometrics V, proceedings of the International Society for Optical Engineering, SPIE 3174 (San Diego, 1997), p. 234-247.

[3] P. R. Wolf, "Elements of Photogrammetry" (McGraw-Hill, Inc., NewYork, 1983), pp. 576-613.