EXPLORING IMAGE PROCESSING CAPABILITIES OF AIBO

by

SRIGOPIKA RADHAKRISHNAN

(Under the Direction of Walter D. Potter)

ABSTRACT

Human robot interaction is an important aspect of developing robots. The level to which a robot emulates human actions determines the success in the development of the robot. The trend of robots interacting with humans using 'emotions' started commercially with Sony's entertainment robots. This trend continued with Honda's Asimo and Sony's Qrio which are not yet available in the commercial market. In research, universities have long been creating robots that interact with humans as normally as humans do. Sony's entertainment robot is called AIBO which is an acronym for Artificially Intelligent RoBOt. AIBO in Japanese means 'Companion'. The selling point of AIBO was its ability to act (and look) like a puppy or a dog. Asimo and Qrio are humanoids which act and look like humans (remember C3PO from Star wars!). This thesis explores the possibilities of making AIBO more than just an entertainment robot by teaching numbers and operators to solve mathematical expressions. AIBO is also taught to recognize and respond to gestures. Neural networks are used as the learning algorithm to teach AIBO the numbers, operators and gestures. AIBO looks at an expression, calculates the result and provides the result the onlooker. It also recognizes gestures and performs actions for them.

INDEX WORDS: AIBO, Image Processing, Neural Networks, Offline Learning, Online Testing.

EXPLORING IMAGE PROCESSING CAPABILITIES OF AIBO

by

SRIGOPIKA RADHAKRISHNAN

B.E (Electronics and Communications) P.S.G College of Technology, Coimbatore, India, 2002

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2005

EXPLORING IMAGE PROCESSING CAPABILITIES OF AIBO

by

SRIGOPIKA RADHAKRISHNAN

Major Professor:	Walter D. Potter
Committee:	Khaled Rasheed Suchendra Bhandarkar

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
December 2005

DEDICATION

To my parents – Thank you for everything.

ACKNOWLEDGEMENTS

I would like to extend my heartfelt thanks to Dr. Walter D. Potter for his guidance and patience in my thesis. He has been a great inspiration to me both personally and professionally. I am grateful for all his help when I had to extend my course of study. I am also grateful for all the automotive knowledge that he has shared with me. I would also like to thank my committee members Dr. Khaled Rasheed and Dr. Suchendra Bhandarkar for their support through the course of my study.

I wish to thank my parents for believing in my decisions and supporting me entirely in all my endeavors. They have sacrificed so much to make me a better person. I wish to thank my sister for being the strong independent woman I respect.

I would like to thank Jegan for his encouragement and love all these years. I would also like to thank Allundu, Britto, Bunty, Divya, Indira, Jones, Kanna, Priyanka, Rama, Sarath, Selvi, Srt, and Tutu for being the incredible friends that they are.

TABLE OF CONTENTS

Page

CHAPTER

Chapter 1

Introduction

Robotics is the art of creating machines that can be programmed to imitate the actions of an intelligent creature usually humans. Every aspect of our lives is being influenced by robots sometimes without our knowledge. We have robots assembling cars at the Ford plant, driving trains in Paris, defusing bombs in Northern Ireland. Think of a boring or a dangerous job and there is probably a robot doing that task.

In light of the active interest shown in Robots, Sony decided to make a robot that can entertain people. In Japan, 3000 units of AIBO were sold in the first 20 minutes of going on sale. AIBO is an acronym for Artificially Intelligent RoBOt and in Japanese means a ‘companion’. AIBO is an autonomous robot dog that can hear and see and interact with people using sounds and actions. AIBO can start off as a puppy and grow to be an adult, by learning to interact with its owner. They also possess a sense of balance and touch. AIBO’s phenomenal success as an entertainment dog led robotics researchers to explore its possibility of being a research robot. AIBO has many characteristics that make it an excellent research robot, as will be discussed in the following chapters. Sony developed a software development framework for AIBO to better its use as a research tool.

There are different features of AIBO that can be researched. They can be used for exploration, mapping and navigation purposes. This thesis involves researching the visual aspects of AIBO’s sensors. It has a camera mounted on its face that can be easily accessed, and that makes it a good platform to create and test recognition tasks. Visual recognition behaviors have been researched on AIBO extensively in tournaments like Robocup where two teams of AIBOs play soccer. Visual recognition in Robocup is mainly based on color information. This research concentrates on visual recognition based on feature extraction rather than on color alone.

Human Computer Intelligent Interaction is an important aspect in communication between humans and robots. The idea behind creating the behaviors discussed in this research is to create behaviors that can simulate intelligent interaction between humans and AIBO. Two recognition tasks are considered here. One of them involves learning numbers and operators in math to solve mathematical expressions. AIBO learns numbers 0-9 and operators +,-,*,/,(,),%,^ and they are combined together to solve math.

The other recognition task considered here is gesture recognition. Gestures are performed to AIBO and it correctly identifies the gestures and performs actions corresponding to the gestures.

Intelligent behavior in robots is in the eye of the beholder. Computers were initially created with the aim of solving math problems. But we do not call a math solving calculator intelligent. But a robot that solves math can be thought of as intelligent because it tries to simulate a human approach to obtaining the solution. This is done by first viewing the characters presented to it, recognizing them, and then solving the recognized numbers and operators by using a set of rules. This research tries to explore the possibilities of incorporating behaviors in robots to make them more human.

Chapter 2

Basic Math via Character Recognition on AIBO Robot Dog [1]

[1] S. Radhakrishnan and W. D. Potter, Basic Math via Character Recognition on AIBO Robot Dog, *Transactions on Information Science and Applications*, 1(3) pp. 127-133, 2005

Abstract

One of the objectives of AI involves creating applications or behaviors on machines that simulate intelligence. This paper presents an application of Neural Networks and Image Processing where AIBO solves mathematical expressions after learning to recognize numbers and operators via a combination of Offline Learning and Online Testing. This application results in a robot that exhibits intelligent behavior (as perceived by the onlooker) by being able to recognize and solve math problems as humans do and can be used as a platform for other recognition tasks.

Introduction

Neural Networks are well suited for non-linear problems since the network itself is inherently non-linear [7]. When we consider situations that involve classification and recognition based on visual input, Neural Networks are one of the many different methods that can be used, some others being Bayesian Learning, Support Vector Machines and K-Nearest Neighbor algorithms.

AIBO, the Sony Robot dog (although intended for entertainment purposes) is an excellent platform for robotics research because of its design structure. It provides a robust robotic platform to test algorithms and applications. The ERS220A model shown in Figure 1 (used in our experiments) comes equipped with a CCD camera, one infra-red sensor and various pressure sensitive sensors.

This paper describes the development of an application to teach AIBO to recognize numbers and characters and as an extension of the recognition, to be able to solve mathematical expressions. The idea is to have the robot look around a room for a mathematical expression (which is pink in color for easier identification), take a picture of it and process the image to give the result of the expression. If the robot recognizes a pink non-expression it identifies it as such and continues to look around for an expression. This application uses neural networks as the learning algorithm and image processing techniques to focus on necessary segments of the image. As a prelude to evaluating an expression the robot was trained to identify 13 different shapes.

The motivation behind creating this behavior on AIBO is to extend its role as an interactive entertainment robot and add aspects of intelligent behavior to it. The ability of robots to detect specific objects based on features rather than color alone is an important research area. This paper addresses this research interest by training AIBO to learn numbers and operators based on their features rather than color. This behavior can be presented to young children as an education enhancement tool to encourage them to learn math.

The rest of the paper is divided as follows: Section 2 explains the framework the software platform uses to create the behaviors on AIBO. Section 3 justifies the choice of the learning algorithm used. Section 4 describes the Image Processing tasks that were performed on the image taken by AIBO. Section 5 explains the offline learning and online testing performed using neural networks. Section 6 mentions the math solving algorithm designed for this behavior. Section 7 and Section 8 describe the results observed and the future work intended for AIBO using this application as a platform.

Background - AIBO and Tekkotsu

Sony AIBO robots were initially marketed as Entertainment Robots but the features that they provide have made them a successful robotic research tool. Table 1 gives the different features that are present in the ERS220A, the robot used in this project.

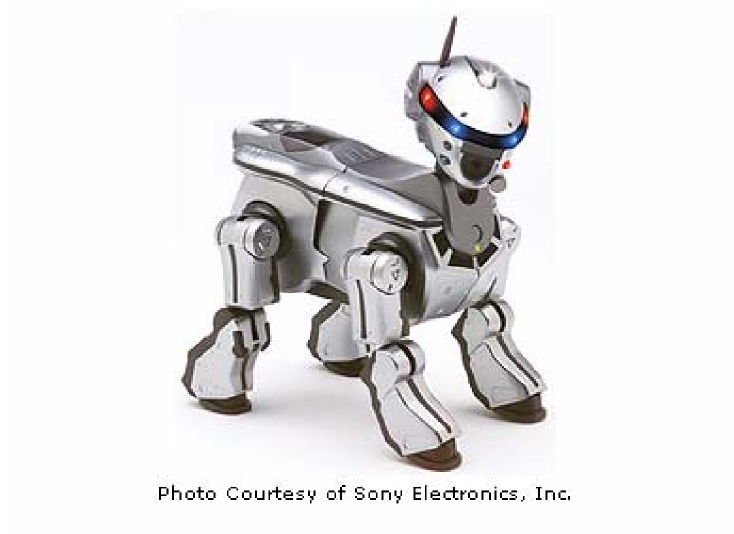


Photo Courtesy of Sony Electronics, Inc.

Figure 1: AIBO ERS220A

Due to an increasing interest in AIBO on the part of AI researchers, Sony started to actively promote a software development environment for the AIBO called OPEN-R. OPEN-R provides the user with modularized hardware and software while supporting wireless LAN and TCP/IP protocol [13]. Tekkotsu is an application development framework for robotic platforms developed by Carnegie Mellon University as part of a grant from Sony. The framework is designed to handle the routine tasks of OPEN-R so the user can focus on higher level programming using C++ [8]. Our application builds on the existing framework of Tekkotsu to develop a new behavior. Behaviors are defined as applications created by users that run on AIBO [8].

Hardware	**Details**
384 MHz MIPS processor	
32 MB RAM	
802.11b Wireless Ethernet LAN card	
Memory stick reader/ writer	
20 joints	18 PID (proportional integral derivative) joints with force sensing 2 Boolean joints
9 LEDs	
Video camera	Field of view: 47.8° high and 57.6° wide, resolutions: 208x160, 104x80, 52x40 up to 25 frames per second
Stereo microphones	
Infrared distance measure	Range: 100-900mm
X,Y,Z accelerometers	
8 buttons	2 pressure sensitive, 6 boolean
Sensor updates every 32 ms	4 samples per update

Table 1: Features of ERS-220A

The Tekkotsu framework provides access to the camera on AIBO. It also allows the programmer to control the motors and create behaviors using motions on AIBO. It is possible to control

AIBO wirelessly using a peer to peer network connection and the telnet console on a host computer or laptop. Tekkotsu has implemented a GUI called Tekkotsu Mon to access the applications that run on AIBO from a laptop. This facilitates an easier debugging process and better PC-Robot communication.

Design Choice

Neural Networks have commonly been employed in classification and pattern recognition problems because of their ability to generalize based on fewer training examples and the tolerance exhibited towards error [6]. The most important reason for selecting Neural Networks as the training algorithm for this application is the portability of these systems from training to testing. Since training and testing are to be conducted on different platforms, it is important to be able to transfer the trained results to the testing platform with ease. Since the trained results of a neural network are a set of weights, they can easily be transferred to AIBO.

AIBO Looks and Learns – Image Processing

Image Processing on AIBO has been extensively researched for the RoboCup Tournament where the robotic dogs play soccer. In this tournament AIBO uses color segmentation to identify and detect different objects in its environment [9]. Color Segmentation is one of the most commonly used methods of object detection and identification in robots. The Sony AIBO comes equipped with a color CCD camera and is tuned to detect pink objects, such as the pink ball that comes shipped with AIBO. Our application uses the pink tracking capability of AIBO in detecting the expression in the environment. The colored mathematical expression is placed against a black background for ease of segmentation. The maximum resolution of the CCD camera on AIBO is 352x288. This is the size of the image that our application deals with. AIBO saves an image after it is at the right distance from the expression and is properly aligned to it. The right distance from the expression is determined by the distance from which AIBO can see all the characters in it. This image is later accessed by the image processing algorithm to be classified. The saved image is in the RAW format which essentially is untouched by any compression algorithm. The RAW file format is used in the Tekkotsu platform as an alternative to the JPEG format.

The image processing can be classified into three parts. The first portion of image processing extracts the pixel values necessary for the following steps. The second portion recognizes the pink pixels and creates a bounding box around the character that has to be recognized. The third portion converts a bounding box of a random size to a constant 20x20.

The Image Processing algorithm runs individually on every character in the expression. The entire process is carried out for every character in the image separately. So a mathematical expression like 8 * (5 ^ 3) / 2 is processed one character at a time. Every character is extracted, a bounding box created and then fed to the neural network as a 20x20 input.

Extraction of Pixels

The first step in image processing is to extract the pixel information from the image so that it can be processed. The image saved to the Memory Stick® is in the RAW format which means the pixels are unchanged from the camera. The color format of the image is YUV where Y stands for brightness, U and V stand for chrominance. U relates to the blue-yellow color components while V relates to the red-green color components of the color image [4]. Since the expressions are all pink in color, it is enough to extract only the V component of the pixel. This method leads to fewer computations on the part of the image processing algorithm. Since the V component contains color information, it is to a large extent independent of varying illumination. This method cannot be recommended for other recognition tasks since a large amount of information is lost by ignoring Y and U components. The V component is extracted and sent to the step where a bounding box is created.

Creating a Bounding Box

The second step of the Image Processing algorithm creates a bounding box around the character. This is done by segmenting the pink pixels from the non pink pixels and extracting the ones that contain information about the character. Bounding boxes are created separately for every character by looking for a break in the pink pixel information between characters. These bounding boxes are then converted into a constant size of 20x20.

Dimension Reduction

Neural Networks require a consistent number of inputs across all training examples [6]. This feature constrains the Image Processing algorithm to use a constant number of pixel values when being fed to the neural network. A reliable number of input values to the neural network were found to be 20x20 or 400 by trial and error. Different algorithms were considered to perform the averaging process to convert a random sized bounding box to a constant 20x20. One algorithm that retained the original shape best was the simple averaging process. The bounding boxes formed for the characters were of arbitrary sizes and to make them all a 20x20 image, averaging operators were used. The bounding boxes were all padded to fit one of 20x20, 40x40, 60x60, 80x80, 120x120, 160x160, 180x180 or 240x240 sized boxes. Now 2x2 or 3x3 averaging operators were applied to them when necessary, to reduce them to 20x20. We did not encounter bounding boxes that exceeded 240 pixels in height or width. Hence sizes over 240x240 were not considered. This is the input to the neural network used to classify the images. Figure 2a shows the image as it is taken by AIBO and saved to the memory stick. This image was originally 352x288 in size. Figure 2b shows the 20x20 input to the neural network after being processed.

Notice that the inputs to the neural network retain the same shape as the original image. Uniform Thresholding has been applied to the pixel values making them high (1) or low (0) where high denotes pink values and low denotes non pink values [1,2]. The same process is carried out for every character of the expression and every 20x20 input of a character is fed to the neural network.

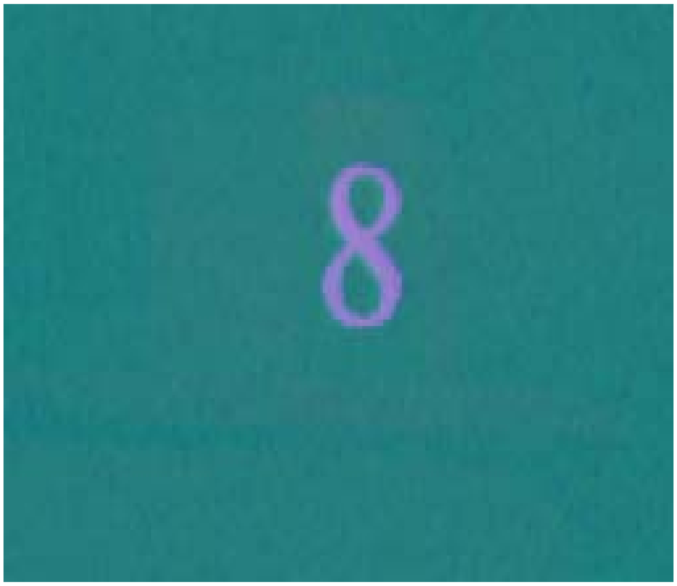


Figure 2a: Original RAW Image taken by AIBO

```
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,1,1,1,1,1,1,1,0,0,0,0,0,0
0,0,0,0,0,0,1,1,1,1,0,1,1,1,1,0,0,0,0,0
0,0,0,0,0,0,1,1,1,0,0,0,1,1,1,0,0,0,0,0
0,0,0,0,0,1,1,1,1,0,0,0,1,1,1,0,0,0,0,0
0,0,0,0,0,1,1,1,1,0,0,0,1,1,1,0,0,0,0,0
0,0,0,0,0,0,1,1,1,1,0,1,1,1,1,0,0,0,0,0
0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,0,0,0,0
0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,0,0
0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,0,0
0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,0,0,0,0,0
0,0,0,0,0,1,1,1,1,0,1,1,1,1,1,0,0,0,0,0
0,0,0,0,0,1,1,1,0,0,0,1,1,1,1,0,0,0,0,0
0,0,0,0,0,1,1,1,0,0,0,0,1,1,1,0,0,0,0,0
0,0,0,0,0,1,1,1,0,0,0,0,1,1,1,0,0,0,0,0
0,0,0,0,0,1,1,1,1,0,0,1,1,1,1,0,0,0,0,0
0,0,0,0,0,0,1,1,1,1,1,1,1,1,0,0,0,0,0,0
0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Figure 2b: 20x20 Neural Network input

The Learning Process

The process of learning is divided into two parts, Offline Learning and Online Testing. The reason for the two different parts is mainly due to time constraints. It takes a prohibitively long time for the learning process to be performed onboard AIBO’s processor. The advantage of having the robot perform real time learning is that it can better react to new circumstances or environments than offline learning allows. But real time learning also increases the interference effect associated with Neural Networks where learning in one zone causes loss of learning in other zones [12]. Since the training examples have been taken under various lighting conditions it well represents the test samples that AIBO might encounter. Hence the disadvantages that one might face by training offline are offset by the varied training examples. The neural network is not trained to detect rotational variance in images since the numbers and operators are viewed upright only.

Offline Learning

Offline Learning and Online Testing are carried out using Neural Networks. The reason for choosing Neural Networks is the ease of portability these systems possess. A learned system can be transferred to any robotic platform by merely transferring a set of weighs which are numbers. Another reason is that the Sony AIBO runs all the behaviors or applications stored on a Memory Stick$^{®}$ which is 16 MB in size. It is important that the Memory Stick$^{®}$ is not used up entirely for the learning process.

The back propagation neural network has been used extensively for classification systems and in pattern recognition [6, 10]. Hence this neural network was considered with 400 inputs which correspond to a 20x20 image as shown in Figure 2b. The number of hidden nodes was selected to be 40 based on trial and error. The system has 18 output nodes, one for each character of the mathematical expression. The 18 characters considered are, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, *, /, ^, %, (,). The input sent to the neural network is not scaled, since it has already been scaled to 0 or 1. The logistic activation function was used on the hidden and output layers.

Two sets of training samples were used for the training process. One set of training samples consisted of one character per image as shown in Figure 2a. Another set of training examples consisted of groups of 6 characters each as shown in Figure 3. Three groups of 6 characters each were used as training samples. The network produced high accuracy levels when trained and tested on individual characters. But training on single character images alone and testing on expression string images was not successful and resulted in low accuracy. Since the test images consisted of a group of characters, it seemed rational to make the training examples a group of characters as well. Each group of characters and each individual character had 11 images as training samples.

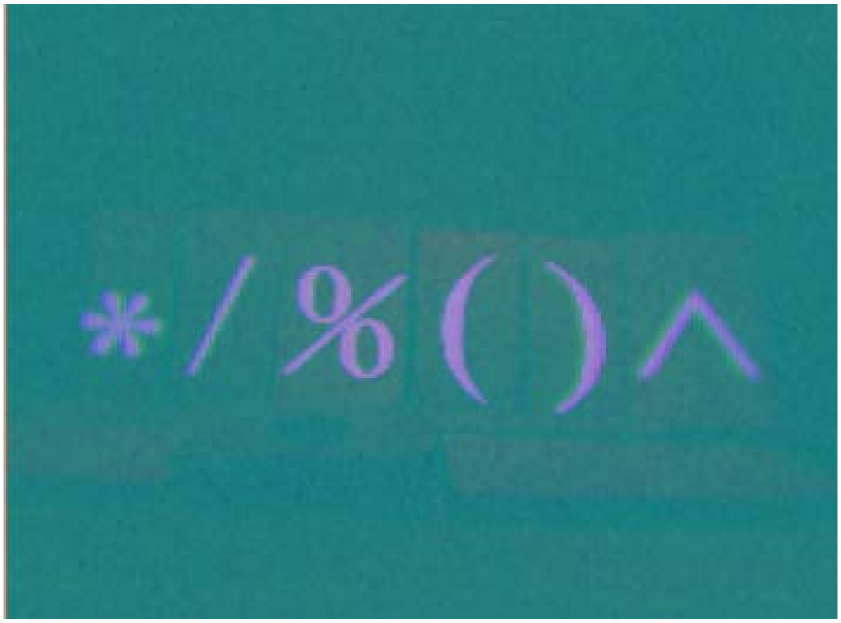


Figure 3: One group of 6 characters used as training sample.

Online Testing

Online Testing is a process of using the learned weights and classifying new instances based on these weights. In this testing process, AIBO runs a feed forward neural network which takes an image taken by the camera, processes the image using the onboard image processor and feeds the reduced image to the neural network. AIBO serves as a platform for testing the results obtained from offline learning. Figure 4 shows a typical expression that AIBO can parse. This expression is segmented into characters, the neural network is run on the individual characters and classified numbers and operators are fed to the algorithm that solves math expressions.

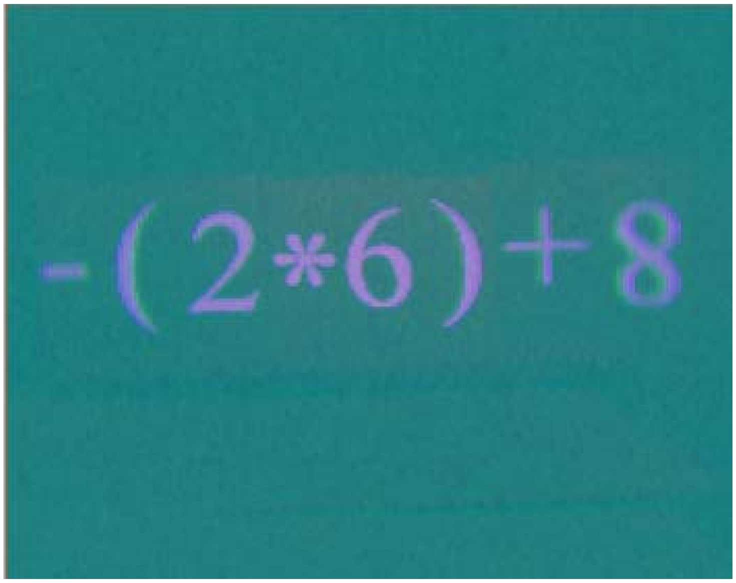


Figure 4: An expression to be solved by AIBO

Solving Mathematical Expressions

As a means of testing the entire process, an algorithm was developed to parse and solve the mathematical expression observed by AIBO. After all the characters in the mathematical expression have been identified, they are analyzed by the Expression Solver which determines if the equation is valid. Every recognized character in the image is sent to the expression solver which then appends all the characters in the image to calculate the result. Non-valid expressions include an expression divisible by 0, parentheses mismatch or just improper expressions with two operators next to each other. If the expression is valid, AIBO proceeds to calculate the result which is printed out on the telnet console on the laptop over the wireless network. The accuracy of the results of the Expression Solver depends entirely on the accuracy of the classification process. When AIBO encounters an invalid expression it shakes its head, says the expression is invalid and waits for the user to give it a new expression to calculate.

Results and Discussion

The experimental verification of the behavior was performed by letting AIBO explore its surroundings looking for pink equations. This part of the experiment has been adapted from ChaseBallBehavior (going towards pink objects) and CameraBehavior (for taking pictures) of the Tekkotsu framework. If AIBO finds a pink object, it takes a picture of it and determines if it is an equation. If it is not an equation, AIBO moves on and continues searching for other pink objects. Once a pink equation is found, it recognizes the characters and prints the result of the equation on the telnet port.

The minimum average error over 100 epochs for the training samples was observed to be 0.0001. The minimum average error for test samples over 200 events was observed to be 0.0167. That is, given a test set of individual numbers AIBO never got the answers wrong unless the character was blurred along the edge of the image. These low error values increased the accuracy of the classification process and if positioned with a full view of the expression AIBO always gave the right answers. But since the characters were being classified all at once, the overall error in calculating the result of an equation increased. AIBO solved the equations correctly 95% of the time.

One hardware constraint observed was the camera on AIBO not being able to handle more than 8 characters in an equation at one time. Panning the head to look for different parts of images has been considered and will be included in future research. When an equation consisted of more than 8 characters, AIBO was not able to correctly identify the characters at the edges of the image. The actual testing time observed on AIBO was around 3-4 seconds per expression which surpasses the time taken by a human to solve the same expression.

Alignment and Positioning of AIBO in front of the expression in order for it to see the entire expression are difficult challenges. At times AIBO would need us to intervene and help out with the alignment. The new panning scheme is expected to eliminate the need for any external help.

Conclusions

The process of object identification has immense potential in the field of Robotics. The ability of robots to identify particular objects based on feature recognition and pattern recognition has applications in fields such as Search and Rescue Missions and Security Monitoring Systems. This application of AIBO can be extended to reading. AIBO can be made to recognize the alphabet and an algorithm can be developed to make AIBO read words and reply to them. Using this application as a base, AIBO has the ability to learn and simulate intelligence in a variety of situations. Also as an extension of this application, AIBO can be taught gestures and can generate responses to them. Our goal is to embed a number of these behaviors on AIBO to reflect a broad spectrum of elementary educational tasks.

These applications can be considered as providing a means of using AIBO as an educational tool exhibiting intelligent behavior to encourage learning among youngsters. Watching AIBO solve mathematical expressions, or reading sentences and replying to them or reacting to gestures can be used to make learning fun for kids. And AIBO is an ideal platform to showcase these tasks because of its status as an entertainment dog.

References

[1] Jahne Bernd, *Digital Image Processing*, Springer-Verlag, 2002.

[2] Nixon, Mark S., and Aguado, Alberto S., *Feature Extraction and Image Processing*, Newnes Publications, 2002.

[3] A. Grossmann and R. Poli, Continual Robot Learning with Constructive Neural Networks, Proceedings of the 6th European Workshop EWLR-6 Brighton, England, August 1997, pp. 95-109.

[4] Miano John, *Compressed Image File Formats*, Addison-Wesley Publications, 1999.

[5] N. Chagas and J. Hallam, A Learning Mobile Robot: Theory, Simulation and Practice, Proceedings of the 6th European Workshop EWLR-6 Brighton, England, August 1997, pp. 142-155.

[6] Bishop, C.M., *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford UK, 1995

[7] J. Ramesh, K. Rangachar, S. Brian G. *Machine Vision*, New York: McGraw-Hill, p.474-479.

[8] Tekkotsu, http://www.tekkotsu.org

[9] RoboCup, http://www.robocup.org

[10] Shalkoff, R.J., *Pattern Recognition- Statistical, Structural and Neural Approaches*, Wiley and Sons Inc., NY USA, 1992

[11] Trier, O. D., Jain, A. K. and Taxt, T., Feature Extraction Methods for Character Recognition – a Survey, *Pattern Recognition*, 29(4), pp. 641-662, 1996

[12] S. Weaver, L. Baird, and M. Polycarpou. An Analytical Framework for Local Feedforward Networks. *IEEE Transactions on Neural Networks*, 9(3), 1998.

[13] OPEN-R SDK http://www.openr.aibo.com

Chapter 3

Gesture Recognition on AIBO[2]

[2] Radhakrishnan, S and Potter, D.W. To be submitted to IAAI06 – Innovative Applications of Artificial Intelligence Conference.

Abstract

This paper explains a method of employing a learning algorithm to efficiently classify gestures made by a human to a robot as a more powerful communication interface for Human Computer Intelligent Interaction. AIBO classifies the gestures performed by a person using neural networks and then performs actions corresponding to the gestures. This application is designed to make AIBO more accessible as a robot attuned to human behavior.

Introduction

Gestures are expressive meaningful body motions- i.e., physical movements of the fingers, hands, arms, face, head, or body with the intent to convey information or interact with the environment. There are three functional goals of human gestures, semiotic which refers to communicating meaningful information, ergotic which refers to manipulating the environment and epistemic which refers to discovering the environment through tactile experience [1]. In this paper we concentrate on the semiotic goal of gestures for humans to communicate effectively with robots. The use of human gestures has become an important research aspect of Human Computer Intelligent Interaction.

Sony AIBO was originally marketed as an entertainment robot but the abilities that the robot possessed in being a research tool led SONY to actively promote AIBO for academic research. AIBO was one of the first robots to change the entire perception of human-robot interaction. Its ability to act autonomously has been an important reason for its success. This paper aims to extend the existing behavior capabilities of AIBO to include gesture recognition.

It is important for robots to recognize objects based on features rather than color because it leads to better recognition capabilities. Many of the recognition capabilities of AIBO rest on its ability to distinguish colors and this aspect has been researched extensively in the Robocup series. This research paper extends existing work by training AIBO to learn gestures based on the features associated with the gestures.

Previous research in gesture recognition on AIBO has been done on a remote computer and the results have been beamed on to the robot through a wireless connection [2]. The research has been done on the same model of AIBO as ours. To create a truly autonomous behavior, we need to be able to perform the processing on board the robot instead of a remote computer. This research builds on our previous work with AIBO to recognize numbers and characters to solve mathematical expressions [3]. The results here are compared to the results obtained in the previous research as far as classification accuracy goes. This methodology can be presented to AIBO to pick up visual cues for various actions that it needs to perform. Our approach currently considers static hand gestures performed in a controlled environment with a constant background.

The rest of the paper is divided as follows: We describe the platform used to program AIBO in the next section. Section 3 justifies the choice of the learning algorithm used. Section 4 describes the Image Processing tasks that were performed on the image taken by AIBO. Section 5 explains the offline learning and online testing performed using neural networks. Section 6 explains how the recognized gestures are mapped to the corresponding actions that AIBO performs. We then provide a section to describe the results obtained with the behavior created. Lastly we discuss future work that can be done to extend this research.

Background – AIBO, OPEN-R and Tekkotsu

AIBO as mentioned earlier is an entertainment robot that has been used extensively for research in universities around the world. There are many features of AIBO that make it an extremely efficient robot. AIBO has a 384 MHz MIPS (million instructions per second) processor that can handle multiple tasks simultaneously without actually affecting performance. It also has a 32 MB onboard RAM. AIBO has a wireless card that can be used to connect to a PC and can transmit and receive data packets to and from the PC. The wireless connection is 802.11b and the PC and AIBO can be connected through a router or over a peer-peer network connection.

Behaviors created for AIBO are created on the PC and stored on a memory stick that is inserted into AIBO. It has 20 joints, 18 of which are PID (proportional integral derivative) joints with force sensing and 2 Boolean joints. It has 9 LEDs that are used to express the emotions of the robot. It has a CCD (charge coupled device) camera with a field of view that is 47.8° high and 57.6° wide, resolutions are 208x160, 104x80, 52x40 and can take up to 25 frames per second. It has stereo microphones and can detect the direction of sound using them. To avoid obstacles AIBO has an infrared distance sensor which has a range of 100-900 mm. It has pressure sensitive and Boolean (on-off) buttons and updates the sensor’s reading every 32 ms. The most important feature of them all is that AIBO is programmable.

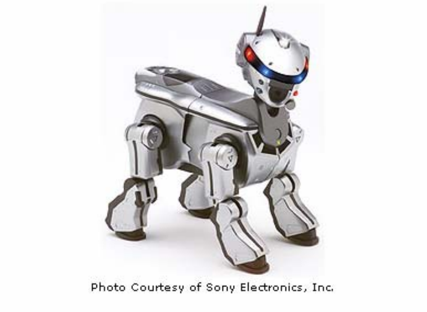


Photo Courtesy of Sony Electronics, Inc.

Figure 1: AIBO ERS220A

OPEN-R is a software development environment used to program AIBO with C++ as the programming language. OPEN-R programs are built as a collection of concurrently running OPEN-R modules which are essentially different simple actions combined to form one single complex task [4]. One complaint regarding the OPEN-R architecture is that it is complex. Also the sample programs given on Sony's website do not have inline comments on them to provide an understanding of the workings of OPEN-R. Access to sensors and actuators using OPEN-R still involves the use of some low level programming skills. This led to the development of the Tekkotsu environment.

The Tekkotsu framework was developed by Carnegie Mellon University using OPEN-R as its basic framework. Tekkotsu allows for creating behaviors that can interact with OPEN-R without the hassle of dealing with low level programming. The Tekkotsu framework provides access to all the sensors and actuators on AIBO by just invoking the necessary values and lets the user concentrate on creating behaviors [12]. It is possible to control AIBO wirelessly using a peer to peer network connection and the telnet console on a host computer or laptop. Tekkotsu has implemented a GUI called Tekkotsu Mon to remotely access the behaviors that run on AIBO from a laptop. This facilitates an easier debugging process and better PC-Robot communication.

The Learning Algorithm

Of the different learning algorithms that can be used (Bayesian Learning, Nearest Neighbor, Neural Networks, Decision trees and others), Neural Networks are most commonly employed in classification and pattern recognition problems. This is because of their ability to generalize on fewer training examples and their robustness to error [5, 14]. Unlike other learning methods

Neural networks that have learned a particular task can be used on any system by simply porting the weights of the network to the system. This leads to saving memory on devices that may not have much memory to start with. Due to the potentially long training time associated with real time training, the actual learning process is conducted on a laptop and the results are saved on to the Memory Stick®. Since training and testing are to be conducted on different platforms, it is important to be able to transfer the trained results to the testing platform with ease. Since the trained results of a neural network are a set of weights, they can easily be transferred to AIBO.

Our experimental goal is to have AIBO look at a hand gesture, take an image of it, process the image to send to the neural network and then let the neural network classify the image based on the existing set of learned weights for the different gestures. The neural network employed in this problem is a back propagation neural network with 400 pixel inputs, a 40 unit hidden layer and a 9 unit output. The 9 gestures that AIBO can recognize are just a sample of the number of gestures it can actually classify.

Image Processing

Image Processing used in the Robocup tournament emphasizes the need for color segmentation and object recognition using color recognition [13]. This is an extremely efficient method when considering specific environments that are defined based on color as with the robot soccer field. But under circumstances that do not pertain to the constraints of Robocup we need a different methodology to recognize different objects. There are two variations to the image processing done in this paper. One method processes grayscale images while the other method processes color images. The Sony AIBO comes equipped with a color CCD camera and is tuned to detect pink objects, such as the pink ball that comes shipped with it. Hence the gestures are performed with pink gloves when dealing with color images. The gestures are performed with white gloves when considering grayscale images. The idea behind performing the same gesture recognition task using different image formats is to evaluate the difference in performance between using color images and grayscale images. The images taken by AIBO are 352x288 in size which is the largest sized picture that AIBO can take. The pictures are taken in the RAW format which is not modified by any compression algorithm. The image is initially taken by AIBO and then processed by the image processing algorithm.

The image processing is classified into three parts. The first part extracts the pixel information from the rest of the image (headers). This part is different for the grayscale and color images as explained later. This pixel information is then sent to the portion of the algorithm that detects bounding boxes. Bounding boxes are detected based on a threshold level. We assume in a grayscale image that the gesture is the brightest part of the image. In the case of color images the gesture is of a different color than the rest of the image. This does not lead to classification of the gestures based on color because we still classify the images based on features using the learning algorithm.

The third part compresses the image to a constant size which is then fed to the neural network. The image processing algorithm runs individually on every image that is saved, by creating a bounding box around the object of interest in the image, compressing the bounded object to a constant size and sending it to a neural network to be classified. This method works the same for both color and grayscale images. Figure 2 shows the different hand gestures that are considered in this research.

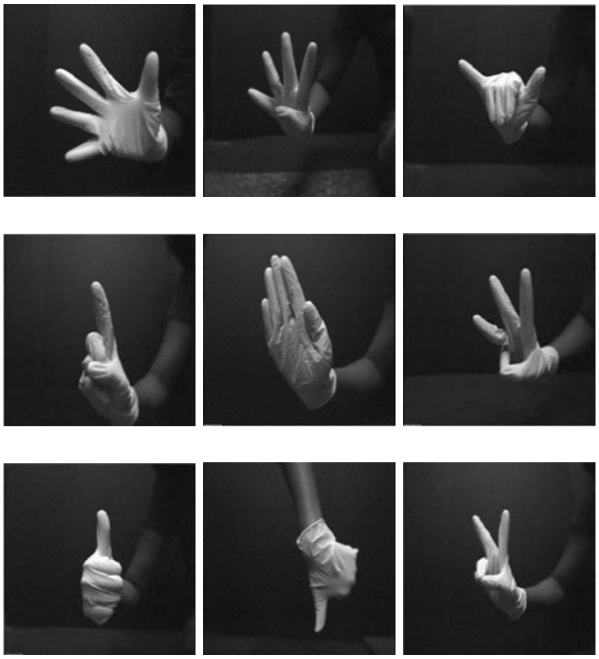


Figure 2: A sample set of the hand gestures

Pixel Extraction from color and grayscale images

Pixel extraction involves the removal of the header and other unnecessary features of images while retaining just the pixel values. The image saved to the Memory Stick® is in the RAW format which means the pixels are unchanged from the camera. The color format of the color image is YUV where Y stands for brightness, U and V stand for chrominance. U relates to the blue-yellow color components while V relates to the red-green color components of the color image [6]. Since the gestures are all pink in color, it is enough to extract only the V component of the pixel. This method leads to fewer computations on the part of the image processing algorithm since it does not deal with the other two dimensions of the color image. Since the V component contains color information, it is to a large extent independent of varying illumination. This method cannot be recommended for other recognition tasks since a large amount of information is lost by ignoring Y and U components. The V component is extracted and sent to the step where a bounding box is created. In the case of grayscale images, the pixel values all relate to brightness levels. The pixels are just extracted and sent to the bounding box segment which is explained next.

Edge Detection: Instead of detecting the images using color and brightness threshold levels, an alternative is to detect the gesture by using edge detection algorithms on the image and comparing the edge detected images to a general hand template. Different edge detection algorithms were used on the image, like Sobel, Roberts and Canny [8]. The Canny edge detector gave the best edge detected image, based on noise levels in the modified image. But edge detection could not be used because when the edge detected image was compressed to a 20x20 size, it lost its shape properties. The reason for this behavior is the edges being too thin in the image. So when a 240x240 image is converted to a 20x20 image there are not many properties of the image that get retained. Hence this method was discontinued.

Creating a Bounding Box

Bounding boxes are created when the necessary parts of the image are to be segmented from the rest of the image. This way the neural network gets only the actual gesture as input which makes the classification process more accurate. A bounding box is created around the gesture based on brightness or certain color values. This is done by segmenting the pink pixels from the non-pink pixels and extracting the ones that contain information about the character. These bounding boxes are then converted into a constant size of 20x20.

Dimension Reduction

Neural Networks require a consistent number of inputs across all training examples [7]. This feature constrains the Image Processing algorithm to output a constant number of pixel values which is then fed to the neural network. A reliable number of input values to the neural network were found to be 20x20 or 400 by trial and error. Different algorithms were considered to perform the averaging process to convert a random sized bounding box to a constant 20x20. One algorithm that retained the original shape best was the simple averaging process. The bounding boxes formed for the characters were of arbitrary sizes and to make them all a 20x20 image, averaging operators were used. The bounding boxes were all padded with non-pixel values (typically 0) to fit one of 20x20, 40x40, 60x60, 80x80, 120x120, 160x160, 180x180 or 240x240 sized boxes. Then 2x2 or 3x3 averaging operators were applied to them when necessary, to reduce them to 20x20. We did not encounter bounding boxes that exceeded 240 pixels in height or width. Hence sizes over 240x240 were not considered. The 20x20 image is the input to the neural network that classifies the images.

The inputs to the neural network retain the same shape as the original gesture as shown in Figures 3a, 3b and Figures 4a, 4b. Uniform Thresholding has been applied to the pixel values making them high (1) or low (0) where high is part of the gesture and low is the background [8, 9].

The Learning Process

The Learning algorithm is divided into two parts, Offline Learning and Online Testing. Offline here refers to learning taking place on a remote computer and not on the actual robot. It takes a prohibitively long time for the learning process to be performed onboard AIBO’s processor. The advantage of having the robot perform real time learning is to better react to new circumstances or environments than what offline learning would allow. But real time learning also increases the interference effect associated with Neural Networks where learning in one zone causes loss of learning in other zones [10]. The training examples for the grayscale images have been taken under various lighting conditions and well represent the test samples that AIBO might encounter. The training examples for color images need not be taken under different lighting conditions since we consider only the V component of the image which is independent of brightness conditions. Hence the disadvantages that one might face by training offline are offset by the varied training examples. The neural network is not trained to detect rotational variance in images since gestures are viewed upright only.

Offline Learning

Offline Learning and Online Testing are carried out using Neural Networks. The reason for choosing Neural Networks is the ease of portability these systems possess. A learned system can be transferred to any robotic platform by merely transferring a set of weighted numbers. Another reason is that the Sony AIBO runs all the behaviors or applications stored on a Memory Stick® which is 16 MB in size. It is important that the Memory Stick® is not used up entirely for the learning process.

The simple back propagation neural network has been used extensively for classification systems and in pattern recognition [5, 11]. Hence this neural network was considered with 400 inputs which correspond to a 20x20 image as shown in Figures 3b and 4b. The number of hidden nodes was selected to be 40 based on trial and error. The system has 9 output nodes, one for each gesture performed. The different gestures are depicted in Figure 2. The input sent to the neural network is not scaled, since it has already been scaled to 0 or 1. The logistic activation function is used on the hidden and output layers.

The training set consists of grayscale and color images. The system considers grayscale and color images separately. A set of 30 images are taken as samples per gesture for each color scheme. This method looks to see which color scheme gives better results. The results are discussed in later sections.

Online Testing

Online Testing is a process of using the learned weights and classifying new instances based on these weights. The behavior on AIBO runs a simple feed forward neural network which takes the weights learned through offline learning and interprets the input gesture that is presented. AIBO serves as a platform for testing the results obtained from offline learning. The behavior runs the same image processing algorithm that is run on the offline learner. The gesture input is processed and sent to the neural network which classifies the input gesture and performs a certain action associated with it. This process is described in the next section.

Gestures to Actions

As a means of recognizing the gestures, AIBO performs certain distinct actions relating to the gestures which provide human interaction between robot and human. Tekkotsu provides us with different motion sequences that can be combined to allow for distinct actions to be performed. Sophisticated motion sequences can be created by dynamically creating a Motion Sequence Command. Different posture files (.pos) can be combined to form unique motions for the gestures. The different motions associated with the gestures are given below.

Gestures	**Actions**
One	Sit
Two	Stand
Three	Turn Left
Four	Turn Right
Five	Glance around
Stop	Freeze up
Thumbs up	Raise arm and wave
Thumbs down	Lower head and shake
Horns	Lie down

Results

Initially the gestures were performed with just the hand. This method proved to be difficult to expand on since some hands are fairer than others and get recognized better. To avoid this effect, every hand depicting a gesture was clad in a white glove or a pink glove to maintain constancy.

When considering the results of grayscale images, the network first was trained using 15 training examples per gesture. This number eventually turned out to provide unsatisfactory results because the over fitting observed gave an error rate of 50% over the test samples. Over fitting occurred due to the presence of very few training examples. The misclassifications observed were typically with images that resemble one another. For example, an image of the gesture representing 5 and another representing 4 were confused by the neural network. This was because a certain finger might be shadowed in ‘5’ and would not get detected properly. The same goes for an image of 1 and the thumbs up sign which were confused for each other about 25-30% of the time. The reason for this confusion is evident with the images of one and thumbs up as shown in Figures 3a, 3b, 4a and 4b. Two different solutions were tried out to resolve this problem. One of them involved decreasing the number of hidden nodes, the reason being that neural networks tend to over fit with an excess of hidden nodes. Another method tried was to increase the number of hidden nodes. But both methods did not affect the results and in fact gave worse results which indicated that the neural network did have a reasonably good number of hidden nodes. The other solution was to increase the number of training samples so that the neural network would have more examples to derive its representation from. The training examples were increased from 15 to 30. This method did increase the accuracy to about 70%.

The images were trained for rotation invariance and illumination invariance (to an extent). Rotation invariance was an important factor in the training because the network should be able to correctly identify hand gestures that are at different angles. When we say rotation invariance, we refer to performing the same gesture at different angles as long as the gesture still makes sense to an onlooker. The gesture can be performed at any position in the image (not necessarily the center) because the bounding box effectively zeros in on the gesture. A set of random gestures were chosen to provide a visual demonstration of this application. The different gestures considered are one, two, three, four, five, thumbs up, thumbs down, horn and stop.

Consider the two test cases, one where we use the brightness values of the image to calculate the position of the gesture and the other case where we consider the color information presented in the image to detect the position of the hand gesture. The two methods were tested and it was observed that classification has higher accuracy when the image is detected with color information rather than brightness information. The color gesture recognition system had an accuracy of 90% as opposed to grayscale images with 70%. The reason behind this result is that on occasion a grayscale image might encounter a shadow that affects the detection of the hand gesture.

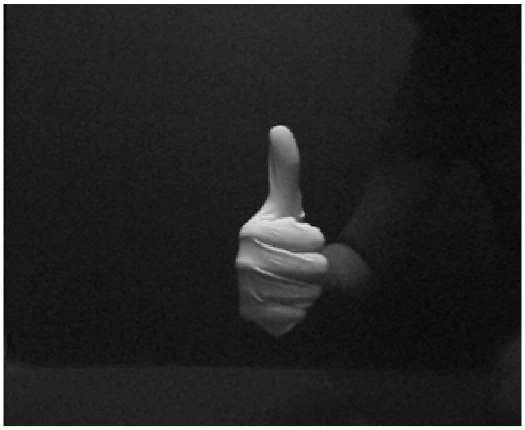
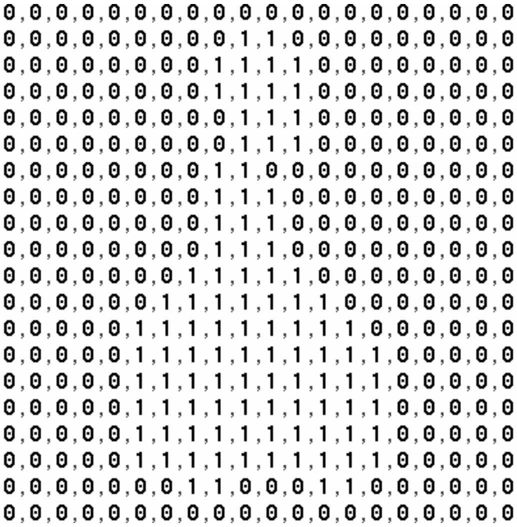


Figure 3a: Image corresponding to the ‘thumbs up’ input



```
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,0,0
0,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0,0,0
0,0,0,0,0,0,1,1,1,1,1,1,1,0,0,0,0,0,0,0
0,0,0,0,0,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0
0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0
0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0
0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0
0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0
0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0
0,0,0,0,0,0,0,1,1,0,0,0,1,1,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Figure 3b: Gesture ‘thumbs up’ as given to the NN

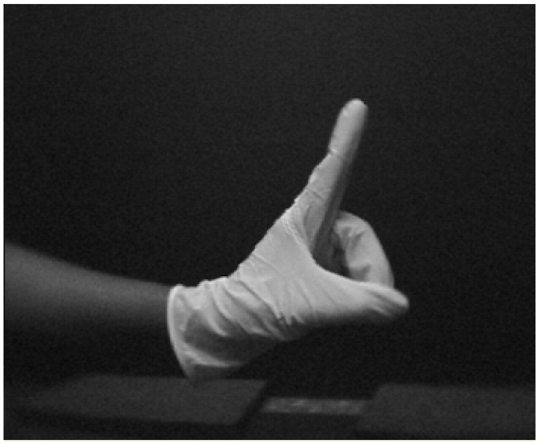
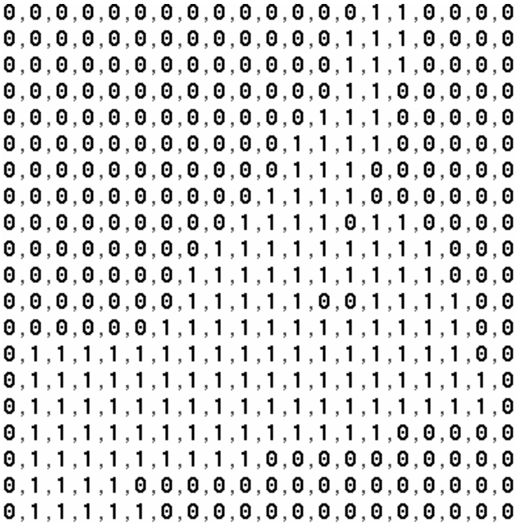


Figure 4a: Image corresponding to the ‘one’ input



```
0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0
0,0,0,0,0,0,0,0,0,1,1,1,1,0,1,1,0,0,0,0
0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,0,0,0
0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,0,0,0
0,0,0,0,0,0,0,1,1,1,1,1,0,0,1,1,1,1,0,0
0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,0,0
0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0
0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0
0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0
0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0
0,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0
0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
0,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0
```

Figure 4b: Gesture ‘one’ as given to the NN

Classification Results: The neural network classification of grayscale images gave much higher error rates than that obtained in the previous research [3]. The error rate of the classification increased because of the influence of brightness values in determining the hand gesture. If a shadow is encountered while performing the gesture there is the possibility of the gesture not being recognized properly. In spite of this disadvantage the results of the grayscale gesture recognition have been encouraging. The results of the color gesture recognition were comparable to those obtained with the previous research.

Figure 5 shows the classification results of gesture ‘one’ where two sets of observations are represented. The two observations represent the classification and misclassification of the gesture ‘one’. The values represented here are the output values of the neural network. The light shades are the correctly classified results while the dark shades are the misclassified results. It can be observed that gesture ‘one’ and gesture ‘thumbs up’ are confused by the neural network and sometimes ‘one’ is misclassified as ‘thumbs up’. Gestures like ‘three’, ‘horn’ and ‘thumbs down’ have a value of 0 since they do not have any similarity to ‘one’. This brings down the accuracy levels to about 70% in grayscale images.

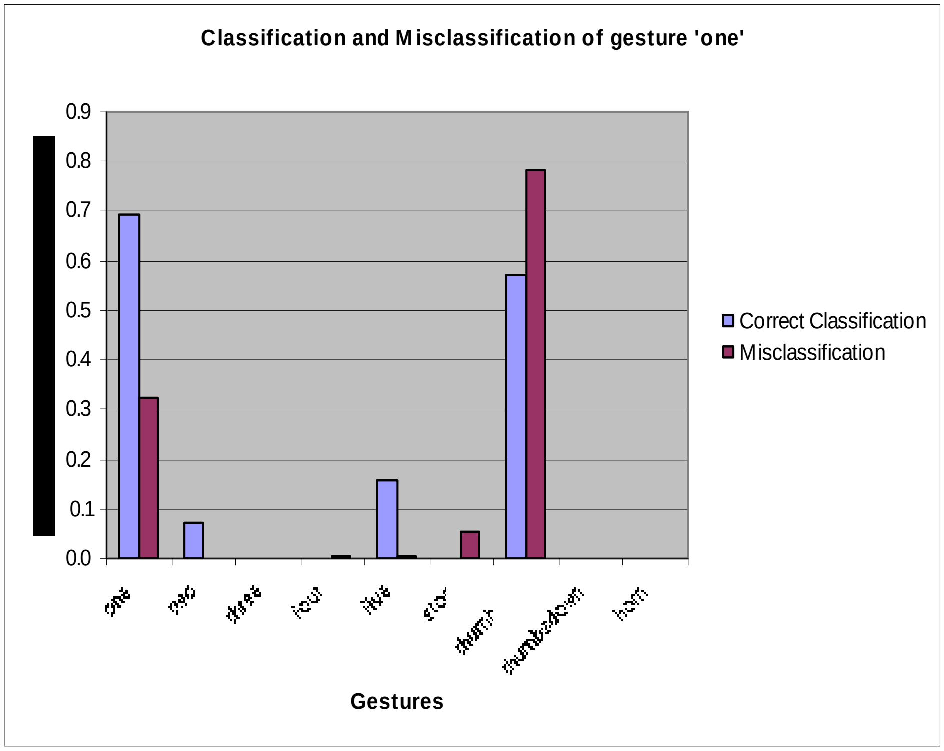


Figure 5: Classification result of gesture ‘one’

The color images also had classification errors with the same set of images since they were too similar to each other. When using color images, the error values are solely attributed to the similarity of some gestures.

This method was tested on other people’s hands and the results were comparable to that obtained with just one’s person’s hand. This is because of the significance of the gloves used to cover the hands. As long as a gesture similar to those trained is presented to AIBO, it can detect them with the same accuracy as presented above irrespective of the hand that presents the gestures.

Future Work

Our research attempts to achieve rotational invariance (to an extent) and view independent gesture recognition on AIBO. This method has worked well on individual images taken by AIBO. The research can be extended to incorporate noise in the environment. We currently use a constant black background to achieve a distinction between the gesture and the rest of the image. Expanding the training samples to include image sequences and dynamic gestures can be explored in future research. It is important to consider that AIBO is computationally limited with respect to implementing image processing algorithms. The reason we chose not to use image sequences was that it becomes too computationally expensive for AIBO to handle. But different techniques can be researched to reduce the processor intensive image analysis computations.

Gesture Recognition has been researched extensively on systems that have enough processor capabilities to run them. It is important to extend such research to platforms that allow for a powerful interface between humans and machines. This platform can be extended to successfully learn and analyze handwriting of different people. It has already been successfully tested on numbers and operators which were combined to solve mathematical expressions.

References

[1] Cadoz, C. (1994) *Les Réalites Virtuelles*. Dominos, Flammarion.

[2] M. Hasanuzzaman, T. Zhang, V. Ampornaramveth, Kiatisevi, P., Shirai, Y., Ueno, H., "Gesture based human-robot interaction using a frame based software platform", *Proceedings of International Conference on Systems, Man and Cybernetics* (IEEE SMC 2004), pp.2883-2888, Hague, Netherlands, Oct. 2004.

[3] S. Radhakrishnan and W. D. Potter, Basic Math via Character Recognition on AIBO Robot Dog, *Transactions on Information Science and Applications*, 1(3) pp. 127- 133, 2005

[4] OPEN-R SDK http://www.openr.aibo.com

[5] Bishop, C.M., Neural Networks for Pattern Recognition, Oxford University Press, Oxford UK, 1995.

[6] Miano John, Compressed Image File Formats, Addison-Wesley Publications, 1999.

[7] J. Ramesh, K. Rangachar, S. Brian G. Machine Vision, New York: McGraw-Hill, p.474-479.

[8] Jahne Bernd, Digital Image Processing, Springer-Verlag, 2002.

[9] Nixon, Mark S., and Aguado, Alberto S., Feature Extraction and Image Processing, Newnes Publications, 2002.

[10] S. Weaver, L. Baird, and M. Polycarpou. An Analytical Framework for Local Feedforward Networks. *IEEE Transactions on Neural Networks*, 9(3), 1998.

[11] Shalkoff, R.J., Pattern Recognition- Statistical, Structural and Neural Approaches, Wiley and Sons Inc., NY USA, 1992

[12] Tekkotsu, http://www.tekkotsu.org

[13] RoboCup, http://www.robocup.org

[14] A. Grossmann and R. Poli, Continual Robot Learning with Constructive Neural Networks, *Proceedings of the 6th European Workshop* EWLR-6 Brighton, England, August 1997, pp. 95-109.

Chapter 4

Conclusions and Future Direction

This research achieves the goal of performing the task of recognizing different objects based on the features possessed by the objects. We consider a non-noisy background while recognizing numbers and gestures. The research can be expanded to include noisy environments that algorithms that adapt to the noise. One has to note that such noisy environments do potentially overwhelm the robot's processor and could lead it to decrease in performance or crash altogether. But that can be remedied by developing a robust image processing algorithm that does not use up too much computation. In gesture recognition we have worked with static gestures and they can be expanded to include dynamic gestures. Also we process one image at a time for recognition. This was done with a view to minimize computational pressure on AIBO. A different algorithm can be developed to use image sequences to detect dynamic gestures.

The applications developed can be considered as providing a means of using AIBO as an educational tool exhibiting intelligent behavior to encourage learning among youngsters. Watching AIBO solve mathematical expressions, or reading sentences and replying to them or reacting to gestures can be used to make learning fun for kids. These behaviors can be extended to include learning the alphabet and reading sentences. AIBO can store a knowledge base of sentences and can answer questions posed to it. Using these applications as a base, AIBO has the ability to learn and simulate intelligence in a variety of situations. Our goal is to embed a number of these behaviors on AIBO to reflect a broad spectrum of elementary educational tasks.

It is important to note that we have explored only the visual cognition capabilities of AIBO. As has been discussed before AIBO has a whole slew of sensors and actuators that can be efficiently used to perform complex tasks that can simulate or exhibit intelligence. And AIBO is an ideal platform to showcase these tasks because of its status as an entertainment dog.