Using Artificial Neural Networks and Image Segmentation to Assist in Mobile Robot Navigation

Mark Stemm
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh PA 15217
email: msaj@andrew.cmu.edu

May 6, 1994

Abstract

An mobile autonomous robot must have a reliable system of navigation that avoids large objects in its path such as recycling bins, etc., but can recognize "important" objects such as doorways. We have developed a system that can recognize such objects as doorways and recycling bins and that can predict the distance to these objects. The system is implemented by using artificial neural networks to classify objects and predict distances, supplemented by image segmentation of the camera image from the mobile robot into color regions.

1. Introduction

Xavier(Figure 1) is an autonomous mobile robot developed by the learning robots group at Carnegie Mellon University [O'S94]. It has a variety of on-board sensors, including sonar, laser, dead reckoning of position, and an on-board camera. One of Xavier's important tasks is to move from one room to another in the corridors of Wean Hall while avoiding obstacles in its path. This task can be divided into several parts:

- Separating doorways from the background.
- Separating obstacles from the background.
- Locating doorways and obstacles relative to Xavier's position.

One approach to this problem is to consider Xavier's sensor vector as a pattern that must be classified as a "door pattern" versus "non-door pattern", or as an "obstacle pattern" versus "non-obstacle pattern". (In

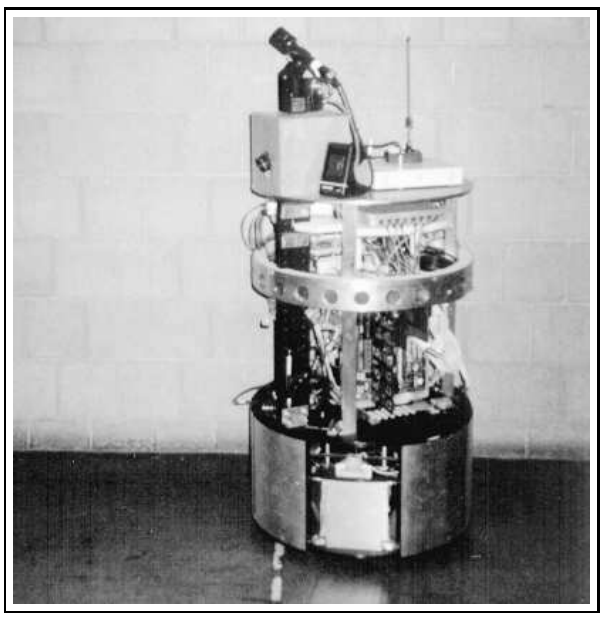

Figure 1: Xavier.

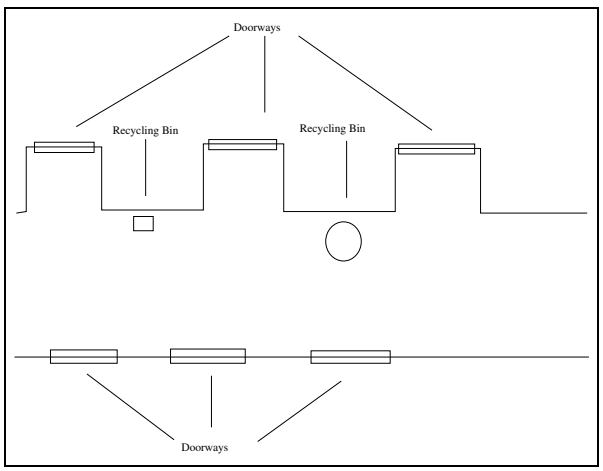

Figure 2: The learning environment.

this paper, we will look at recycling bins as examples of obstacles to be avoided.) Artificial Neural Networks have shown good performance as pattern classifiers in similar domains with high amounts of noise and input variability. Pomerlau [Pom90] has used neural networks to create a road following system based only on image data.

Training artificial neural networks that could classify the sensor state as one of these types is one solution to this problem. In addition, neural networks can be used to provide distance information to both doorways and obstacles, giving Xavier information on how to approach or avoid these features. This paper describes the methods used to train these neural networks, their performance, and areas for future work. In addition, one method used to decrease the training time of the networks and increase the performance is discussed; segmentation of the camera image into color regions close to the desired color of the object.

2. Problem Domain

The environment that we used for training and testing of the system was the corridor outside the learning robots laboratory (Figure 2). Several features about the corridor are of interest, and have an impact on the approach taken:

- The doorways on one side of the corridor are recessed into the wall, while they are not in the other side. This will tend to make sonar and other depth finding information more important than image information. In both cases, the doorways are a different color than the surrounding wall; image information will be useful in any case.

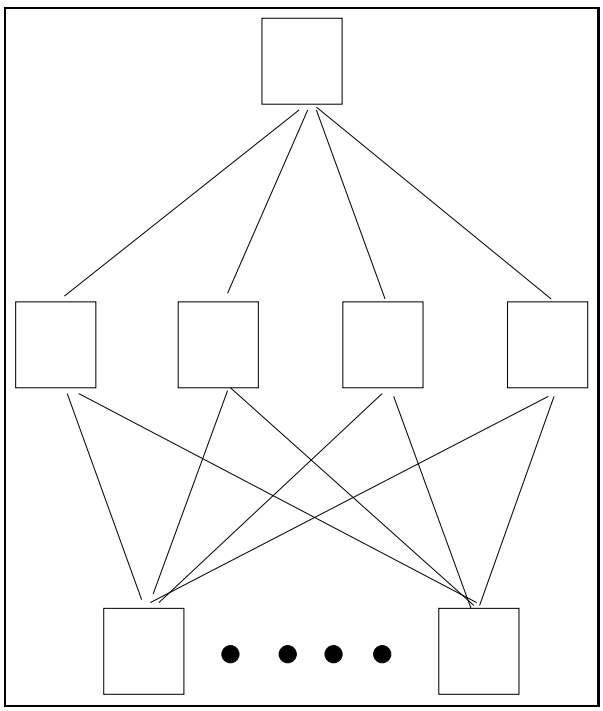

Figure 4: Network architecture.

- There are obstacles in the corridor that Xavier must avoid; notably the large and small blue recycling bins. Distance information will also be important for these recycling bins, because they are not flush with the corridor wall.

3. Data Collection

We developed two programs(CTRECORD and CTREPLAY) [O'S94] that are used to collect sensor data from Xavier as it is operated in the environment, and to replay this data at a later date. CTRECORD records the current sensor snapshot (including the current sonar, laser, and robot position, as well as the current image from the camera (at 1/4 resolution) and saves these sensor frames for later examination and network training. A classification is given to each snapshot and recorded with the sensor information. CTREPLAY is used to examine this data and possibly change snapshot classifications "off-line" at a later date. Figure 3 shows an example session with CTREPLAY.

4. Network Architecture

The basic network architecture that we used was a single hidden layer feed-forward neural network.(Figure

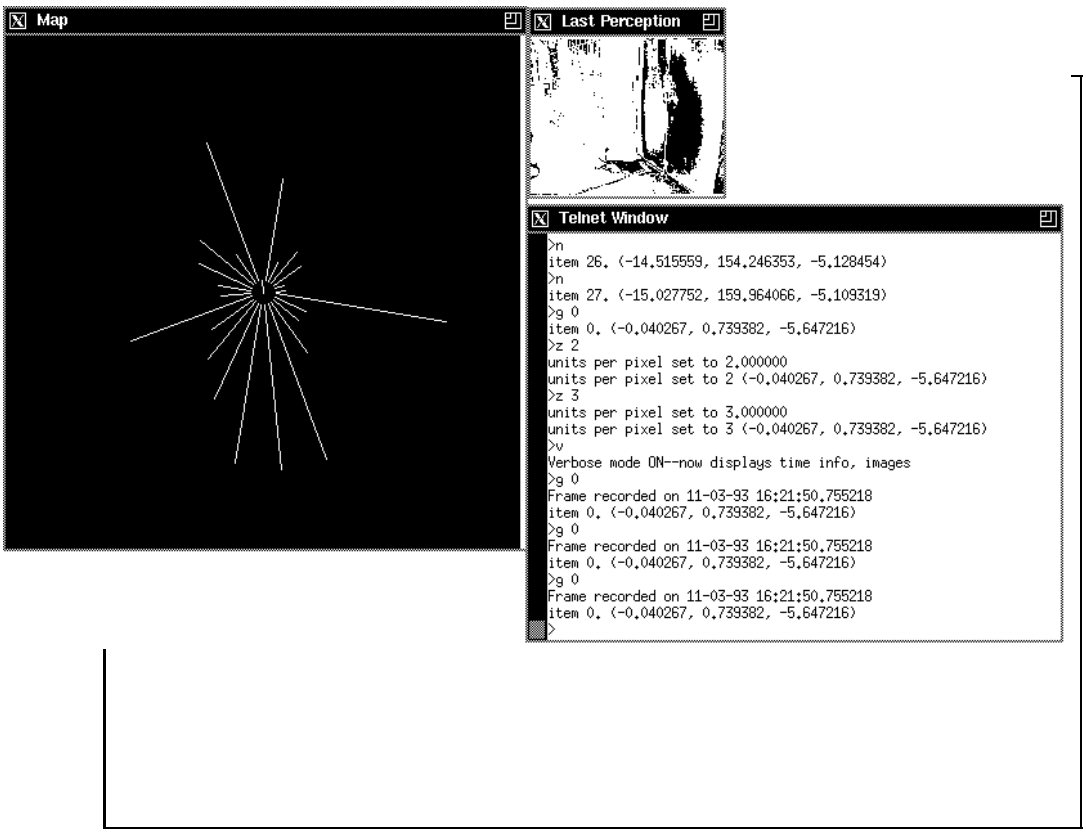

Figure 3: Example data replay session.

4). The input layer consisted of sensor information from Xavier. Each of the input units is fully connected to a hidden layer of 5 units, which is in turn connected to an output layer of a single unit. There were two types of networks trained; classifiers and distance predictors.

4.1. Classifier

One classifier network is trained for each type of object (doorway or recycling bin). The classifier networks consisted of 54 input units. 24 inputs were sonar readings from the sonar ring around Xavier, and 30 inputs were the red, green, and blue values of 10 sub-sampled pixels taken from a single stripe from the center of the image. The output unit produces a value between 0 and 1; 1 if it found the object(door or recycling bin), and 0 if it did not.

4.2. Distance Predictor

Like the classifier networks, one distance predictor network is trained for each type of object. The distance predictor's input layer is identical to the input layer of the classifier. The only difference is in the output layer, where the output unit produces the distance to the next door or recycling bin in centimeters(divided by a constant to assure that it is between 0 and 1).

4.3. Training

We trained both the classifier and distance predictor networks using the Quickprop algorithm [Fah88] with cross-validation. In this method, we partition the data randomly into a training set and a test set. The training set will be used to train the network, and the test set will be withheld for performance checking. The weights between the input layer and the hidden layer and the hidden layer and the output layer are set to random values. The network is presented with each (input, output) pair of the training set. The activation from the input is propagated through the network to the output layer. The network's response is compared to the desired output. If the network's response does not match the actual response, then the weights of the interconnections between units are changed slightly to produce an output closer to the desired output. This process is repeated for every member of the training set, and is

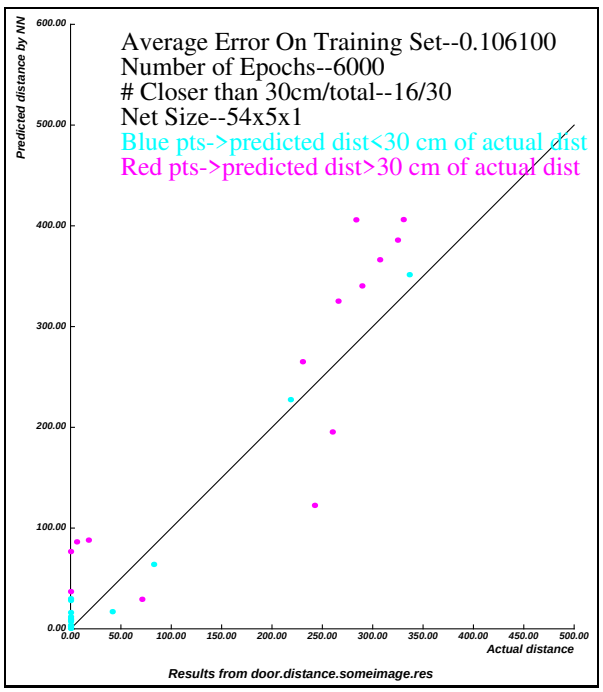

Figure 7: Results from the distance predictor for doors.

called a single *epoch*. Periodically, the performance of the network is tested using the test set. This process is repeated until the average error on the members of the test starts to increase(as the network begins to "overtrain" on the training set), or until the error on the test set falls to an acceptable level.

5. Preliminary results

5.1. Classifier

The classifier networks easily classified the sensor state into recycling bin vs non-recycling bin frames. See figure 6. [1] For the recycling bin classifier, 125 training examples and 45 test examples were used. The bin classifier classified approximately 90% of the test set frames correctly with only 1500 epochs of training. For the door classifier, 125 training examples and 73 test examples were used. The door classifier's performance was not as good, however(Figure 5). 6000 epochs of training were necessary, and the average error on the training set was higher. In addition, the performance on the test set was not as accurate; more frames were misclassified, and the output of the network was "fuzzy" rather than very close to 0 or 1.

[1] For all of the classification results, points moving along the x axis represent the output of the network as it moved to successive sensor snapshots along the corridor.

5.2. Distance Predictor

The distance predictor produced acceptable, although not very impressive, results. See figure 7. For the doorway distance predictor, 87 training patterns and 30 test patterns were used. 6000 epochs of training were necessary, and the network was able to predict the distance within 30cm only for 50% of the test set frames.

6. What is the network using, anyway?

It is interesting to compare the results from the above networks with networks trained only using Xavier's sonar information. See figures 8, 9, and 10. In all cases, the network was able to produce similar results *without using the image information at all.* This surprising result has a simple explanation. In both cases, the distance information is prominent, and overwhelms the image information in importance in learning the outputs. This shows a drawback of the current system– the distance information will not always be prominent, because the doorways will not always be recessed into the wall as they are in the training examples. We decided to develop a method that would make the image information more useful to the network, and to increase the performance of the networks as a whole.

7. Using Image Segmentation

We decided on using segmentation of color regions of the image. This approach starts by looking for a desired color in the image, and then segmenting the image into regions that are close to the desired color. This works because both the doorways and recycling bins have large homogeneous color regions that indicate the position of those objects.

7.1. The image segmentation algorithm

- Determine what color the object you are looking for is as a (R,G,B) 3-vector. For now, this is done manually.
- For each pixel in the image:
  - Take the vector difference between the (R,G,B) vector for that pixel and the desired

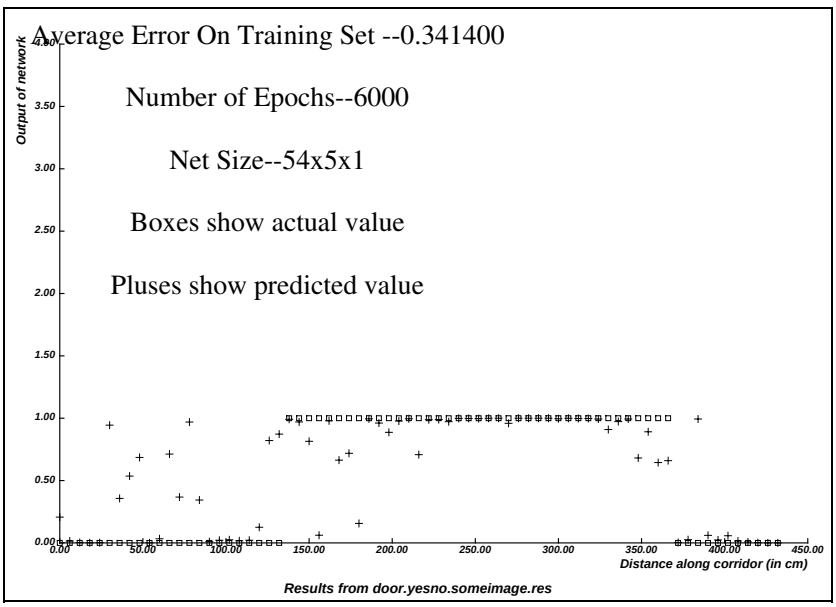

Figure 5: Results from the classifier for doors versus non-doors.

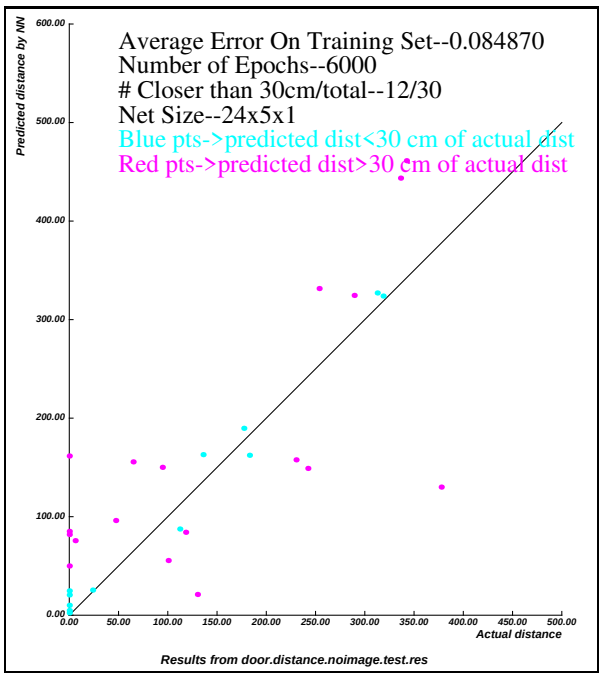

Figure 10: Distance Predictor results for doors using only the sonar information.

color. This difference vector is scaled by the sum of the lengths of the pixel and desired color vectors, as a method of normalization to account for changes in intensity. Unlike the thresholding method of [ea93], this method works well for arbitrary colors rather than only green ones.

– If the length of the difference vector is close to 0, write out white in an output "mask" image–otherwise, write black.

- the resulting "mask" image will show what pixels in the original image were close to the desired color.

- Segment the image into connected regions.

- Give the center(row, col) of this region as well as the maximum and minimum row and column of the largest region to the network along with the sonar, laser, etc information.

Figures 11 and 12 show the output of the algorithm. Figure 11 shows a sample image from the robot camera, with the outlines of the connected regions in the mask image in red. Figure 12 shows the mask image resulting from the pixel-by-pixel differencing.

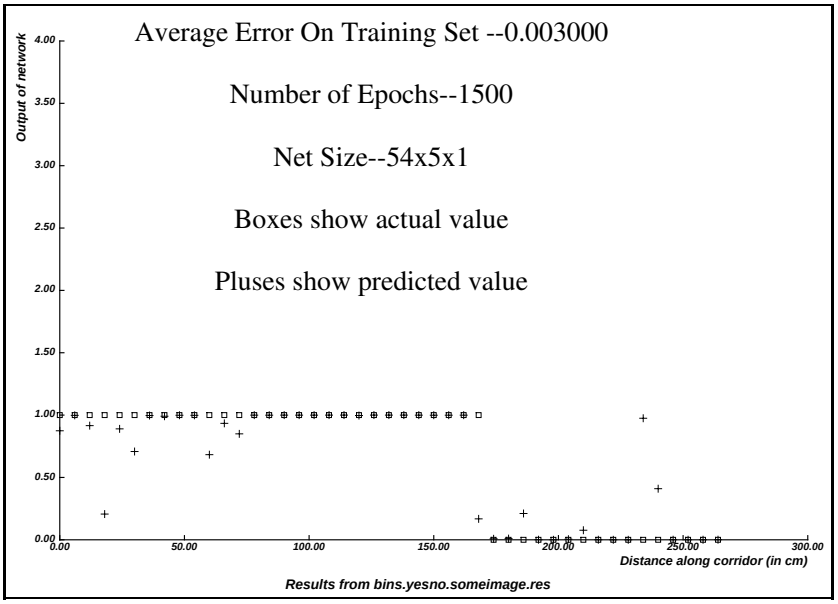

Figure 6: Results from the classifier for bins versus non-bins.

7.2. Advantages

There are several major advantages to using this approach rather than giving pixel values to the network directly:

- The network is smaller. Using this method, the image information is encoded in 6 values rather than 30(10 pixels * 3 RGB values). The resulting network will train faster than the older network.

- Before, the network only considered a single stripe of pixels from the image. If that stripe was placed incorrectly, the image information could be useless (for example, if the image stripe never included the recycling bin). Using this method, we are giving information from the whole image to the network; using the recycling bin example, no matter where the recycling bin is in the image, some information about it will get to the network.

- The inputs mean more; the network does not have to do any decoding of pixel values to geometry information–this geometry information is already encoded by the preprocessing step..

8. Results using image segmentation

The addition of the preprocessing step improved both the accuracy of the predictors and locators, as well as decreased the training time. See figures 13, 14, and 15. In all three cases, the average error on the training set decreased. The networks classified more of the frames correctly as bin/non-bin or door/non-door[2]. The Distance Predictor had significantly more predicted values within 30cm of the actual values(29/40 versus 12/30 on the test set). The number of training epochs also decreased on the door classifier and distance predictor for doors, by a factor of 1/6 and 1/3, respectively.

9. Conclusions/Future Work

The trained classifier and distance predictor networks have been shown to effectively classify doorways and obstacles(recycling bins), as well as make accurate predictions of the distances of these objects. They could be used as a lower-level part of a navigation system on a mobile robot, to help it maneuver to the goal state while avoiding obstacles. Areas for future work include:

[2] Notice that the predictor made most of its misclassifications at snapshots close to the yes-no and no-yes borders, which raises the question: who actually is doing a better job of classifying–the network or the trainer?

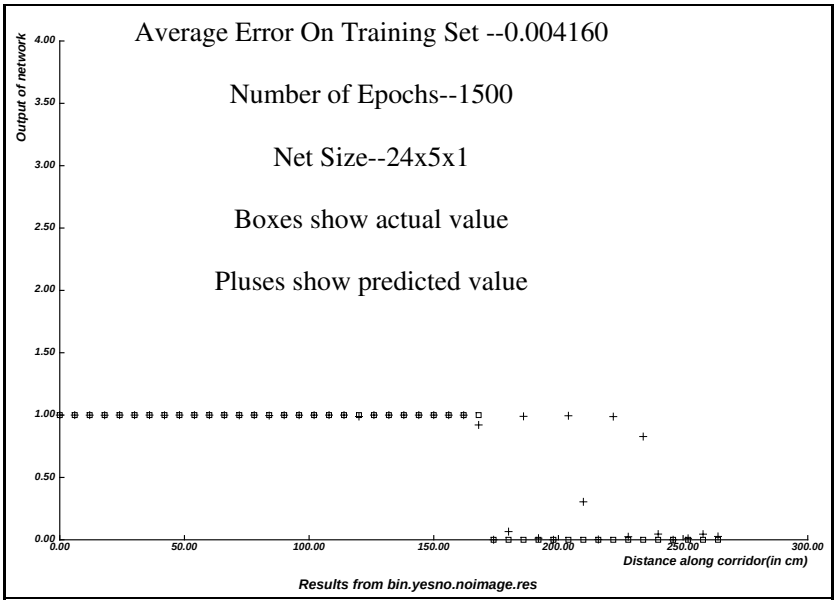

Figure 8: Classifier results for bins using only the sonar information.

Figure 11: Sample camera image and region outline.

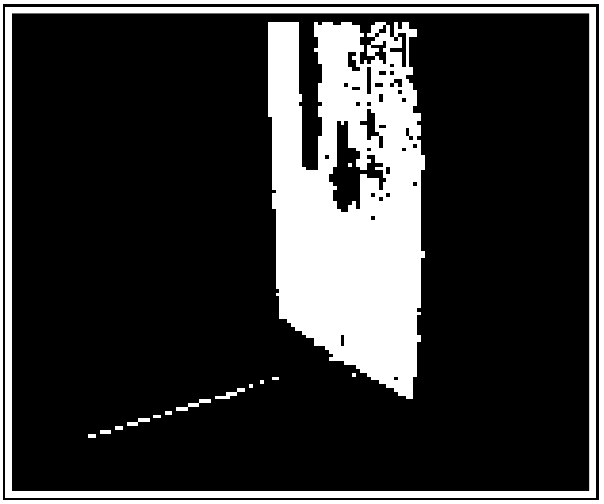

Figure 12: Mask image resulting from image preprocessing step.

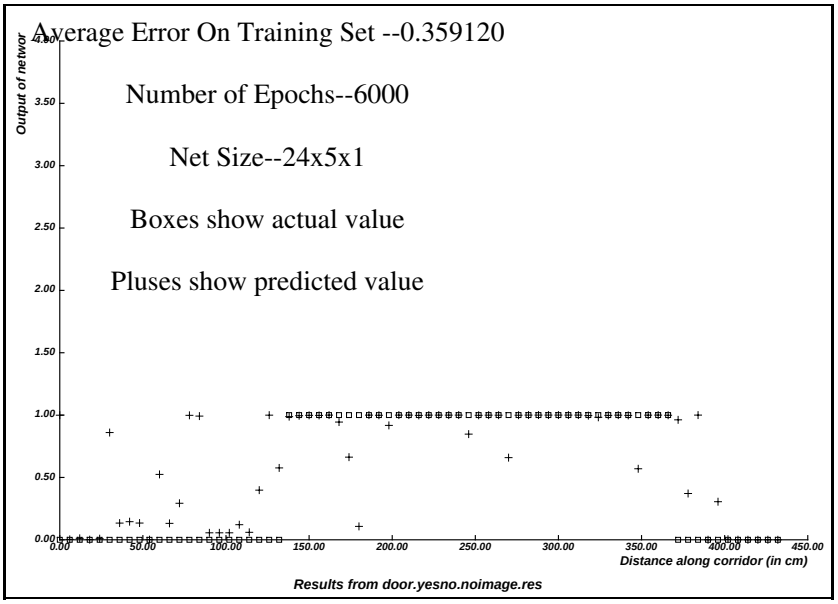

Figure 9: Classifier results for doors using only the sonar information.

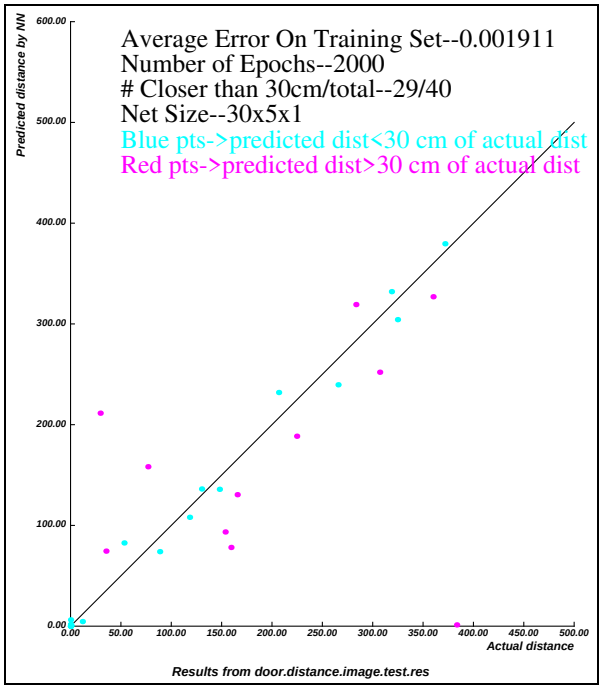

Figure 15: Distance Predictor results using image segmentation.

- Currently, we select the color for the image segmentation by hand. There is no guarantee that the color we are looking for will produce the best results. This method could be automated, by iterating through several network training/testing cycles, using a different color to segment the image each time. Eventually the color most useful in locating the objects will be chosen by the system.[3]

- Other computer vision methods like line detection could be used to pass more information to the network, like the actual outline of the door frame or obstacle.

10. Acknowledgements

Thanks go to Tom Mitchell for proof-reading this paper and for advising me on all of work done throughout the year. Thanks also to everyone in the Learning Robots Lab for the time on Xavier and for answering my "How does ⟨foo⟩ work on Xavier?" questions. Thanks also go to the Digital Mapping Laboratory for the image segmentation software, especially to Stephen Ford for the information on color image segmentation and for sanity-checking my ideas on the pixel differencing.

[3] It may be possible that the grey of the walls or the black of the door outlines will produce better classifiers than the brown of the doors.

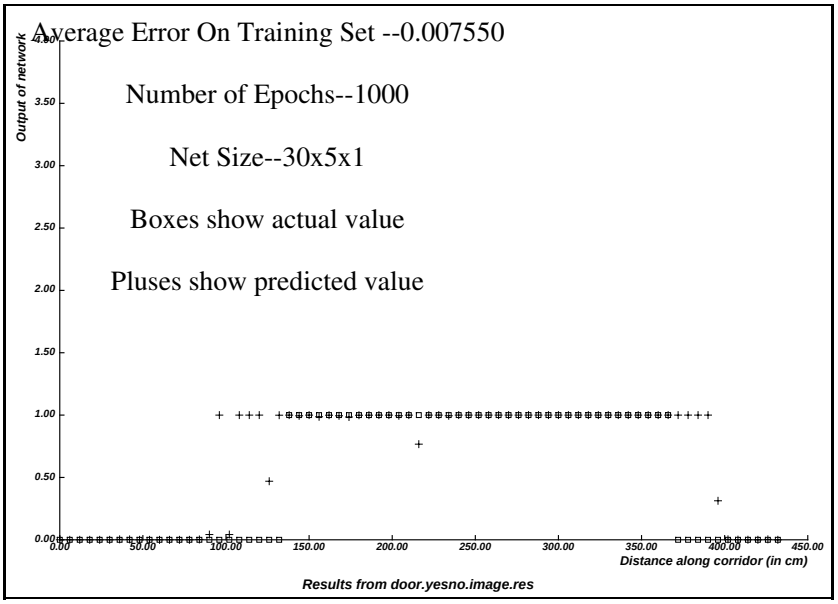

Figure 13: Classifier results for doors using the image segmentation.

References

[ea93] Clint Bidlack et al. Visual robot navigation using flat earth obsticle projection. Technical report, University of Michigan, 1993.

[Fah88] Scott Fahlman. Faster-learning variations on back-propigation: An empirical study. In T.J. Sejnowski G.E. Hinton and D.S. Touretzky, editors, *1988 Connectionist Models Summer School*. Morgan Kaufman, 1988.

[O'S94] Joseph O'Sullivan. Xavier Manual. Learning Robot Lab Internal Document - contact josullvn@cs.cmu.e du, January 1994.

[Pom90] Dean Pomerlau. *Vision and Navigation: The Carngeie Mellon Navlab*, chapter 5. Kluwer Academic Publishers, 1990.

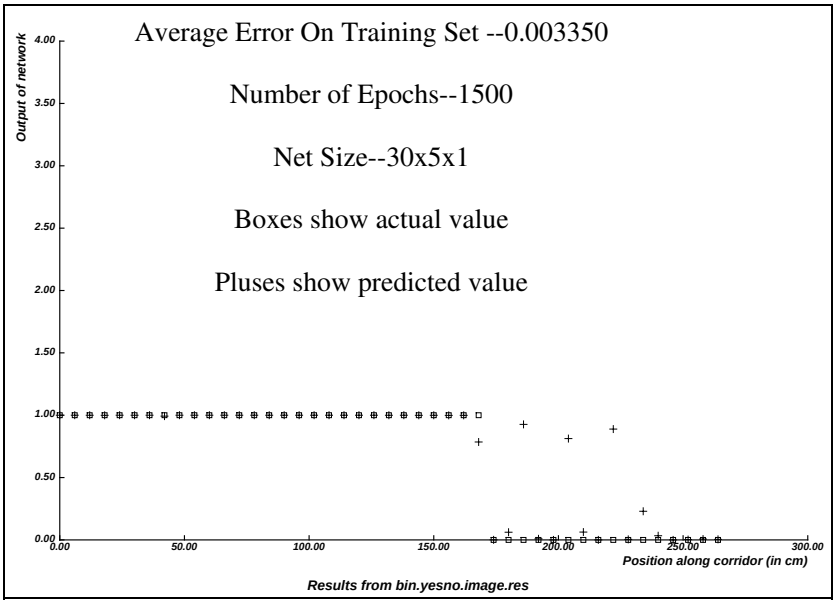

Figure 14: Classifier results for bins using the image segmentation.