1-31-2013

Human inspired pattern recognition via local invariant features

Dominic Ron Maestas

Dominic Ron Maestas

Candidate

Mechanical Engineering

Department

This dissertation is approved, and it is acceptable in quality and form for publication:

Approved by the Dissertation Committee:

Dr. Ron Lumia, Chairperson

Dr. Rafael Fierro

Dr. John Russell

Dr. Peter Vorobieff

HUMAN INSPIRED PATTERN RECOGNITION VIA LOCAL INVARIANT FEATURES

BY

DOMINIC RON MAESTAS

B.S., MECHANICAL ENGINEERING
UNITED STATES AIR FORCE ACADEMY
1998

M.S., MECHANICAL ENGINEERING
UNIVERSITY OF NEW MEXICO
2003

DISSERTATION

Submitted in Partial Fulfillment of the
Requirements for the Degree of

Doctor of Philosophy

Engineering

The University of New Mexico
Albuquerque, New Mexico

DECEMBER, 2012

Acknowledgments

I am especially grateful to Professor Ron Lumia—my advisor. It has been one of the most challenging educational experiences of my life. Not only was it a pleasure working with him, but also to be able to study under such a renowned scholar and person of genius, I am in gratitude for the experience.

I also would like to express my thanks to Professor Rafael Fierro, Professor John Russell and Professor Peter Vorobieff. From my comprehensive exam, to journal reviews, and dissertation feedback it has been an honor and a privilege.

This would not have been possible without the support of Intel Corporation and its Tuition Reimbursement Program. In particular, I want to thank Jeffrey Berger for his mentorship, encouragement, and belief in my abilities to improve my professional development skills. Thank you to, fellow P.E., Lincoln Busselle for his belief in education and developing others so that they can realize their full potential.

I also am grateful to Dennis Shaulis for approving my coursework and his mentorship as a Department Head. Thanks to David Lang, Ravi Periasamy, and Joe Baca for your support to get into Graduate School and for being positive role models in my life.

To my lovely, beautiful, and intelligent wife, I love you so much and you bring a touch of class into my life that I am most appreciative and grateful for having you in my life. Mom and Dad, thanks for being great parents and role models and getting me over this last hurdle in my academic life. I could not have gotten better parents.

This work was supported in part by the DOE URPR Grant # DE-FG52-04NA25590.

Man's Flight Through Life Is Sustained by the Power of his Knowledge. - Austin Miller

Never has this been so true and applicable to my life. I have always made academics, learning, and knowledge an essential component to my life's activities. I have been rewarded not only with the high flight that comes along with such accomplishment, but the freedom that comes along with independent thought and scientific exploration in an ever competitive and increasingly flat world. For that, I am especially appreciative since most humans in the world never experience such a life full of richness--only struggle and no ability of realizing their full potential.

DOMINIC R. MAESTAS

Human Inspired Pattern Recognition Via Local Invariant Features

By

Dominic Ron Maestas

B.S., Mechanical Engineering, United States Air Force Academy, 1998
M.S., Mechanical Engineering, The University of New Mexico, 2003
Ph.D., Engineering, The University of New Mexico, 2012

ABSTRACT

Vision is increasingly becoming a vital element in the manufacturing industry. As complex as it already is, vision is becoming even more difficult to implement in a pattern recognition environment as it converges toward the level of what humans visualize. Relevant brain work technologies are allowing vision systems to add capability and tasks that were long reserved for humans. The ability to recognize patterns like humans do is a good goal in terms of performance metrics for manufacturing activities. To achieve this goal, we created a neural network that achieves pattern recognition analogous to the human visual cortex using high quality keypoints by optimizing the scale space and pairing keypoints with edges as input into the model.

This research uses the Taguchi Design of Experiments approach to find optimal values for the SIFT parameters with respect to finding correct matches between images that vary in rotation and scale. The approach used the Taguchi L18 matrix to determine the optimal parameter set. The performance obtained from SIFT using the optimal solution was compared with the performance from the original SIFT algorithm parameters. It is shown that correct matches between an original image and a scaled, rotated, or scaled and rotated version of that image improves by 17% using the optimal values of the SIFT.

A human inspired approach was used to create a CMAC based neural network capable of pattern recognition. A comparison of a 3 object, 30 object, and 50 object scenes were examined using edge and optimized SIFT based features as inputs and produced extensible results from 3 to 50 objects based on classification performance. The classification results prove that we achieve a high level of pattern recognition that ranged from 96.1% to 100% for objects under consideration. The result is a pattern recognition model capable of locally based classification based on invariant information without the need for sets of information that include input sensory data that is not necessarily invariant (background data, raw pixel data, viewpoint angles) that global models rely on in pattern recognition.

Contents

List of Figures

List of Tables

Acronyms

ANOVA Analysis of Variance

CMAC Cerebellar Model Articulation Controller

DOE Design of Experiments

FOV Field of View

MSER Maximally Stable Extremal Region

NCC normalized cross correlation
NN Neural Network

RANSAC Random Sampling and Consensus

SIFT Scale Invariant Feature Detection
SURF Speeded-Up Robust Features

UNM University of New Mexico

VIP Video and Image Processing

Chapter 1: INTRODUCTION

1.1 Goal

The goal of the proposed research is to develop a pattern recognition model that consists of a cerebral framework that locally trains, classifies, and recognizes patterns based on local invariant keypoints extracted from image scenes. This was accomplished based on local information within a scene and trained via the implementation of an algorithm based on keypoints utilized as inputs into a neural network similar to those inspired by Cerebellar Model Articulation Controller (CMAC) neural network and in the human visual cortex. The parameters were optimized to extract invariant keypoints and establish a model of pattern recognition.

1.2 Problem

Successful pattern recognition is determined by the probability that a pattern under consideration is the same as a known pattern stored in a database with similar attributes. Current successful pattern recognition schemes are global in nature and rely on input sensory data that are not necessarily invariant (background data, raw pixel data, viewpoint angles). The net result is that most of these schemes require large intensive databases to use probability based matches from images and their associated pixel information. To be successful, comparisons must be near perfect matches to the trained database and errors occur that lead to incomplete or false recognition. The most common methods for pattern recognition often involve mathematical and statistical methods that utilize Bayesian based algorithms. While these algorithms provide sufficient pattern recognition against known databases, the algorithms operate differently than the way humans recognize patterns.

Today's algorithms rarely take advantage of visual cortex inspired approaches to pattern recognition which result in large complex methods for specific tasks, but do not provide robustness for task variation. The focus of this research attempts to resolve a common problem with computer vision pattern recognition methods. The problem occurs when these created pattern recognition schemes are compared to how easily humans recognize patterns. Using the vast array of neural networks within the brain, it is proposed that humans use a cerebral approach to pattern recognition and take advantage of information from scene to scene without change—also known as invariance. A model of human-inspired and locally based pattern recognition can be developed using modern day technology to recognize patterns more like humans. The model takes advantage of the benefits of feature extraction used to detect boundaries and edges, neural networks to process information like the human brain, and invariant data that can be manipulated and extracted to control information used for pattern recognition. The result of this approach is the ability to use local information within a scene to train, classify, and recognize patterns without the need for a large external database.

Chapter 2: LITERATURE REVIEW

2.1 State of the Art

Vision is increasingly becoming a vital element in the manufacturing industry. As complex as it already is, vision it is becoming even more difficult to implement in a pattern recognition environment as it converges toward the level of what humans visualize. Relevant brain work technologies are allowing vision systems to add capability and tasks that were long reserved for humans. The ability to recognize patterns the way humans do is a good goal in terms of performance metrics for manufacturing activities. To achieve this goal, advances gained from computer vision, such as the ability to use invariant keypoints, were integrated with neural networks to create a unique model that achieves pattern recognition analogous to the human visual cortex.

2.2 Human Vision Object Recognition

To create a model of pattern recognition, it is essential to first examine theories on how humans perform pattern recognition. To do that the human physiology must be investigated in terms of learning and function. We need to know how the brain sees information. To know this we need to know the answers to the following two questions: how the brain gets the information (i.e., what are the inputs?), and how the brain translates and processes the information.

David Marr [31] was one of the first researchers to examine the brain. Marr presented his theory that the cerebellum learns to perform motor skills via learning movements or actions and learning to maintain posture or balance which are the reflexes to the observed actions. While this first approach did not answer the question of how humans see, it set the basis work for following investigations. In particular, Marr [30] presented to the Royal Society of London that the link between the cerebellum and cerebrum could be established in terms of learning by using criteria (form, color, texture, and movement). This criterion enabled the nervous system to split up its visual input into components or classification units for different objects and can be used in visual bonding and coding features based on continuities in capturing information that can be suitably invariant as a means to look for patterns of coherent matter (objects) [30]. Visual bonding is a technique for joining visual data using fixed criteria. To strengthen his argument, Marr decided to play "Devil's Advocate" and investigate what was termed the Anticortex. Marr's counter argument to the existence of a cerebral cortex suggests that an anticortex is used for primitive functions like storage and free association. Currents (spikes and firings) can be used like an SRAM structure, in today's terms, for the cerebrum to easily associate items as permanent or temporary. Then the information in memory can be used to recall instructions based on past experiences and used in future behaviors. The result is efficient control of behavior by means of a simple memory [32].

Another early researcher was Frank Rosenblatt. In terms of pattern recognition, Rosenblatt's research that is most applicable was his invention of the perceptron model. Rosenblatt [40] distinguished between object identification and recognition. Object Identification can be thought of as something of interest present in the environment so that the information is registered into the brain in a certain form and can serve as a sensor for future recall (i.e., stimuli) [40]. Object recognition can be thought of as the recall and influence that the stored information has on future behavior (i.e., knowing the difference between an attacking animal for survival vs. a harmless creature) [40]. He then combined object identification and recognition into what he called statistical separability which is using the same association and applying it to a different situation like future behavior [41]. While the first set of experiments and theories that Rosenblatt presents are primarily concerned with how the brain organizes information, statistical separability proves the brain can be adapted to apply stored data to other activities.

Now that we have a general idea how the brain utilizes the information, the next question is how does it get the information (i.e., what are the inputs?)? The eye is the window in which visual information from the world is brought into the brain for processing. As shown in Fig. 1, the major components of the eye are the iris, pupil, cornea, lens, retina, macula, fovea and optic nerve.

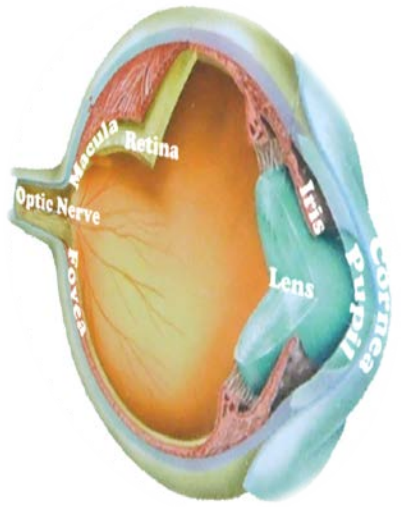


Fig. 1. Eye and its Labeled Components [11]

While the rods and cones of the retina serve as the primary sensors for processing light information, the primary component of the eye, where feature extraction takes place, is in the fovea [8]. Biederman [8] conducted some of the first experiments to prove human object recognition using feature extraction to prove the fovea segments images of unknown objects into blocks, cones, triangles, or cylinders.

The fovea is most effective in processing image information using eye movement and periods of fixation—known as human saccades [16]. Siagian and Itti [46] discussed plausibility of scene recognition by conducting an experiment where subjects could recognize objects in their peripheral vision while still focusing on primary objects directly in front of them. Zhang [54] used human subjects to show how the fovea detects "hot spots" to track the eye movement during scene recognition to then create a simulated eye movement model based on invariant features from Scale Invariant Feature Transforms (SIFT).

From the fovea the information is sent through the optic nerve and split into two paths on each side of the eye into the optic chasm for coding so that the visual cortex can reprocess the information for the brain. As shown in Fig. 2, the visual cortex is the part of the brain responsible for processing the information transmitted from the world through the optic nerve and chasm. It decodes the information for the brain and is composed of 5 regions-namely V1-V5. V1 is the primary visual cortex region and is responsible for most of the processing that takes place with V2 being the secondary visual cortex handling overflow. V3-V5 are visual association areas used by the brain for signals not processed at V1 or V2. They do not just respond to whether there is a presence of light or not, but process columns of light in specific directions—in essence they are feature extractors [8]. There are 3 types of cells that act as the feature extractors: simple, complex, and hypercomplex [34]:

- Simple cells: Respond to the presence of columns of light at a particular orientation and position.
- Complex cells: Respond to columns of particular orientations moving across the retina.
- Hypercomplex cells: Responded to moving columns but also had a strong inhibitory region at their end.

Since the transmittance from the world through the optic nerve and chasm is divided for each eye, the cells cause neural responses based on similar objects and that determines which one of the three cells activates as a primary stimuli response. The aforementioned is primarily done in V1. Regions V2-V5, also known as the striate cortex, are mainly used for auxiliary visual processing or overflows of information streaming down the optic nerve. Primary function of the striate is for identification and location.

Striates can be used to get basic shape, orientation, and awareness like those commonly described as survival instincts in human vision to determine if a large animal is in the field of view (identification) and coming to attack (location).

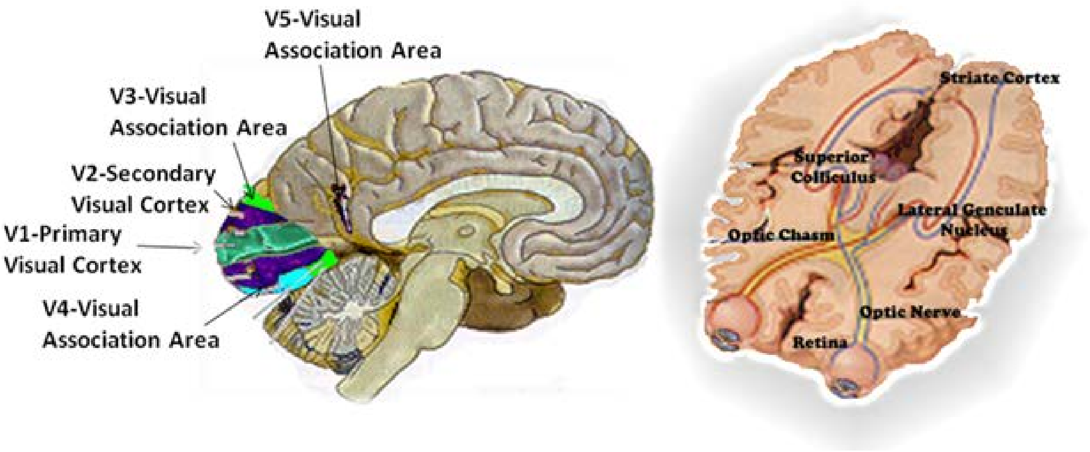

V5-Visual Association Area
V3-Visual Association Area
V2-Secondary Visual Cortex
V1-Primary Visual Cortex
V4-Visual Association Area
Striate Cortex
Superior Colliculus
Lateral Geniculate Nucleus
Optic Chasm
Optic Nerve
Retina

Fig. 2.The Human Visual System and its Associated Areas and Cortices [18]

Now that we know how the brain gets its inputs, the next question is how does it translate and process the information once inside? Spikes, action potentials, and oscillation rhythms are key representations in translating what humans see into motion. The brain utilizes inputs from the various cortices and acts as a virtual simulator in the brain to construct models of motions [56]. Neurons are the inputs and are the basic structure used to transmit data in the brain as shown by Paugam-Moisy [36] in Fig. 3.

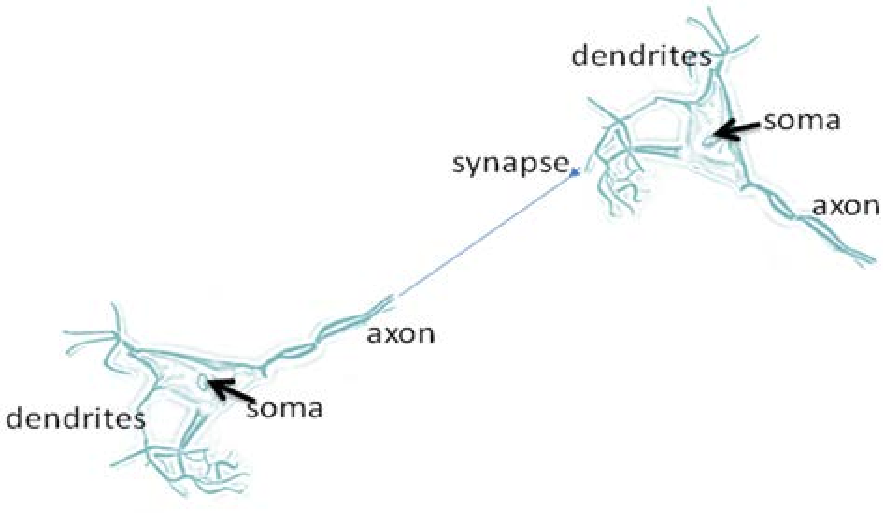


Fig. 3. Neural Structure Showing A Synapse Connection [20]

In review, we examined theories on how humans perform pattern recognition. We investigated the human physiology in terms of learning and function to know how the brain sees information. To create a unique model of pattern recognition, we looked into research on how the brain gets the information (i.e., what are the inputs?), and how the brain translates and processes the information so that our model is human inspired and can recognize patterns using invariant features that are naturally present in the world.

2.3 Computer Based Object Recognition

To create a model of pattern recognition, it is essential to first examine theories on how computers are used to perform pattern recognition. To do that the current models and algorithms must be investigated in terms of learning and function. We need to know how computers see information. To know this we need to know the answers to the following two questions: how computers get the information (i.e., what are the inputs?), and how algorithms translate and process the information.

The answer to the first question is fairly straight forward as shown in Fig. 4. Computers get inputs from cameras either through image scenes or video processing and translating the information into pixels that can be read and extracted by algorithms.

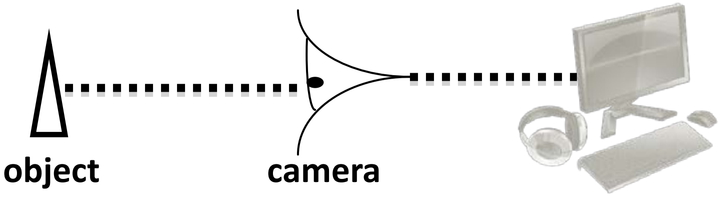


Fig. 4. Typical Computer Vision Setup with an Object, Camera, and Computer

One of the most notable algorithms is the Canny edge detector. Canny [9] created an algorithm that takes objects input via a camera and detects boundaries and edges of objects to image process scene information terms of lines and curves. Canny's success showed that information related to specific objects can be taken and segmented into useful information. The information can be used to show where natural boundaries occur. The information can be translated into data a computer can recognize. The information can be used to index patterns, but nothing is done about determining whether the information is invariant. Riesenhuber and Poggio [39] introduced a nonlinear maximum operation that was invariant to position and scale on the one hand but feature specificity must be built up through separate mechanisms to increase feature extraction. Serre, Wolf, and Poggio [45] followed up by applying Gabor filters to introduce a novel set of features for robust feature extraction. They attempted to apply the visual cortex model of learning where most processing takes place in the V1 region using simple cells and extend it to the V2 region. The main feature extraction technique used by them was through edge detectors [45].

One of the most notable approaches in extracting image information was introduced by Lowe [26]. Its primary purpose was image matching to gain information about structure during scene recognition. Lowe [26] used local reference frames to show that in relation to them scale and rotation is invariant from pixel to pixel. Algorithms can be written to take advantage of the large number of features that densely fill the image scene based on Lowe's [26] four stages of image matching: scale-space maxima detection, keypoint localization, orientation, and keypoint descriptors.

The second question is at the heart of computer based pattern recognition. Most of the research follows the following form. Create a vision system, use objects or characters as targets, and create a large global database of information that takes new information into the algorithm and computes a best guess to the closest match between the input data and the stored database information.

Guy Wallis and Edmond Rolls [51] investigated the use of networks that solved invariant problems related to visual recognition using the trace rule. The trace rule looks at the original cell or neuron information input into the system and tracks its transformation (translation, size, and view) matrix used to discern one object from another in the natural environment via a global database of input scenes. Research conducted by Rowley changed the object to a human face. Rowley [43] used a window of pixel information to extract frontal views of faces to recognize different faces and then transform them based on invariant attributes of a globally trained database to detect the presence of facial features via a neural network. Kanade [53], followed up in 2001 developing action units based on facial expressions. Action units are used in processing information so that facial object recognition occurs at a salient level by extracting facial features. The combination of the two, (1) information processing via neural networks to store and (2) retrieval with the ability generalize facial features based on action units, allows for better facial recognition. The result the ability to visualize human expression and emotion programmed into a global database but input sensory data that are not necessarily invariant.

Viola and Jones [50] contributed to information processing with respect to object recognition at the same time by establishing classifiers which contained information in the form of rectangular areas. Using this format allowed for images to integrate easily and resulted in fast information processing. The effect was that an image could be contained or be segmented into features. Having a vast number of features created the ability to rapidly and easily compute and classify facial expressions subtle and small in size with a 3% improvement in the number of false classifiers rejected. Jonnalagadda, et al., [22] developed a new shape based viewpoint selection process that uses the local geometric features of an object. The local features of the object are assembled into simple geometric primitives and these primitives are then classified into shapes, which are used to hypothesize the global shape of the object.

Around the same time that Lowe came out with SIFT, Fergus [18] utilized shape, appearance, and scale to create an entropy based detector capable of object recognition. The difference between Fergus and Lowe is in the fundamental engineering. Lowe created a scale invariant feature transform algorithm to feature match between scale and orientation between corresponding images of the same scene using Gaussian processing to produce scale space images. He subtracted the Gaussians from each image and what was left were the invariant scale and orientation features. Fergus' approach was Bayesian based and presented more of an engineering or computational view of object recognition using shape and scale as primary data. Fergus [18] approached recognition by detecting features and then calculating a likelihood ratio and compares it to a threshold that indicates presence or absence of an object. The major contributions of both Lowe and Fergus are in the invariant nature of their methods. Chikkerur‘s [13] approach enabled identification and the position of objects in visual scenes using a Bayesian framework. Similar to Fergus, Chikkerur [13] successfully developed a spatially based algorithm of attention to reduce uncertainty in shape information and feature-based attention to reduce the uncertainty in spatial information to recognize objects using a global database.

In review, we examined theories on how computers are used to perform pattern recognition by investigating current models and algorithms. We now know how computers see information by investigating how computers get the information (i.e., what are the inputs?), and how algorithms translate and process the information to recognize objects.

2.4 Scale Space Algorithms

One of the earliest works in the field was by Rosenfeld and Thurston [42]. They developed a routine that worked to detect edges via the use of a differential operator. They used grouping of regions to create a scale space of a scene so that identified edges and curves could be mathematically classified [42]. Witkin [52] expanded the idea of perceptual grouping into one of the earliest scale-space filtering methods that used Gaussian convolutions to compute signal information from its extrema values. This created the ability to examine scale information from large groupings of signal data sets [52]. A visual scheme to detect edges and curves was proposed by Perona and Malik [38] to include anisotropic diffusion in the scale-space for curve and edge detection.

One of the most notable technologies in extracting image information was introduced by David Lowe [26, 27] and is known as the Scale Invariant Feature Transform (SIFT). Its primary purpose was image matching to gain information from image frames to use in pattern recognition. Next, Zhang [54] used human subjects to show how the fovea detects "hot spots." He used "hot spots" to track the eye movement during scene recognition. With this information he created a simulated eye movement model based on invariant features from SIFT. Algorithms [26, 27, 48, and 49] have been written to take advantage of the large number of features that densely fill the image scene based on Lowe's four stages of image matching: scale-space maxima detection, keypoint localization, orientation, and keypoint descriptors.

Scale-space algorithms [6, 23] that make use of the basic SIFT structure include Speeded-Up Robust Features (SURF) and Maximally Stable Extremal Region (MSER) to extract information using vision systems by taking the integral images that result in a time reduction to compute SIFT features. Bay, et al. [6] improved SIFT by using a fast approximation of the Hessian matrix in integral imaging and reducing the artifacts normally associated with Gaussian discretization. Grabner, et al. [20] improved the computational efficiency of pattern recognition in terms of faster processing and provided a framework that can be used to recover feature information using parametric analysis. Cheung [12] successfully demonstrated SIFT pattern recognition by matching medical image scans against a trained database. Montesano and Zhang [33, 54] both use the scale-space approach via a given database trained with keypoint and descriptor information of an object to group features based on similar attributes into categories that can be used for pattern recognition.

Veldaldi and Fulkerson [49] created a SIFT based algorithm that could be used in MATLAB that examined other parameters used to control feature matching through a series of thresholds controlling flat areas, edges, and corners. Maestas, et al. [29] utilized a histogram approach that provides insight into obvious mismatching errors. Keypoints that are different in comparative images, but matched using SIFT, has an angle between the two corresponding points that deviates from the rest of the angles between matches. In the histogram, these mismatches appear as outliers and prime candidates for elimination during feature detection. It is expected that in a perfect match situation at nominal scale without rotation, the set of matched feature points from one image to the other form parallel lines.

2.5 Scale Invariant Feature Transform (SIFT)

To use the SIFT algorithm, one chooses a number of specific parameters. The parameters are octaves, levels, and thresholds. Octaves include information about the number of times that the Difference of Gaussians (DOG) is performed and where to start taking the DOG, known as the first octave. Levels refer to the number of levels per octave within the DOG subspace. Thresholds are divided among peak, edge, and normal. The thresholds establish which image points to accept between edge and non-edge and provide controllability of image pixel data. Lowe used local reference frames to show that scale and rotation do not change from pixel to pixel [26, 27]. This property is referred to as invariance in the local frame. One of the primary reasons SIFT works is because of the large number of keypoints it generates from an image [26]. The density of keypoints allows for frames to be created in clusters. Small objects in a scene background can be recognized as long as three features are correctly matched, either from two images or from a training database via Euclidean distance [26].

SIFT develops a scale space by looking for extrema points and then extracts the point's position, orientation and scale. It is the result of four major steps [26, 27]: (1) create an invariant scale space, (2 and 3) find and manipulate the keypoints, and (4) bin and pair the keypoints.

2.5.1 Create an invariant scale space

The initial step is to create representations of the original image. To accomplish this, Gaussian blur is applied to the original image. Creating the scale space in this manner ensures scale invariance. The parameters that control the initial scale space representation are the number of octaves used and the number of levels required. The number of octaves implemented refers to how many sequential times the original is reduced by half the previous octave, Lowe used a default of four. The number of levels implemented refers to how many blurs including the original are completed for within each octave, Lowe by default used five.

Blurring involves convolution using a Gaussian:

$$B(x,y,\sigma)=G(x,y,\sigma)*I(x,y) \qquad (2.1)$$

where B is the blurred image, G is the Gaussian operator, I is the original image, x,y are the location coordinates, and σ the scale parameter.

Convolution involves the Gaussian operation equation:

$$G(x,y,\sigma)=\frac{1}{2\pi\sigma^2}e^{-(x^2+y^2)/2\sigma^2} \qquad (2.2)$$

2.5.2 Find and Manipulate Keypoints

The Difference of Gaussians (DOG) is used to find unique keypoints from images that are scale invariant. Within an octave, starting from the original image, the next blurred image is subtracted from the previous image. The result is *n-1* images, where *n* is the number of levels parameter and contains invariant data left after subtraction. This is accomplished by scaling the images in the Gaussian operation equation above by σ^2 function which means the Laplacian ∇^{2}*G becomes σ^{2}*∇^{2}*G and scale invariant when keypoints are created. Keypoints are determined by looking for extreme values both high and low by comparing one pixel to its neighboring pixels one level above, at the same level, and one level below. There are 27 values to compare, e.g., one pixel of interest compared to the values of 26 nearby pixels. If it is a minimum or a maximum it is kept as a keypoint. The process is repeated for all levels except the top and bottom levels. The entire set of images where the DOG was applied is an octave. New octaves are then created by scaling them down to create another scale space by an integer value. Typically a factor of 2 is used so that the entire octave space is covered and then the DOG is computed.

Once keypoints are generated, subpixel information needs to be generated. The purpose of the subpixel generation is due to the fact that max and min values can occur between pixels and can lead to mismatches when the keypoints are manipulated. Taylor series approximations are set equal to zero to solve for keypoints, $\hat{\mathrm{x}}$. The equation to find keypoints is as follows [26, 27]:

$$\hat{\mathrm{x}} = -\frac{\partial D^{-1}}{\partial x^2}\frac{\partial D}{\partial x} = 0 \tag{2.3}$$

Filters can be applied to the magnitudes of min and max values to eliminate low contrast areas.

2.5.3 Keypoint Binning and Pairing

The values of the gradient magnitude and orientation are computed for each keypoint. The equations used are:

$$\nabla \mathrm{mag}\,(\mathrm{x}, \mathrm{y}) = \sqrt{(\mathrm{L}(\mathrm{x}+1, \mathrm{y}) - \mathrm{L}(\mathrm{x}-1, \mathrm{y}))^2 + (\mathrm{L}(\mathrm{x}, \mathrm{y}+1) - \mathrm{L}(\mathrm{x}, \mathrm{y}-1))^2} \tag{2.4}$$

and

$$\angle_{x,y} = \tan^{-1}((\mathrm{L}(\mathrm{x}, \mathrm{y}+1) - \mathrm{L}(\mathrm{x}, \mathrm{y}-1))/(\mathrm{L}(\mathrm{x}+1, \mathrm{y}) - \mathrm{L}(\mathrm{x}-1, \mathrm{y}))) \tag{2.5}$$

where L is the smoothed Gaussian image at the closest scale to the keypoint. A histogram of the orientations can be made for keypoint locations which align keypoints from one image to another. The most highly counted orientation is then attached to that keypoint location to pair up the same keypoints in other images. Points are paired up and matched to their closest location from a different image using a Euclidean distance metric via histogramming angles between image points. The resulting binned information is used to pair points from images to create matched features between different rotations and scales.

2.6 Taguchi Design of Experiments (DOE)

Design of Experiments (DOE) is an approach that computes variation in a process to minimize the number of experiments required to achieve an optimal set of parameters. Traditionally, DOE has been used to determine the optimal setting for machine parameter values. In our research, DOE is used to determine the optimal values for the parameters of the SIFT algorithm.

The Taguchi method, a particular type of DOE algorithm is a way to plan, conduct, analyze, and determine optimal settings for a system using orthogonal arrays. The orthogonal array is the method by which relatively few experiments span a large experiment space [44, 47] versus that of a full factorial set of experiments.

Since the Taguchi DOE method performs relatively few experiments, it is unlikely for the optimal answer to be one of the prescribed experiments.

However, it is possible to use the experiments to determine the optimal settings as well as to predict the output expected from the use of those settings. Fig. 5 shows the delta between Taguchi over Full Factorial. The additional number of full factorial experiments that would need to be run corresponds to the Lx matrix from Taguchi, where x is the number of factors being tested. For example, the 4 on the x-axis represents an L4 Taguchi matrix and would require the same number of experiments using a full factorial design. Therefore no additional experimental benefit (or effectiveness=0) is gained by using Taguchi over a Full Factorial design for a two level design. As a result more experimental levels can be introduced via Taguchi design to increase the probability of finding the best answers for a given possible set of variable combination of parameters, e.g., a mixed 2 and 3-level design. Using the best set of answers is then used to find the optimal settings to achieve the best system performance.

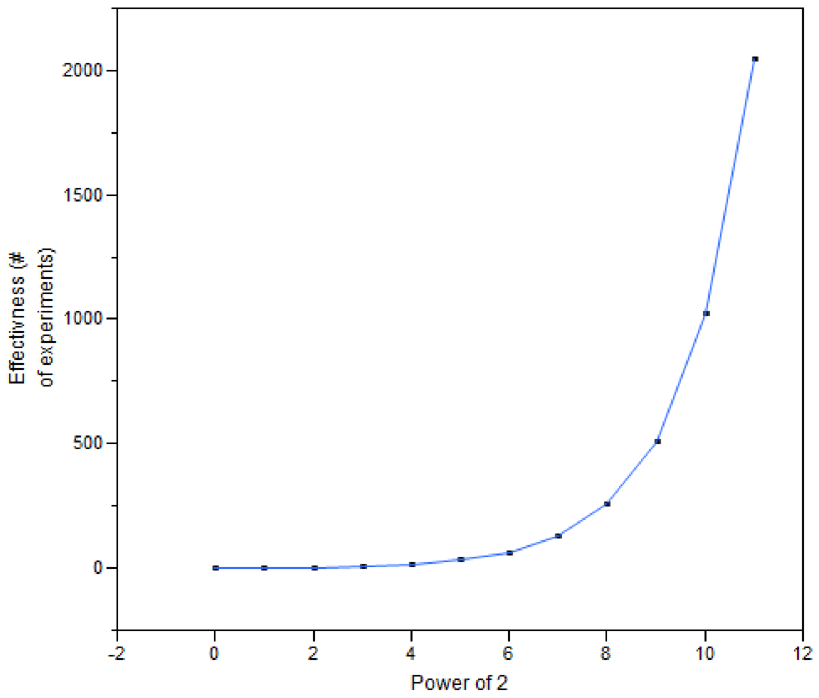


Fig. 5. Additional number of full factorial experiments that would need to be run versus the Taguchi Design

2.7 Neural Networks

David Marr was an early examiner of the brain. Marr's [30] theory was that the cerebellum learns to perform motor skills via learning movements or actions and learns to maintain posture or balance in reaction to observed actions. This first approach did not answer the question of how humans see, but developed the foundation for further work for following investigations. Marr [30] presented to the Royal Society of London that the link between the cerebellum and cerebrum is established by using criteria (form, color, texture, and movement). The nervous system uses these criteria and splits up its visual input into components or classification units based on object features. Object features can be used in visual bonding (find common features in nature) and coding based on continuities in capturing invariant information to look for patterns of coherent matter (objects) [30]. James Albus [3] quickly adapted Marr's work to create the theory of cerebellar function used in robotic arms by mimicking the function of the biological cerebellum to control and coordinate movements. Duda and Hart [15] published their own neural network around the same time that was capable of classifying features with a scene.

Another early researcher was Frank Rosenblatt. Rosenblatt's [40] research that is most applicable is his invention of the perceptron. Using a perceptron a neural network can be implemented that is able to classify objects and resulted in Rosenblatt's network able to identify objects. Peeling and Moore [37] used the Rosenblatt perceptron to create a Multi-Layer Perceptron (MLP) for isolated digit recognition. Though primarily a Bayesian approach, Peeling and Moore's [37] work was a precursor to adaptive non-parametric networks [25].

One of the most well-known non-parametric networks is the nearest neighbor method. The result of the nearest neighbor method is the ability to easily classify. Friedman [9] extended the nearest neighbor method by creating a metric by which inputs are graded based on expectation. The expectation is that inputs that are more important have a higher grade than lower valued inputs. This ability to statistically partition inputs is the precursor to weighting inputs in present day statistically pattern recognition networks [19, 21]. Cortes and Vapnik [14] created a support vector machine capable of classifying digits. Their statistical approach is a linear discriminant function that uses a perceptron based neural network for two group classifications [14]. Bell and Sejknowski [7] created a self-associative network call individual component analysis that is adapted from the results one normally attains from using principal components analysis. Bell and Sjenowski determined that edges are detected by linear filters alone and are a top level invariant feature in natural images. They suggest that other levels such as scale, rotation, and illumination can be second level invariant features capable of detection using non-linear statistical pattern recognition techniques [7].

Several researchers (Rowley [43], Kanade [53], Viola, and Jones [50]) have used non-linear discriminant functions couple with a multi-layer perceptron neural network for classification. Research conducted by Rowley [43] changed the object to a human face. Rowley used a window of pixel information to extract frontal views of faces to recognize different faces within and then transform them based on invariant attributes of a globally trained database to detect the presence of facial features via a neural network. Kanade [53], followed up by advancing the state of the art of information processing by developing action units based on facial expressions. Action units are features used in processing information so that facial object recognition occurs at a salient level by extracting only facial features. The combination of the two, information processing via neural networks to store and retrieve with the ability generalize facial features based on action units, allows for better recognition in terms of being able to visualize human expression and emotion programmed into a global database. Viola and Jones [50] contributed to information processing with respect to object recognition at the same time by establishing classifiers which contained information in the form of rectangular areas. Using this format allowed for images to integrate easily and resulted in fast information processing. The effect was that an image could be contained or be segmented into a variety features. Having a vast number of features created the ability to rapidly and easily compute and classify facial expressions subtle and small in size with a 3% improvement in the number of false classifiers rejected.

Park, et al., [35] created a Parzen classifier neural network that utilizes a polynomial radial basis function to create classifiers via fuzzy clustering. Using fuzzy clustering, if-then rules are created to separate and classify polynomials [35]. The advantage of such a radial based function network is a rapid convergence to a global optimal approximation-basically fast learning [35]. Most recent Kim, et al., [24] used a hybrid approach by combining self- organizing maps, introduced by Kohonen, with that of learning vector quantization theory, and nearest neighbor networks theory that designates a classifier as a prototype. Kim proved that classifier near boundaries play a more important role than classifiers in the interior of a boundary. The result is more accurate global classifiers, but increased error during recognition.

2.8 Cerebellar Model Articulation Controller (CMAC)

In 1969 David Marr published the theory of cerebellar cortex and two years later, in 1971, James Albus expanded this state of the art by introducing his theory of cerebellar function. Albus [3] utilized Marr's theory and applied it to robotic arm control by mimicking the function of the biological cerebellum to control and coordinate movements. Albus' theory was put into practice over the next four years to develop a Cerebellar Model Articulation Controller (CMAC) [1]. Albus was trying to solve the problem of knowing and coordinating joint dynamics at every instant in time based on the hierarchical dependencies required to perform a task. To achieve robot manipulation, a neural network connected the function of the cerebellum (motion and coordination) to the perception (what is to be manipulated) based on patterns in nature. The neural network created binary inputs and outputs weighted by pairs of mappings [1]. The mappings started with sensory cells, then input to association cells, adjusted for weights, and output to response cells based on the weighting results. The reason this worked was because of the classical nature of using binary stimuli (1 or 0). The result is that there would be overlaps in the information taken from a scene based on similar information being processed as a digital signal as shown in Fig. 6.

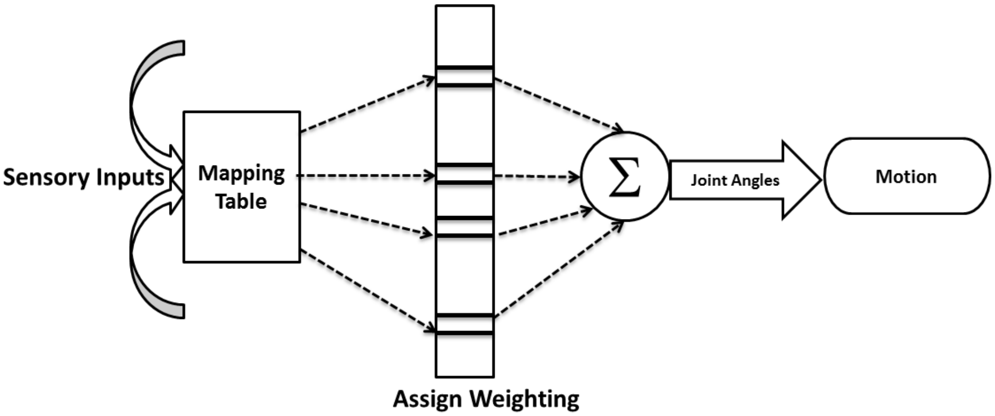


Fig. 6. A Typical CMAC Neural Network with Sensory Inputs and Output Motion

The data are distributed by summing ones and zeros in each space location [2]. An advantage to the early work in CMAC is that all levels are treated the same way, given either a 1 or a 0, making it easy for conventional linear mapping—though never demonstrated explicitly [1,2].

In the early part of the 1990s, Albus came out with his "Outline for the Theory of Intelligence." Albus [4] quantified "Intelligence" as a measure of value that can be used to describe real world objects in terms of computer programmable inputs that could be used as the basis for object recognition. The more a system can recognize correctly the more intelligence the system possesses.

In 2001, Albus [4] expanded this research on his "Theory of Intelligence" into processes of the mind that linked internal models of the brain to world models in quantifiable terms. Albus [4, 5] consolidated 20 years of research into "An Introduction to the Science of Intelligent Systems." This was the first attempt by anyone to quantify the processes of the mind into computational terms that could be used to create a basic structure for perception as shown in Fig. 7. The basic instruction defines perception as the correspondence between the internal world model and external real world and how humans behave as a result [4, 5]. Next, human behavior utilizes the world model to generate action to achieve goals like that of object recognition [4, 5].

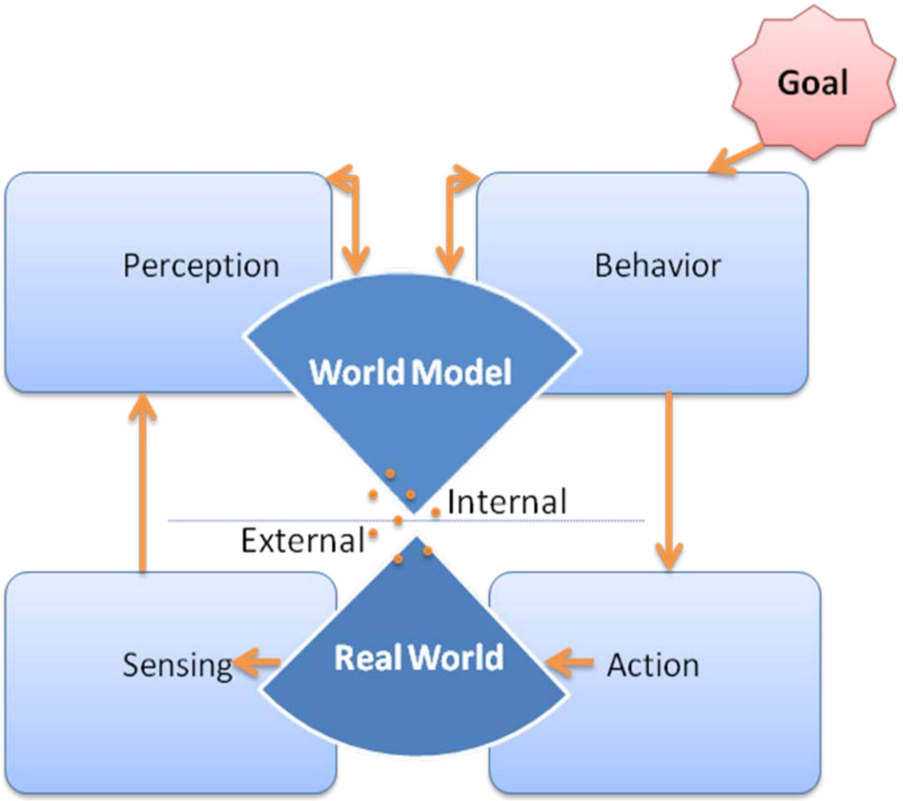


Fig. 7. A Basic Structure of Human Vision Known as the Perception Model

Albus related sensory processing to the regions of the visual cortex. For example, in the V1 region, input from the optic nerves are overlaid, since there are two, one from each eyeball. Then the information is registered and sent to V2. In V2, similarities and differences are computed and compared to the world model and converted into basic structures like points, lines, curves, edges, and boundaries updating the internal model. V3 and V4 detect motion and surface textures and create segmented groups of information so that it can be used for recall and comparisons. The V5 region contains basic information and is used for storage. Basically, Albus used his “Outline for the Theory of Intelligence” paper to setup a research direction related to vision in terms of relating biological concepts about the world into quantifiable inputs used to test hypotheses between computational visual systems to those of other mammalian visual systems. Value state representations created the quantization of human emotional states (pain, confidence, happiness, uncertainty) into state variables which could be programmed using computers.

The Albus theory consists of both top down processing converging with bottom up information. The theory is that top down processing emanates from within the brain and recognizes objects by either classifying them or recalling information about the object from memory. CMAC can then be adapted as a memory storage neural network. Bottom up processing relies on the eye's optic nerve and fovea to gather information and transmit it to the visual cortex. The visual cortex locates and stores the information in one of the regions. Albus’ work on CMAC introduced a new capability of using neural networks for a plethora of tasks in a human inspired way. He leveraged Marr’s initial theory to create a basic structure. Later in the 20th century Albus used it to measure intelligence and quantify it in a way that is suitable for computers to utilize in real world tasks to create perception models. Perception models relate what goes on inside the human brain to a model that is programmable by humans using computers to solve real work tasks.

2.9 Conclusion from Previous Work

An opportunity exists to take advantage of linear invariant information contained in edges and boundaries and associate them with second level non-linear information from high quality keypoints described in previous sections that are invariant to scale, rotation, and illumination. This information can be used as any input to create a network capable of local classification which results in a human inspired local model of pattern recognition vs. the global models currently developed that are inflexible and non-human in the processing of information due to input sensory data that is not necessarily invariant (background data, raw pixel data, viewpoint angles).

Chapter 3: LOCALLY BASED PATTERN RECOGNITION

3.1 The Model

The model proposed creates a pattern recognition scheme that leverages the most current research to create a network that locally trains, classifies, and recognizes patterns based on local invariant keypoints extracted from image scenes. The model is trained via the implementation of an algorithm based on keypoints. The keypoints are utilized as inputs into a neural network similar to those used in a Cerebellar Model Articulation Controller (CMAC) neural network and in the human visual cortex. The parameters (octaves, levels, and illumination) were optimized with respect to accuracy to extract invariant keypoints. The keypoints are localized meaning based on input image data only. Using the localized invariant points, they are used as inputs that are used in a CMAC-inspired neural network to establish a model of pattern recognition. As shown in Fig. 8, the final system resembles a CMAC neural network and is comprised of inputs, a hidden layer, and outputs. It works by taking inputs, and using feature extraction, detects edges and boundaries to create an image where invariant keypoints are identified. Based on the key point information, objects are classified and fed into the network to perform pattern recognition.

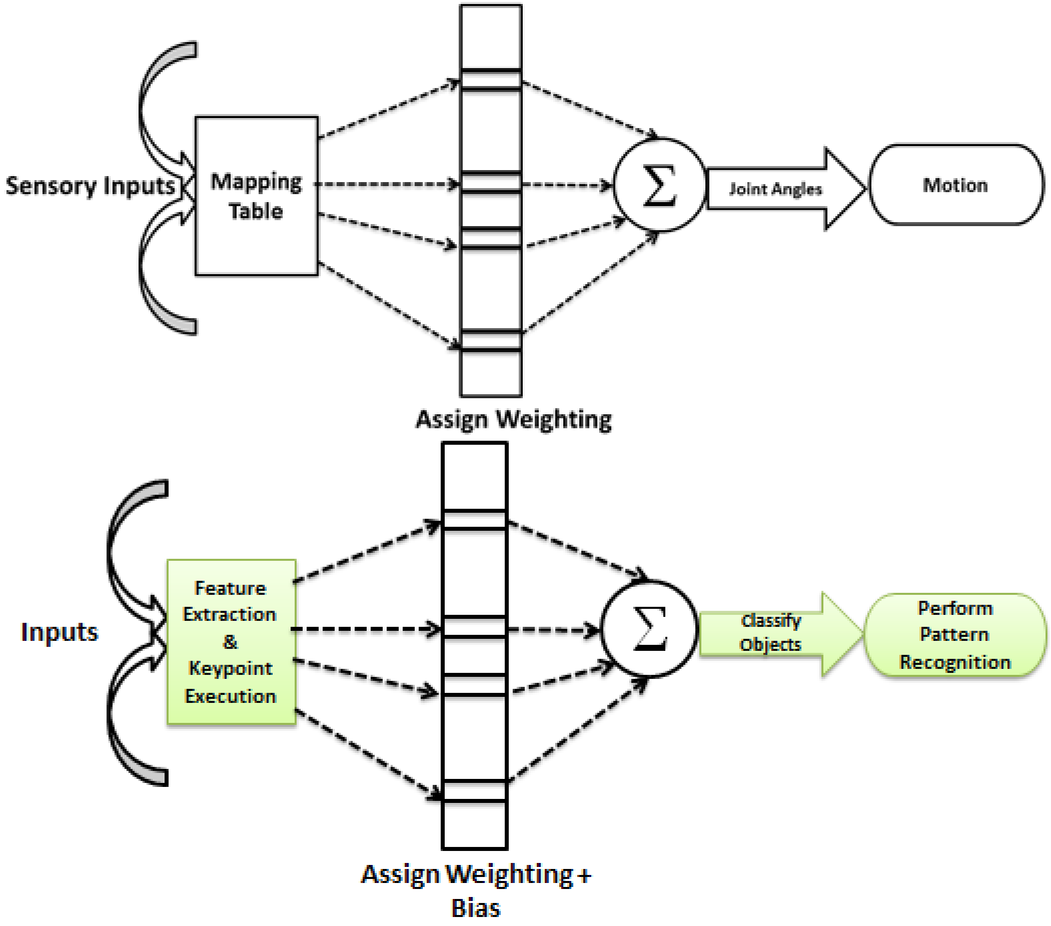


Fig. 8. Model of Invariant Local Pattern Recognition (lower) vs. CMAC (upper)

The above model of invariant local pattern recognition can then be adapted by using Rosenblatt's perceptron to be extended to use the weights and bias which can be trained to produce the correct pattern recognition for a given set of input objects. The translated system level diagram is shown in Fig. 9. Each object datum is assigned a weight and then sent through a transfer function along with a bias resulting in an output that is either one or zero. Next, the network can be trained so that the desired output coincides with the input objects. Simulink training presents a subset of the object data to network so that the weight and bias are automatically adjusted until the weight and bias does not change with additional input of object data.

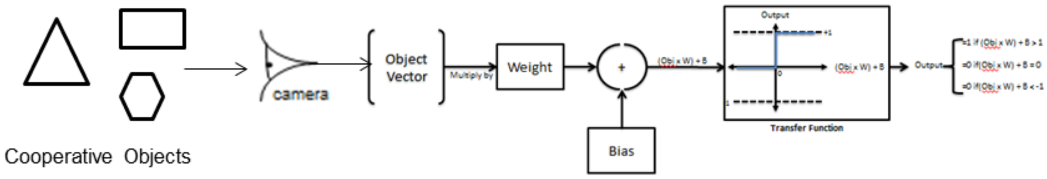


Fig. 9: Simulink System Flow Diagram of Classifying Objects

3.2 The Contribution

The proposed research develops a unique pattern recognition model that consists of a cerebral framework that locally trains, classifies, and recognizes patterns based on local invariant keypoints extracted from image scenes. The term “local applies” to data stored in the neural network without the need for a large database of object attributes. The gap revealed by the literature review clearly shows that while a variety of global models and information are used in pattern recognition by computers, the ability to recognize using only local information is minimal to none. The ability to model more like humans is ingrained local pattern recognition so that edges and invariant points in nature can used to teach computers to see. The proposed model provides the ability to establish pattern recognition within a scene based on local information and training as part of the process without the need for a large external database. This is the first time the pattern recognition occurs on a local level using only invariant information regardless of scale (near or far) or rotation.

3.3 The Impact

The proposed model links the current global models of pattern recognition to that of human based pattern recognition by being able to process scene information relying solely on localized image data extraction-a departure from current algorithms. Currently algorithms are task specific and work well if a target object and all its attribute variations are already trained, but struggle to correctly recognize new objects if they are not part of an externally trained database because of the noise that lack of invariance introduces. The proposed model establishes pattern recognition within a scene based on local invariant information that works with any number of objects—a cluttered scene. Training a system to identify multiple patterns of interest provides a robust pattern recognition scheme. Neural Networks in robotic vision applications are emerging as new field. Neural Network robotics integrates the fields of mechanical engineering, robotics research, and computer vision that is human inspired and serves as a computational brain with the ability to train, classify, and recognize objects. Applications of algorithms thought only to be reserved for the human brain can be attempted by using this new method of local pattern recognition based solely on invariant inputs.

Chapter 4: IMPROVING KEYPOINT FEATURE QUALITY

4.1 Introduction

While the SIFT algorithm is widely acknowledged to be one of the best available image processing algorithms for feature extraction over a wide range of scales and rotation, this chapter explores whether its performance can be improved by choosing values for its parameters that are different than the current ones. We explore choosing SIFT parameters through an optimization process, specifically the Taguchi design of experiments. The hypothesis is that choosing the key parameters, e.g., octaves levels, the number of times the Difference of Gaussians is completed, and various thresholds using the Taguchi design of experiments results in improved feature extraction. It is certainly possible to choose specific values for each SIFT algorithm parameter and run experiments for each possible combination of variables. This brute force approach is both time consuming and unnecessary. Instead, a Taguchi Design of Experiments (DOE) is used to choose the values of the parameters that improve SIFT feature extraction compared to the original SIFT algorithm's parameter values.

4.2 Theory

Once we have completed the basic SIFT steps by creating an invariant scale space, finding and manipulating keypoints, and binning and pairing keypoints, we can improve the feature quality by creating a metric for feature detection.

4.2.1 Feature detection

Metric Formulation: The Taguchi method requires a performance index that quantitatively measures the results of experiments. The optimization occurs versus SIFT parameters: octaves, levels, and illumination. However, in terms of matching there is gap in determining how to measure the quality of matches.

To determine whether a match is considered good or bad, a scoring method needs to be applied to rank matches to differentiate between good and bad matches—also known as rank filtering. Normalized cross correlations (NCC) are used to score the quality of points. NCC is invariant to brightness changes and compares the same regions between images. The result is a ranking of image attributes such as keypoints. The highest quality of points are being filtered and used as correct matches. The NNC equation is given as:

$$NCC = \frac{\sum_{i=1}^{N} \alpha_i \cdot \beta_i}{\sqrt{\sum_{i=1}^{N} \alpha_i^{\ 2} \cdot \sum_{i=1}^{N} \beta_i^{\ 2}}} \qquad (4.1)$$

where α and β are the resulting sums from two equally sized regions for two images under process, i is from the first region segment to N, the last region block being compared between two images. NCC ranges between -1 and 1, where a score of 1 means two identical regions.

Once the scoring is in place we can filter on randomly grouped NCC point scores using a randomly sample and consensus (RANSAC). First a random sample of keypoint pairs is grouped together. The group NCC is averaged. This process is repeated until all permutations are scored. The highest scoring group is then used to filter on and the rest of the point pairs are filtered out and this group becomes the correct matches.

The total of number of matched points can be compared prior to the rank scoring and filtering to determine the percentage of matched pairs that were correct matches. The percentage correct is calculated using the correct matches divided by the total number of matches via the following equation.

$$\%_{correct} = \frac{\text{\# of Correct Matches}}{\text{Total\# of Matches}} \qquad (4.2)$$

It is a good measure because matching points between similar images are illumination invariant by scoring based on the normalized cross correlation. Secondly, the quality of resultant image matching is higher than default SIFT parameters so that the effect of each parameter (octaves, levels, and illumination) is orthogonal with respect to other parameters.

Fig. 10 shows the effect of utilizing a rank filter. First, traditional matching of keypoints can result in mismatched points as shown in red and has a lower NCC value than matched points in green. Because they would typically be included in final matched results, the mismatches result in poorer feature detection. By applying a rank filter, we can quantify using our measure of goodness. The rank filter value of the feature detection match under various parameter changes is computed for percent correct before and after keypoint pair elimination from the results of RANSAC. In the Fig. 10, for both scale and rotation the percent correct would be 2 correct matches out of 3 total matches--66%.

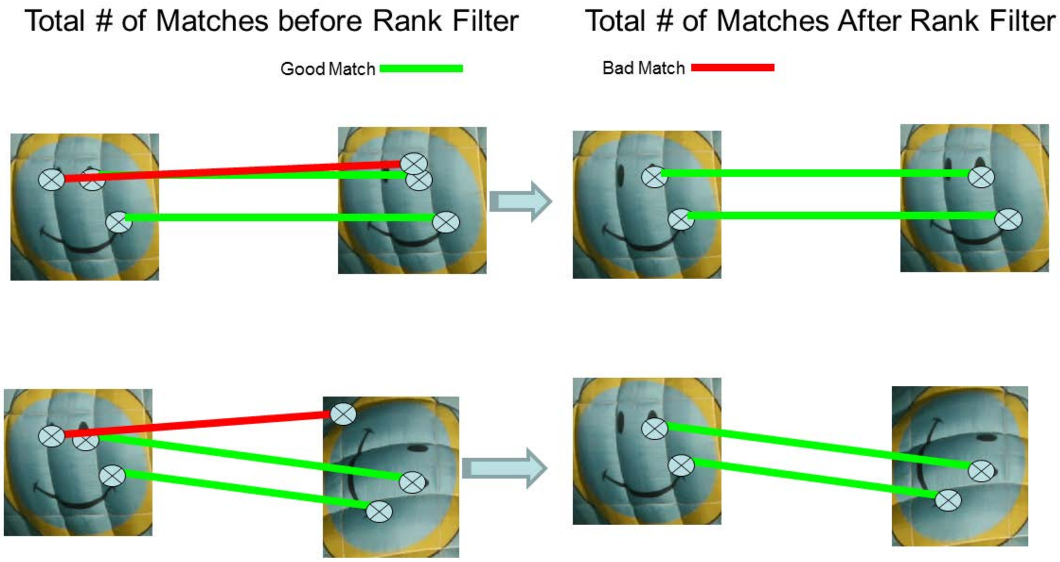

Total # of Matches before Rank Filter
Total # of Matches After Rank Filter
Good Match
Bad Match

Fig. 10. Example of Matching Keypoints on different images before and after Rank Filtered

By choosing the best parameters for the algorithm, the %correct increases--improving which keypoint pairs are utilized during feature detection (i.e., only high quality matches are accepted).

Feature Detection Methodology: The algorithm for feature detection is composed of the following steps.

1. Capture images and compare two images at different rotation, scale, and illumination.
2. Determine the keypoints and rank using NCC between each image.
3. Apply rank filtering to determine the correct matches.
4. Eliminate undesired matches and compute the %correct.

As shown in Fig. 11, the aforementioned steps produced a code so that the Taguchi DOE can be used to test the set of SIFT algorithm parameters that produces the best matching.

```
1.  If{Images Exists}
2.    Read in Images
3.      Compute the scale invariant features
4.        Find and Manipulate Keypoints using DOG from consecutive Frames
5.          Determine the NCC
                Match points within Frames
6.          Randomly Sample and Group the Correct
               Matches
7.      Eliminate False Matches
8. EndIf{Show Images Overlaid}
```

Fig. 11. SIFT Matching Algorithm Code Steps of Matlab Implementation

4.2.2 L18 Experimentation

Setup: The equipment used during the experiments includes 4 Basler A605fc-2 cameras assembled on a vision table with camera mounts on Panavise 15″ goosenecks at each corner. Lenses were 9 mm Fujinon HF9HA-1B. Lighting is controlled using two umbrella light assemblies to ensure diffuse illumination that can vary between 250-650 watts. The vision table cameras are connected via IEEE 1394 fire wires to two PCI-fire wire cards (2 cameras per card) as shown in Fig. 12. The station controller uses a core 2 duo processor from Intel, with 3 gigabytes of onboard Random Access Memory (RAM). Experimental analysis used Matlab with image acquisition from a Canon Powershot camera under different conditions and brightness. JMP 9.0 [56] statistical software is used for analyses of the Taguchi Design, effects rollup, and optimal parametric prediction. The implementation consists of capturing an image pair. Then matched keypoints are detected using SIFT as the method of extraction. Using the NCC equation the keypoint matches are ranked. Next, RANSAC filters out in correct matches. The final metric is computer to determine the number correct matches.

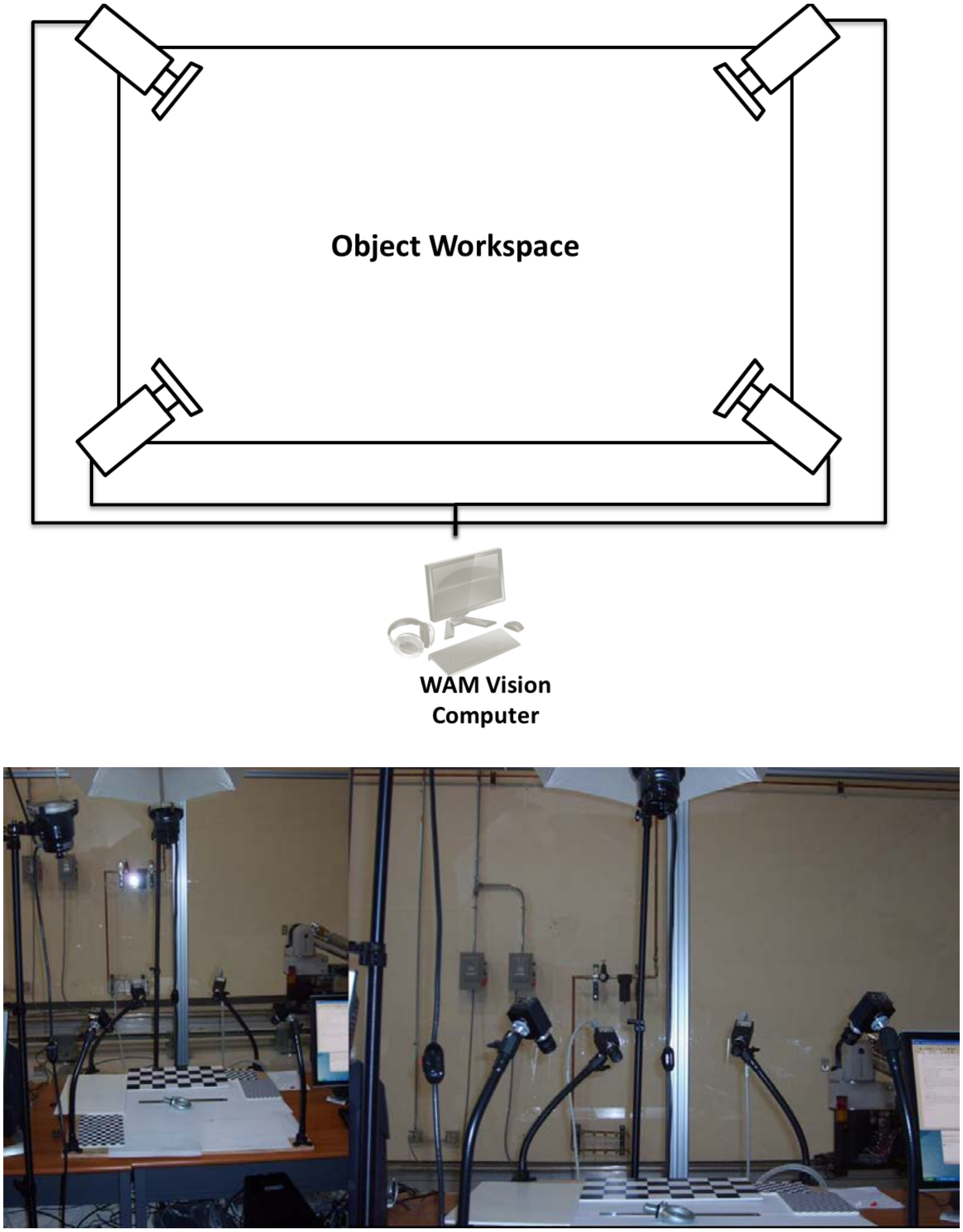


Fig. 12. Multi-Image Experiment Setup Diagram Schematic and Photos of the Vision Table

Parameters: The following list describes the key parameters or factors controlled by SIFT algorithms. A mixed 2-3 level of each factor is used via an L18 Taguchi meaning one factor will have two levels of parameter values to test while the other factors use three levels. Because a rank filter is used, other threshold parameters normally used in SIFT algorithms to control feature detection are no longer of primary concern. Additionally, the number of experiments required by being able to use an L18 design instead of an L32 design is reduced with additional columns being dedicated to investigate the presence of interactions.

1. **Octaves: (low=2 med=4 high=6)** is the number of times that the Difference of Gaussians is performed to create a scale-space.
2. **Levels: (low=3 med= 5 high=8)** determine the number of levels per octave within the Difference of Gaussians subspace.
3. **Illumination: (low=-30% brightness high=+30% brightness)** are the lighting levels associated with each image from umbrella lights or using image contrast control.

4.2.3 Taguchi Setup

The experimental set up consists of 3 factors using a mixed 2-3 level design along with their interactions as shown in Fig. 13. The linear graph is interpreted as follows. To determine the interaction between columns 1 and 2 of Taguchi's L18 coded design use the column 3. To determine the interaction between columns 2 and 5 of Taguchi's L18 coded design use the column 8. To determine the interaction between columns 1 and 5 of Taguchi's L18 coded design use the column 6.

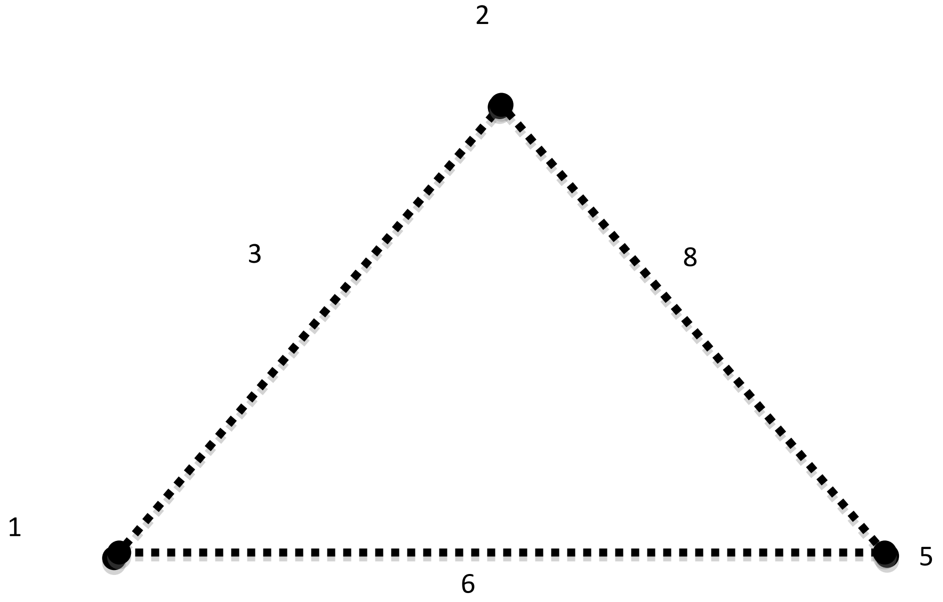
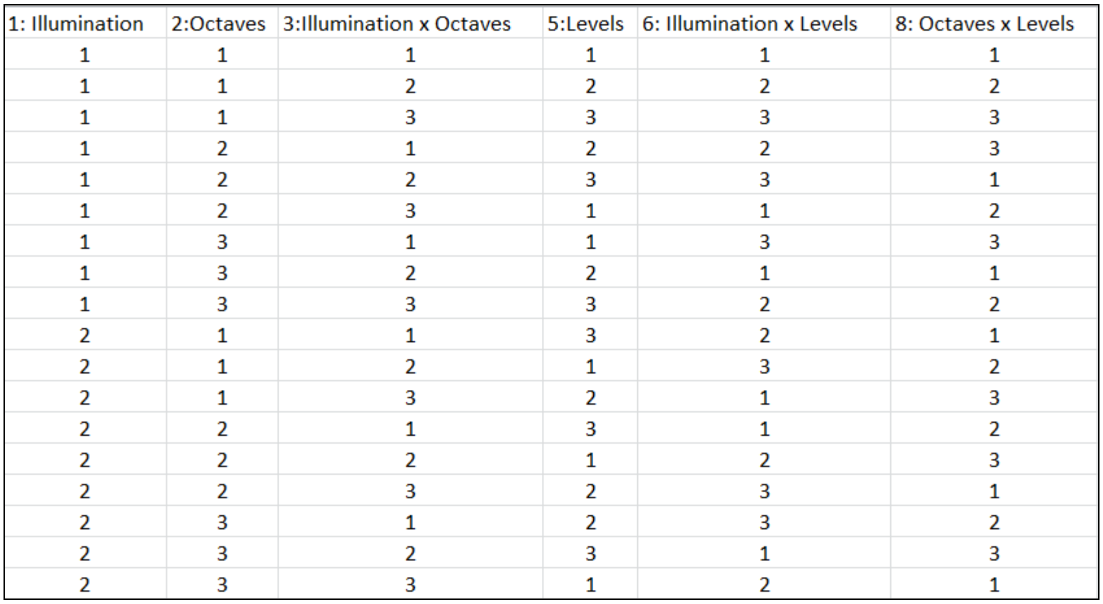


Fig. 13. L18 Linear Graph with Interactions for Mixed 2-3 Design

Using the aforementioned factors (octaves, levels, and illumination) the resultant L18 experimental coded design is shown in Table 1 along with their interactions, denoted by "x."

Table 1. L18 Design Parameters



1: Illumination	2:Octaves	3:Illumination x Octaves	5:Levels	6: Illumination x Levels	8: Octaves x Levels
1	1	1	1	1	1
1	1	2	2	2	2
1	1	3	3	3	3
1	2	1	2	2	3
1	2	2	3	3	1
1	2	3	1	1	2
1	3	1	1	3	3
1	3	2	2	1	1
1	3	3	3	2	2
2	1	1	3	2	1
2	1	2	1	3	2
2	1	3	2	1	3
2	2	1	3	1	2
2	2	2	1	2	3
2	2	3	2	3	1
2	3	1	2	3	2
2	3	2	3	1	3
2	3	3	1	2	1

Since we are using a bigger is better metric for the Taguchi DOE, the goal of the experiments is to determine the set of parameters that creates matches that produce the highest percentage of correct matches (%correct). The %correct is the index of performance that is used to optimize the parameter set. The experimental mean is calculated to compare to the predicted and actual means from Taguchi Design and its validation trial, where x is percent correct for the experiment, n is the number of runs, j is the number of experiments, and i is the experiment number.

$$\%_{\text{Correct-Experimental}} = \frac{\sum_{1}^{j} x_i}{n} \qquad (4.3)$$

The signal to noise ratio for each experiment, i, for a bigger is better experiment is represented by

$$\frac{S}{N}_i = -10\log_{10}\left(\frac{\frac{1}{\%_{\text{correct}}{}^{2}}}{n} \right) \qquad (4.4)$$

Next, the effect of each factor is known and used in the optimal prediction with respect to each of the parameters. The result is predicted to lead to the highest percentage of correct matches and reduces the number of invalid matching between comparative images.

To compute the optimal values, η, Taguchi's Additive Model (no interaction of parameters is assumed) for orthogonal arrays [8] is used:

$$\eta = \mu + \sum_{1}^{j} \left(\overline{x_j} - \mu \right) \qquad (4.5)$$

where μ is the overall mean for all experiments run, $\overline{x_j}$ is the effect mean from 1 to j, the number of experiments.

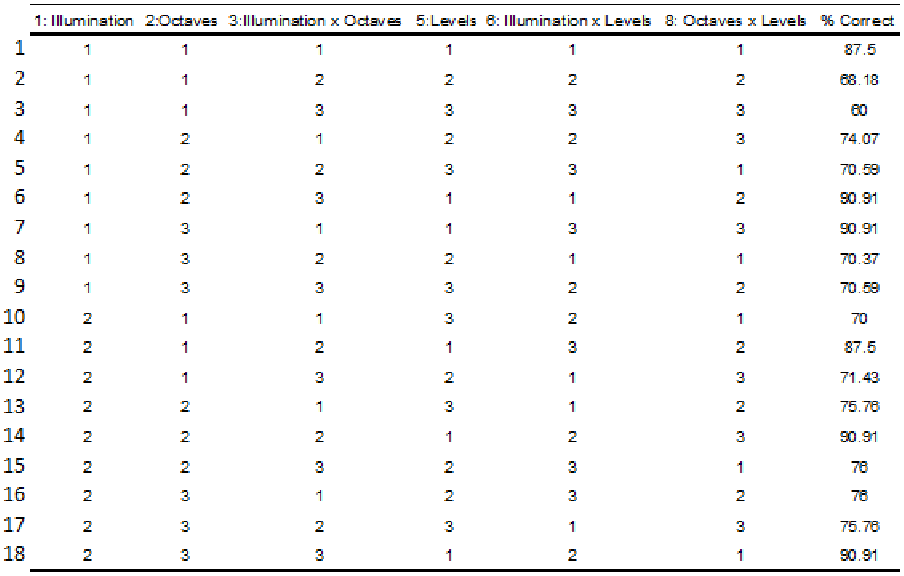
4.3 Experiments

With our theory established and a metric formulated. The experiments are run to determine if keypoint feature quality is improved.

4.3.1 Results

Using a random run order, 36 experiments (18 experiments across 2 runs) were conducted and the results are shown in Table 2 corresponding to the L18 design codes and shown in the table below.

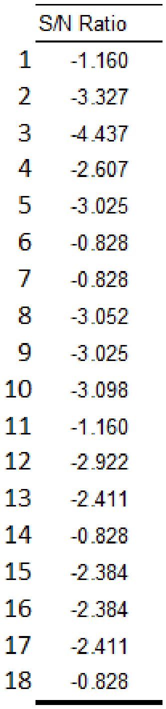
Table 2. L18 Design Results for % Correct



	1: Illumination	2:Octaves	3:Illumination x Octaves	5:Levels	6: Illumination x Levels	8: Octaves x Levels	% Correct
1	1	1	1	1	1	1	87.5
2	1	1	2	2	2	2	68.18
3	1	1	3	3	3	3	60
4	1	2	1	2	2	3	74.07
5	1	2	2	3	3	1	70.59
6	1	2	3	1	1	2	90.91
7	1	3	1	1	3	3	90.91
8	1	3	2	2	1	1	70.37
9	1	3	3	3	2	2	70.59
10	2	1	1	3	2	1	70
11	2	1	2	1	3	2	87.5
12	2	1	3	2	1	3	71.43
13	2	2	1	3	1	2	75.76
14	2	2	2	1	2	3	90.91
15	2	2	3	2	3	1	76
16	2	3	1	2	3	2	76
17	2	3	2	3	1	3	75.76
18	2	3	3	1	2	1	90.91

Using equations (4.3) and (4.4) the mean and signal to noise ratio were calculated for each run to estimate the optimal parameters to use for feature detection. The %Correct-Experimental (i.e., mean of the L18 experiments) for the L18 of experiments is 77.63%. The results for signal-to-noise ratio for each run are computed with equation (4.4) and shown in Table 3.

Table 3. L18 Design Signal-to-Noise Ratios

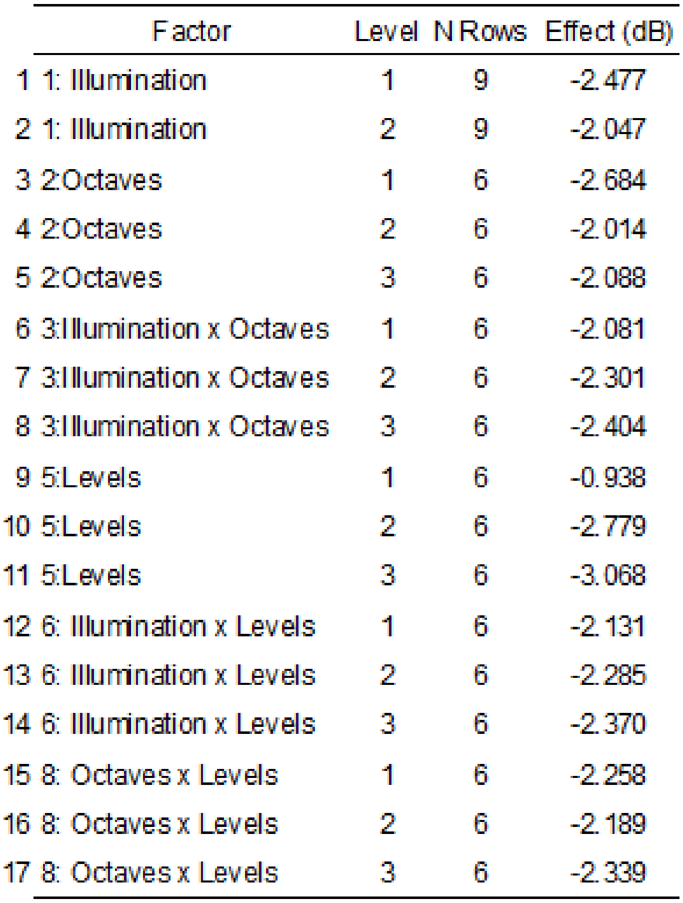


	S/N Ratio
1	-1.160
2	-3.327
3	-4.437
4	-2.607
5	-3.025
6	-0.828
7	-0.828
8	-3.052
9	-3.025
10	-3.098
11	-1.160
12	-2.922
13	-2.411
14	-0.828
15	-2.384
16	-2.384
17	-2.411
18	-0.828

4.3.2 Analysis

The Signal to Noise data are rolled up based on Factors, Levels, and the associated effect for each of the levels tested per parameter can be analyzed in Table 4.

Table 4. L18 Effect by Factor



	Factor	Level	N Rows	Effect (dB)
1	1: Illumination	1	9	-2.477
2	1: Illumination	2	9	-2.047
3	2:Octaves	1	6	-2.684
4	2:Octaves	2	6	-2.014
5	2:Octaves	3	6	-2.088
6	3:Illumination x Octaves	1	6	-2.081
7	3:Illumination x Octaves	2	6	-2.301
8	3:Illumination x Octaves	3	6	-2.404
9	5:Levels	1	6	-0.938
10	5:Levels	2	6	-2.779
11	5:Levels	3	6	-3.068
12	6: Illumination x Levels	1	6	-2.131
13	6: Illumination x Levels	2	6	-2.285
14	6: Illumination x Levels	3	6	-2.370
15	8: Octaves x Levels	1	6	-2.258
16	8: Octaves x Levels	2	6	-2.189
17	8: Octaves x Levels	3	6	-2.339

The data are then charted (Fig. 14) to reveal which effects have the greatest impact.

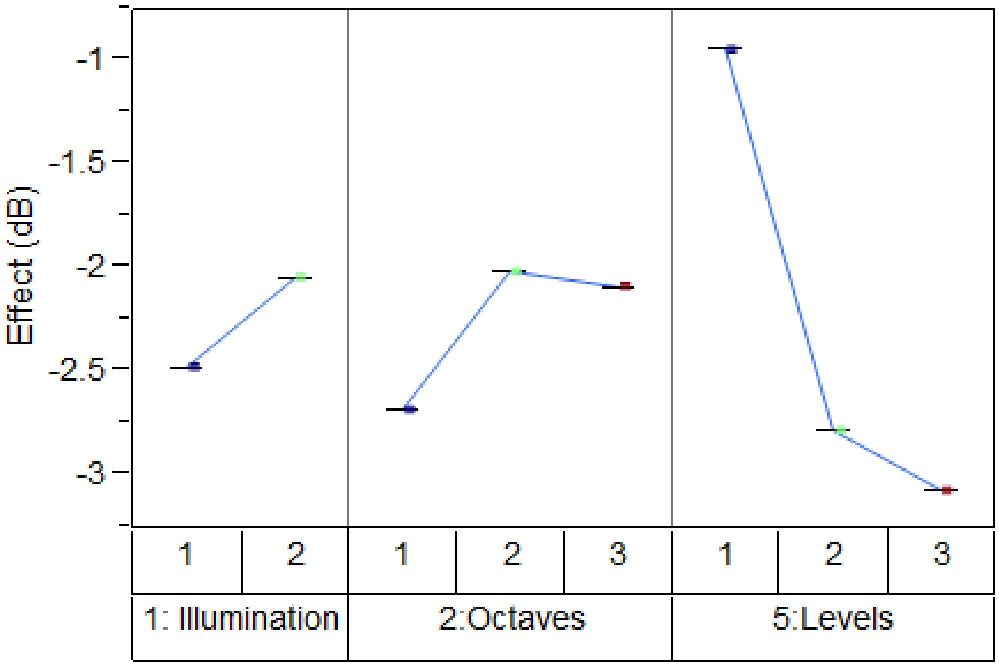


Fig. 14. Effect versus Scale-Space Parameters Showing the Effect of each Parameter Setting

Choosing the highest effect for each reveals which parameter level should be chosen that determines the optimal prediction values using equation (3).

4.3.3 Optimal Prediction

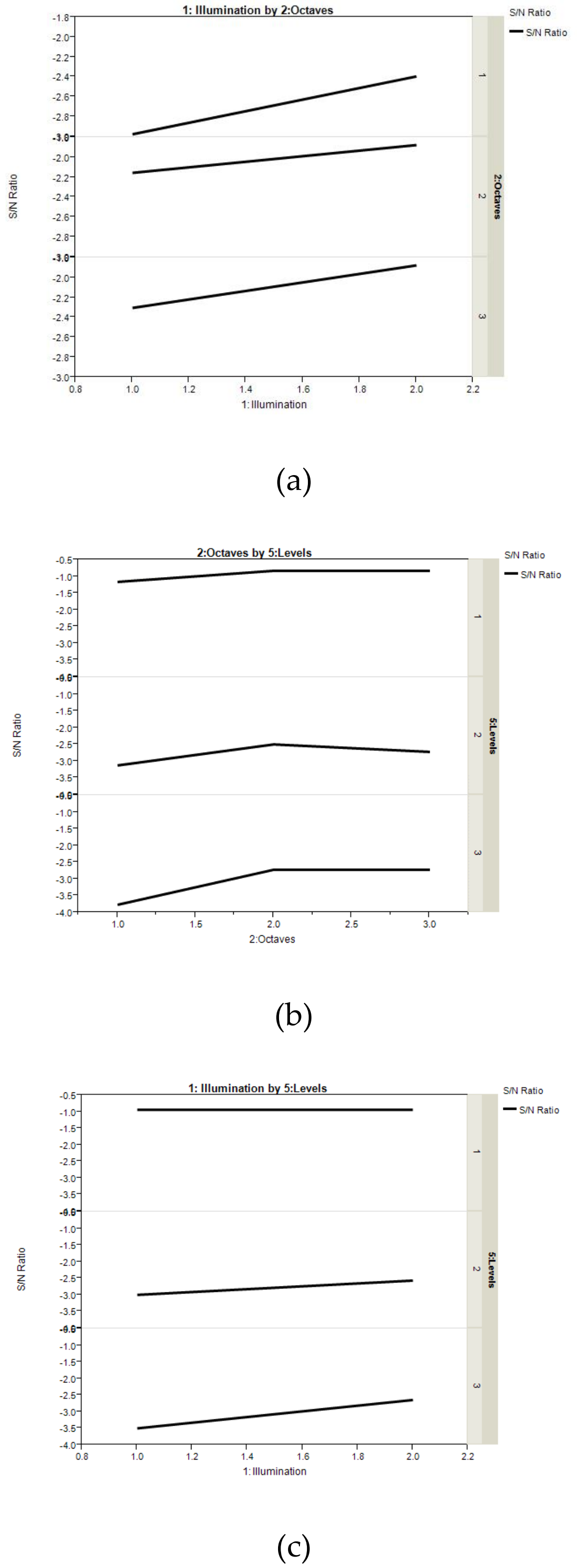
The goal of Taguchi DOE is to run a set of experiments using orthogonal arrays versus running all the experiments of a full factorial design. Then, a prediction of the optimal parameters can be computed by using Taguchi's Additive Model so that positive matching leads to improved feature detection.

The overall mean of all experiments run (%Correct-Experimental) is 77.63%. The next step is to investigate how the effects and levels are impacted by each factor. The information from the effects roll up, as shown in Fig. 14, and equation (4.5) are the basis of determining the optimal parameters based on the L18 results aforementioned. From Fig. 14, we can see how each parameter affects the feature detection algorithm. Then we can choose the parameter setting with the highest effect to improve feature detection. Choosing the highest effect on each reveals the following optimal parameters:

- Octaves = 2 (4 octaves)
- Levels = 1 (3 levels)
- Illumination = 2 (30% Brighter)

From equation (4.5), the %Correct-Predicted is 93.5%. The experiment is repeated for the optimal setup. The %Correct-Optimal is 90.9%.

Interactions and Analysis of Variance (ANOVA): Recalling from our Taguchi linear design two way interactions are plotted to determine if there are any interactions in our results. Fig. 15 a-c show no obvious interactions.



(a)

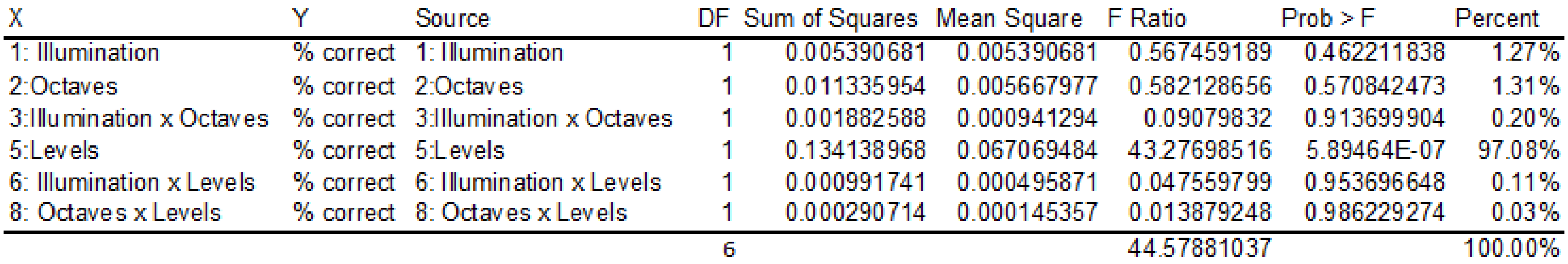
(b)

(c)

Fig. 15. Two Way Interactions (a) Illumination by Octaves (b) Octaves by Levels (c) Illumination by Levels

The next step is to complete an ANOVA to determine how each of the primary factors affects the results. From Table 5, there are 6 degrees of freedom (3 from the primary variables and 3 from their associated interactions).

Table 5. ANOVA Results



X	Y	Source	DF	Sum of Squares	Mean Square	F Ratio	Prob > F	Percent
1: Illumination	% correct	1: Illumination	1	0.005390681	0.005390681	0.567459189	0.462211838	1.27%
2:Octaves	% correct	2:Octaves	1	0.011335954	0.005667977	0.582128656	0.570842473	1.31%
3:Illumination x Octaves	% correct	3:Illumination x Octaves	1	0.001882588	0.000941294	0.09079832	0.913699904	0.20%
5:Levels	% correct	5:Levels	1	0.134138968	0.067069484	43.27698516	5.89464E-07	97.08%
6: Illumination x Levels	% correct	6: Illumination x Levels	1	0.000991741	0.000495871	0.047559799	0.953696648	0.11%
8: Octaves x Levels	% correct	8: Octaves x Levels	1	0.000290714	0.000145357	0.013879248	0.986229274	0.03%
			6			44.57881037		100.00%

The data in Table 5 show that levels is the SIFT parameter that affects performance the most. The contribution of levels is 97.1% of the actual optimal experimental run. Further examination of octaves and illumination reveals a contribution of ~ 2.5%. Through the ANOVA it is observed that the difference between predicted and the optimal is ~ 2.6%. The difference is attributed to a slight correlation among variables. However the error associated is small that the optimized SIFT parameters can be accepted as correct.

4.3.4 Conclusions from Analysis

From the above results, we can conclude that we have found the optimal SIFT parameters via Taguchi DOE. The optimal parameters in feature matching that are orthogonal with respect to each other since Taguchi was utilized. The results show that Levels have the greatest effect on the output. Now that the experimental parameters are optimized, a range of patterns are analyzed in the next section to determine how well the optimal settings work versus the standard SIFT parameter values used by David Lowe.

4.3.5 Experimental Results

SIFT Comparison: Experiments using SIFT with original and optimized parameters were run. Experiments compare the quality of feature detection for scale, rotation, and scale and rotation for the set of images shown in Fig. 16. The experimental results are summarized in Table 6.

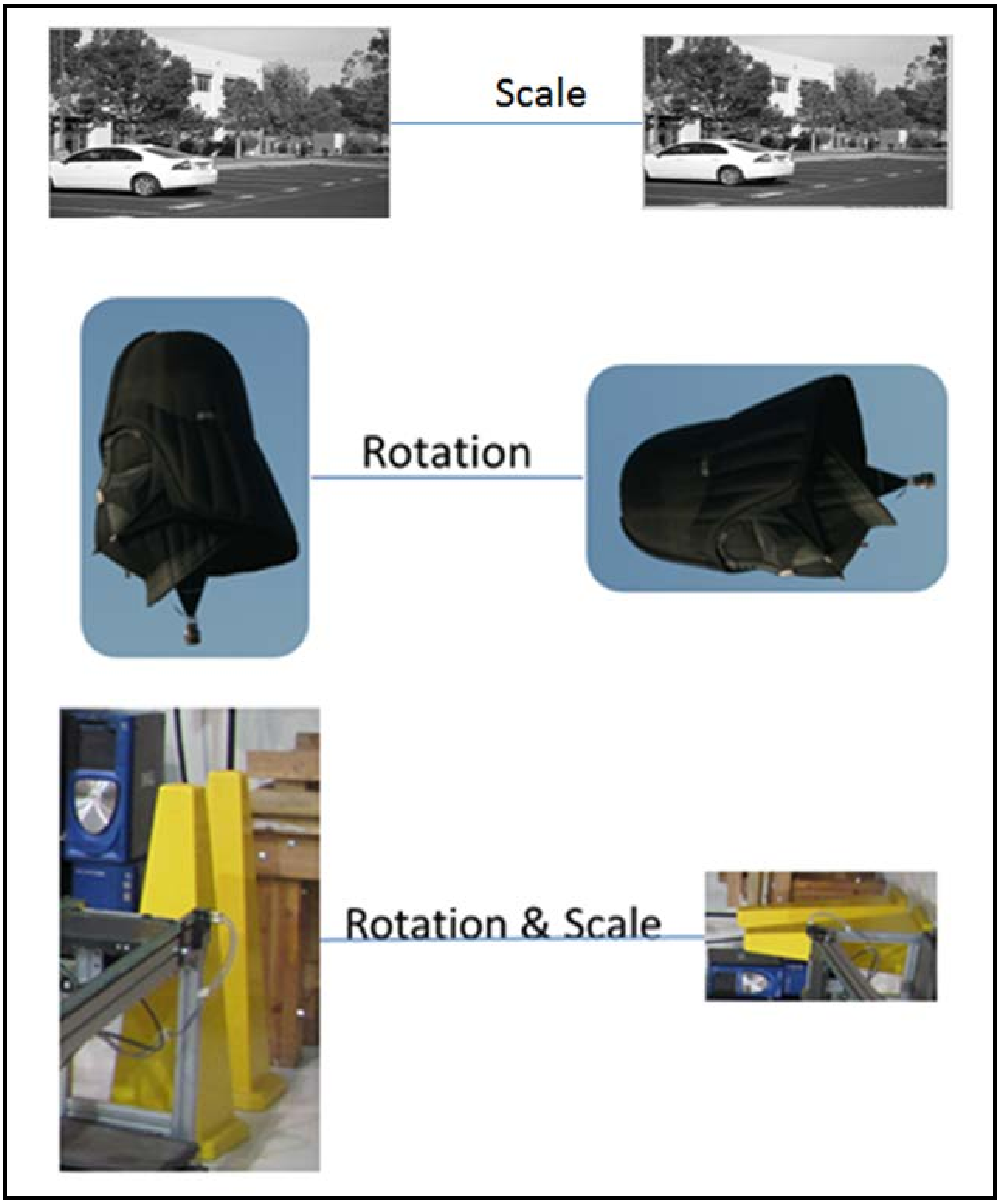
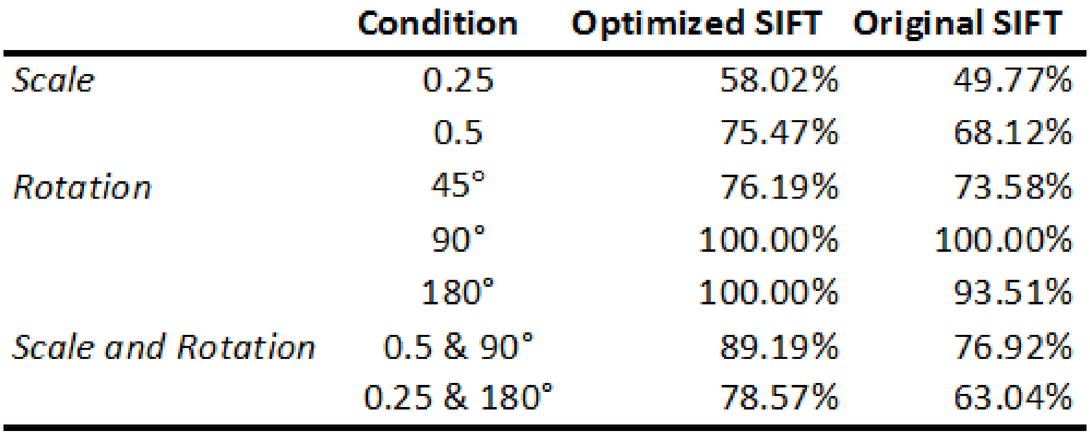


Fig. 16. Image Comparison for Scale and Rotation for 3 Different Images

The first observation shown in Table 6 is that the SIFT based algorithm that uses the optimized values for its variables performs better than the unoptimized variables for the images in Fig. 16.

Table 6. Results of SIFT vs. Optimized Parameters



	Condition	Optimized SIFT	Original SIFT
Scale	0.25	58.02%	49.77%
	0.5	75.47%	68.12%
Rotation	45°	76.19%	73.58%
	90°	100.00%	100.00%
	180°	100.00%	93.51%
Scale and Rotation	0.5 & 90°	89.19%	76.92%
	0.25 & 180°	78.57%	63.04%

Optimized SIFT parameters produced %correct matches that met or exceeded SIFT run with the original parameter values.

4.4 Conclusion

The Taguchi Design of Experiments approach was used to compute optimized values for each SIFT variable. The analysis used the L18 Design and Optimization. A comparison of SIFT using original and optimized variable values showed that the optimized values produced better performance.

Taguchi DOE was used to choose the best set of algorithm parameters so as to improve feature detection between two frames of the same pattern.

The L18 results demonstrated a 17% improvement of %Correct-Optimal in comparison to %Correct-Experimental. Additionally, the time it takes to run the Taguchi based experiments is far less than it would take to run a full factorial design to find the optimized answer should more levels be investigating for each factor. The optimal parameters show how effective design can save time and experimentation by conducting far fewer experiments. Maestas, et al., [29] L32 design proved that there would have to be 65,536 experiments using a full factorial design to test every possible combination versus the 32 experiments conducted per run. The algorithm executes in approximately 5 minutes per experiment and would take 7.5 months in processing time to conduct all the required experiments versus the 2 hours and 40 minutes of processing of the L32 design run.

Chapter 5: PATTERN RECOGNITION INSPIRED BY HUMANS

5.1 Introduction

It is essential to understand how humans perform pattern recognition and to what end can this be emulated in pattern recognition model technology space. To do that human physiology must be investigated in terms of learning and function. We need to know how the brain sees information and then create a model using neural networks. The purpose of this chapter is to create a model that is inspired by how the brain gets the information (i.e., what are the inputs?). The brain translates and processes the information to classify and recognize patterns.

5.2 Theory

The basis of human pattern recognition for visual information has been described in terms of neurons and synapse with the eye being the primary transmission piece to gather and distribute information to the brain. To create this type of behavior in a computer we must first create an object matrix of information using a camera lens, detect and keep only the features that are import to recognition, and create a network that processes this information. This model is analogous to human pattern recognition as aforementioned.

5.2.1 Feature Extraction

The idea is to detect edges accurately and then extract the SIFT based features from object image and add the edge data into to the object vector. As shown in the equation below, the gradient is calculated and can be separated into weak and strong edges based on the value of the edge magnitude. Strong edges have gradient values greater than an arbitrary threshold based on the desired output to detect edges. Weak edges can be ignored in the final output images. The result is to create an image that reveals the boundaries or edges of objects. The equations to find the local maxima that meet this arbitrary criteria are identified as edges with their corresponding direction

$$\left|\nabla \mathrm{I}\right| = \left|\nabla \mathrm{I}_x\right| + \left|\nabla \mathrm{I}_y\right| \geq a \in 0 \rightarrow 1 \qquad (5.1)$$

$$\text{where } \left|\mathrm{I}\right| = \sqrt{\mathrm{I}_x^2 + \mathrm{I}_y^2} \text{ and } \vartheta = \tan^{-1}\frac{\mathrm{I}_y}{\mathrm{I}_x}$$

The edge extraction methods available for implementation of (5.1) in Matlab are Canny, Sobel, Prewitt, or Roberts detection. Canny is chosen because this method finds edges by looking for the local maxima of the gradient of the input image using (5.1), which is the gradient result of using the derivative of a Gaussian filter.

Feature extraction also provides a way to label target objects and is the link between the target object and its SIFT based keypoints extracted using the process from the previous chapter. The system flow is shown in Fig. 17 The first step acquires the image input via PCIE Firewire and is Gaussian smoothed via equation (5.2) that outputs the magnitude and phase.

$$s(x,y) = I(x,y) * G(x,y) \qquad (5.2)$$

where, I(x,y) is the 2-D function of the original input image and G(x,y) is the Gaussian operator $G(x,y,\sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}$. Using this data the local maxima are computed.

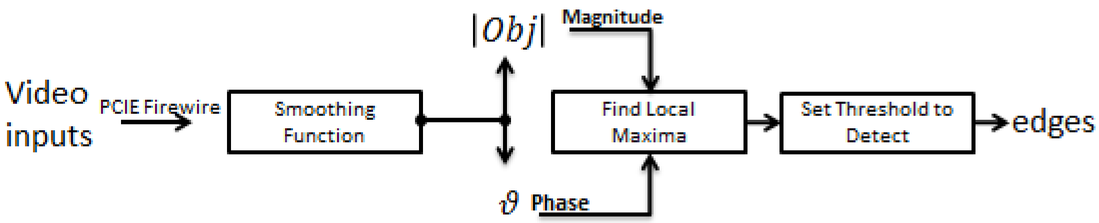
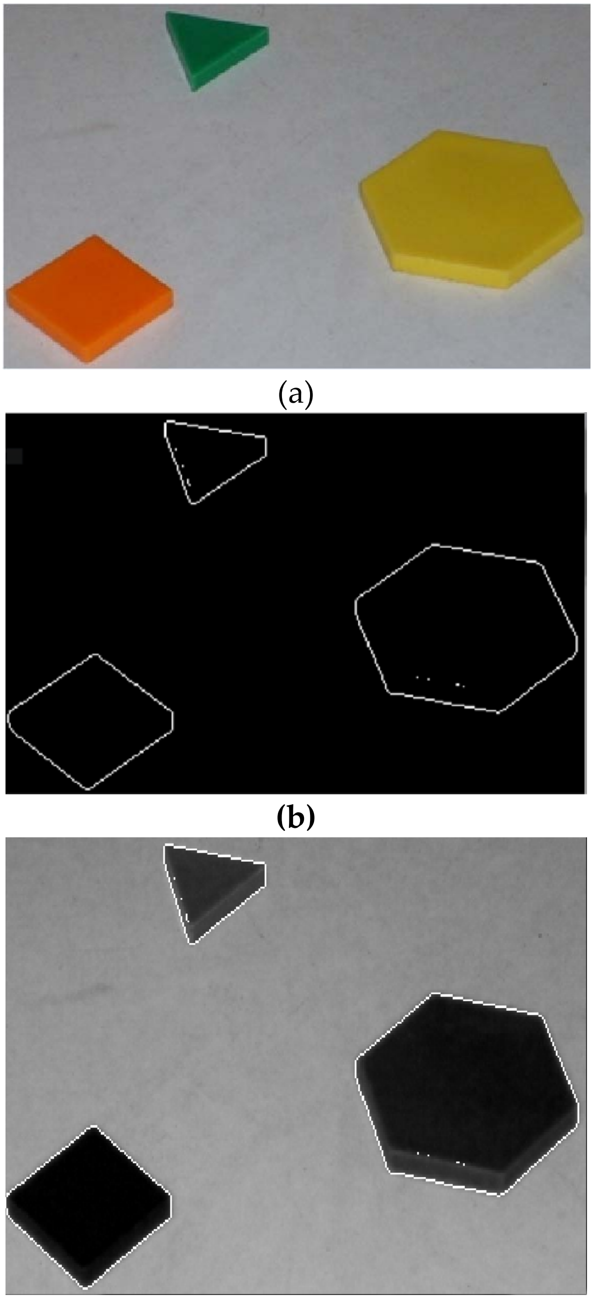


Fig. 17. Feature Extraction System Flow from Video Input to Edges

After the edges are found, they are stored in the object matrix with SIFT based keypoints associated with the object vector to be used as an input to the network. Fig. 18 is an example of an image that we will utilize in our pattern recognition experiments. The image consists of three different shapes: a green triangle, an orange square, and a yellow hexagon. The top image (a) is the image capture and the middle image (b) is the edge detection associated with the objects under consideration. The bottom image (c) is an overlay of the edges on the grayscale original.



(a)

(b)

(c)

Fig. 18. (a) Original Image Capture (b) Original Image Afer Edge Detection (c) Edge Overlay to Original Image

5.2.2 Creating the Object Matrix

The input images are imported using a camera. Simulink software utilizes the Video and Image Processing (VIP) toolbox to process information from a camera. An object can be placed within the object workspace on a vision table within the field of view (FOV) of a four camera system as shown in Fig. 19. Four cameras ensure the entire subspace is covered so that object can be captured into a desired vector format. The basic process is to place an object on a vision table, and use the VIP blockset in Simulink to capture a snap shot image which is converted into a pixilated scene for processing.

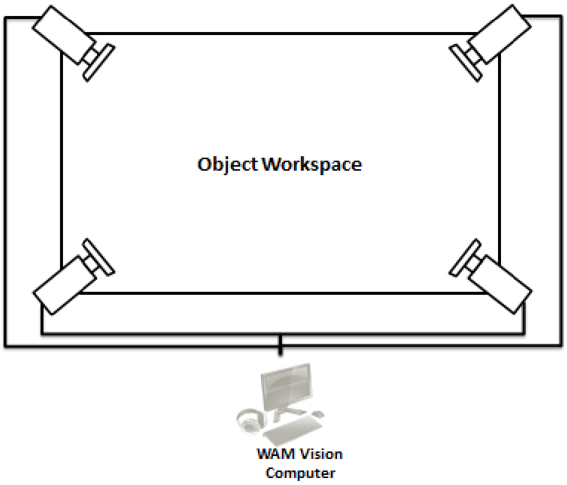
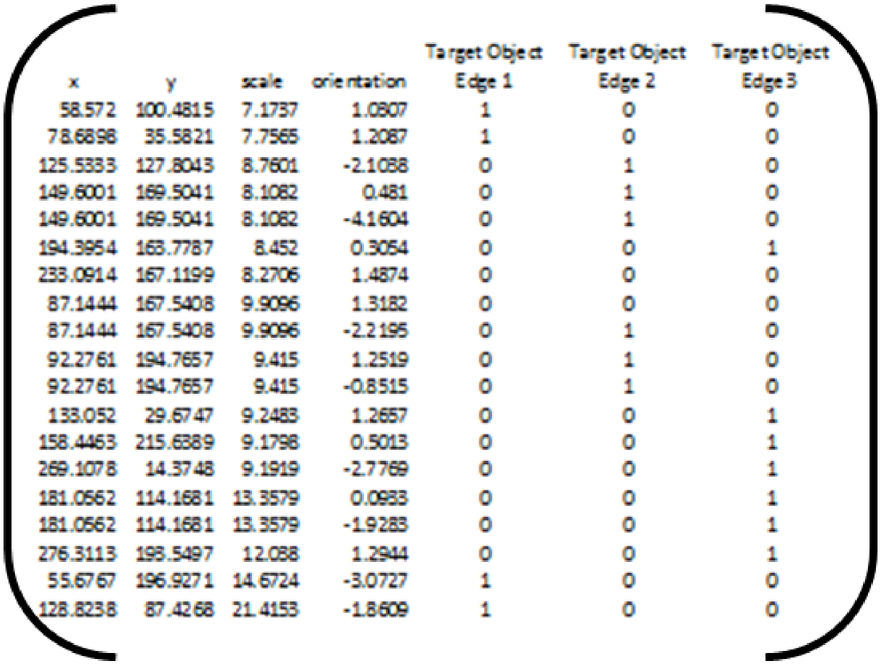


Fig. 19. Hardware Setup Schematic with 4 Cameras Connected to Station Controller

Edges of target objects and invariant keypoint data associated with each input image are extracted. The result is an object vector that is a combination of two separate matrices: one for the edges of targets and one of keypoints as shown in Fig. 20.



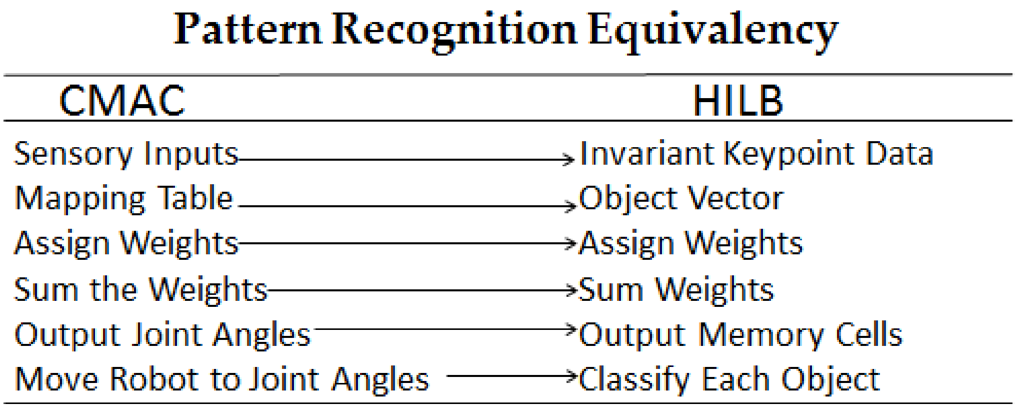
x	y	scale	orientation	Target Object Edge 1	Target Object Edge 2	Target Object Edge 3
58.572	100.4815	7.1737	1.0807	1	0	0
78.6898	35.5821	7.7565	1.2087	1	0	0
125.5333	127.8043	8.7601	-2.1088	0	1	0
149.6001	169.5041	8.1082	0.481	0	1	0
149.6001	169.5041	8.1082	-4.1604	0	1	0
194.3954	163.7787	8.452	0.3054	0	0	1
233.0914	167.1199	8.2706	1.4874	0	0	0
87.1444	167.5408	9.9096	1.3182	0	0	0
87.1444	167.5408	9.9096	-2.2195	0	1	0
92.2761	194.7657	9.415	1.2519	0	1	0
92.2761	194.7657	9.415	-0.8515	0	1	0
133.052	29.6747	9.2483	1.2657	0	0	1
158.4463	215.6389	9.1798	0.5013	0	0	1
269.1078	14.3748	9.1919	-2.7769	0	0	1
181.0562	114.1681	13.3579	0.0983	0	0	1
181.0562	114.1681	13.3579	-1.9283	0	0	1
276.3113	193.5497	12.088	1.2944	0	0	1
55.6767	196.9271	14.6724	-3.0727	1	0	0
128.8238	87.4268	21.4153	-1.8609	1	0	0

Fig. 20. Object Matrix of Keypoint Pairs and Image Targets

The object matrix is the input into the classification portion of the neural network. It associates the edges to the keypoint information for high quality keypoints extracted from Chapter 4. The reason the object matrix is constructed in this way is so that each object can be associated with its corresponding edge boundary. Because human perception is based on recognizing salient features in a scene we ensure that the most salient features (keypoint and edges) are paired and used as our sensory input into the neural network. This is analogous to Albus's CMAC where he associated sensory inputs with joint angles in robotic arm motion. Each

CMAC network component, shown in table 7, and its equivalent component in our human inspired locally based (HILB) neural network to shown the structure equivalency of using such a network like that of the Albus's approach.

Table 7. CMAC Translated to HILB



Pattern Recognition Equivalency

CMAC	HILB
Sensory Inputs →	Invariant Keypoint Data
Mapping Table →	Object Vector
Assign Weights →	Assign Weights
Sum the Weights →	Sum Weights
Output Joint Angles →	Output Memory Cells
Move Robot to Joint Angles →	Classify Each Object

5.2.3 Feed Forward Network

At this point the object matrix contains information that is localized and invariant to the object it is associated within the scene along with its corresponding edge data from that of Fig. 21. Each keypoint was extracted using our Taguchi optimization described in Chapter 3 to improve keypoint feature quality. The method has applied a rank filter to get rid of points not associated with objects so that only high quality keypoints are used as inputs into the neural network.

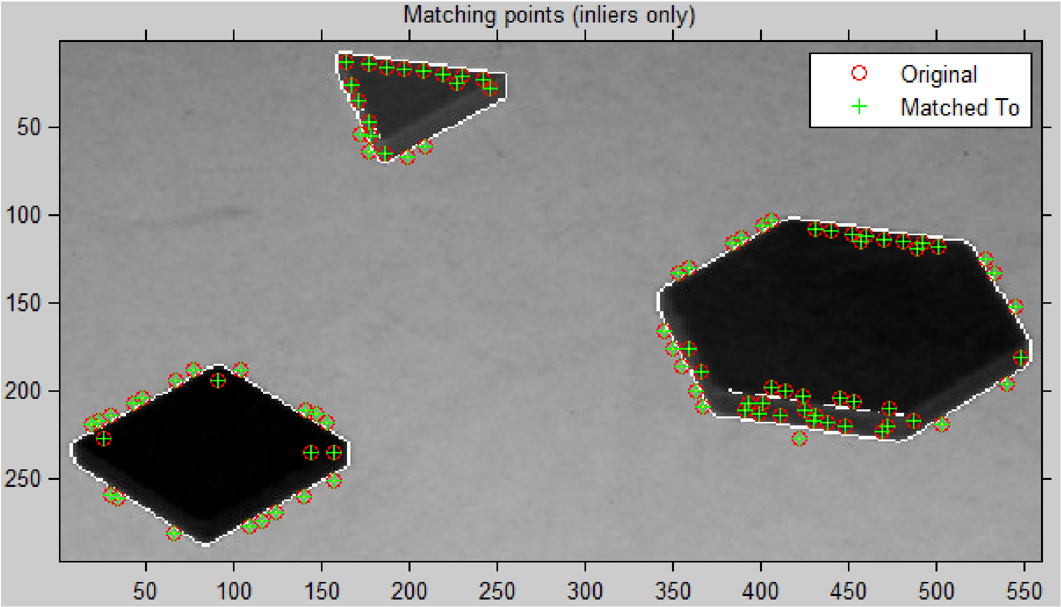


Fig. 21. Keypoint and Edge Overlay Used for Object Matrix

The object matrix that can be broken into two separate matrices: one for the edges of targets and one for SIFT based keypoints. The two matrices become the inputs into the classification portion of the neural network since they are associated with each other. Each keypoint-target row vector from the object matrix is fed into a neural network. The network is trained using a gradient based approach of backward propagation since we are providing the inputs to the network as edges and keypoints. The samples are randomly divided up into training (15%), validation (15%), and test (70%). Because we utilize the Simulink, the Neural Network Toolbox takes the weights and bias, adjusting them accordingly via back propagation. If the weights and bias are not leading to desired network behavior, the model can be retrained by increasing the percentage allocated as the training subset from say 15% to a different percentage. Since retraining selects a random subset to comprise the training data a different set of data points will be used, which affects the weight and bias values used for classification. The technical details are presented in the next section where our theory of keypoint classification is described. A portion of the data is partitioned for training and to establish the weight and bias for the neural network. A logarithmic transfer function is implemented to attain successful classification- a primary goal in pattern recognition. The classification occurs without the need for a global training database in which input sensory data that is not necessarily invariant (background data, raw pixel data, viewpoint angles

Using a neural network model the data from within the scene is partitioned as test data. The rest of the invariant keypoint data is fed into the neural network to adjust the controls (weighting, bias, and transfer functions) that are used to determine the probability that a pattern exists via successful classification.

5.2.4 Theory of Keypoint Classification

Classification is completed using Simulink by taking each external keypoint input from the object matrix, where q is the number of objects, r is the number of samples and multiplying it by a weight, w_{rx1}, adding a bias, b_{qx1}, to it and collecting it into A_{qx1} as a 1 or 0. If the result from (5.3) is greater than zero it is stored as a scalar element the output vector. The output for each cell can be represent as A and relates to the object, Obj and via the scalar product with the weight, W, and adds the bias. The general vectored equation is:

$$\mathbf{A}_{qx1} = (Obj_{qxr} \text{ x } W_{rx1}) + [1]_{qx1} B. \qquad (5.3)$$

Since we are classifying multiple objects and everything can't be a 1 or 0, a memory cell is needed. We choose what values to store in the memory cell by using a transfer function, tf. We choose the weighting and bias of the inputs based on the testing phase of target object attributes to achieve the desired classification. As shown in Fig. 22, keypoint data from the object matrix are extracted from the object vector and multiplied by a weight. A bias, B, is added to the product of object multiplied by the weight and shifts the output so that it can be included if the transfer function result as shown in 5.4.

$$tf_{qx1}=\text{logsig}(\mathbf{A}_{qx1}) \qquad (5.4)$$

If the result of tf is greater than 1 the transfer function outputs tf=1. If the result is equal to zero or less than or equal to -1, the output of the transfer function tf=0. If we have more than one object the transfer function results become a vector and if they greater than one are accumulated into an element in a matrix, M_{qx1}, known as memory cells that is the result of scalar multiplication and represented as a vector that classifies each target object based on its cell value.

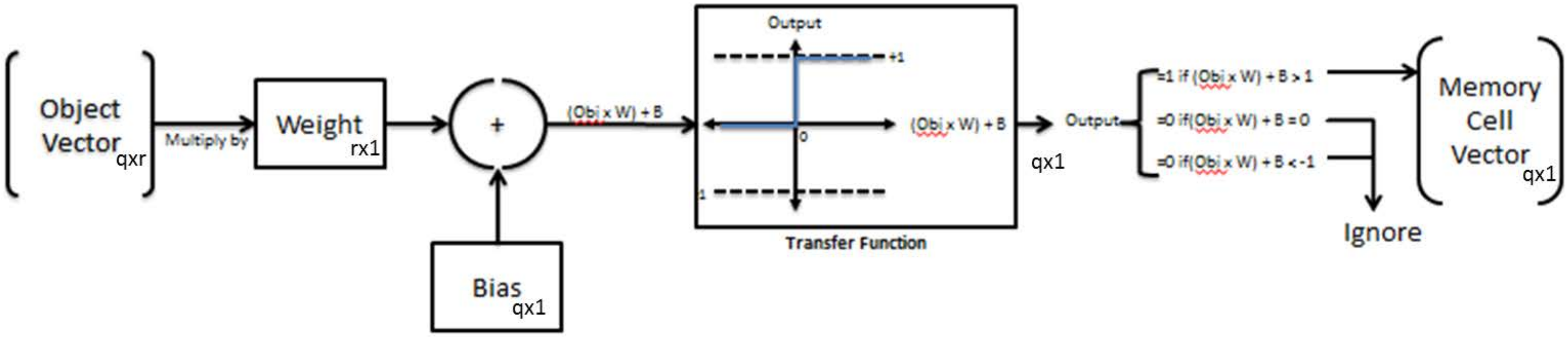


Fig. 22. Classification Based on the Object Vector (where q is the number of objects, r is the number of samples)

5.3 Experiments

The set of experiments to be conducted is based on classifying objects. The objects contain invariant information which has been extracted to the object matrix for input into the neural network using the keypoint-edge pairs as our inputs and target objects. Pattern recognition occurs with successful execution of experiments that classify a small number of objects, then a statistically significant set of objects classified, and a final comparison to demonstrate extensibility. The purpose is to show that (1) pattern recognition is achievable using our model and (2) once pattern recognition is achieved it can be extended to a large number of objects.

5.3.1 Perform Pattern Recognition on a Few Objects

The first part of pattern recognition consists of training the target objects. For illustration purposes, consider the 3 object input image from Fig. 23. The image consists of three triangles-in our case we use a triangle, square, and hexagon. Feature extraction (edges) and keypoint features (SIFT) establish the edges and keypoints that are represented as values in the object vector matrix. Once the object vector matrix is created a portion of the matrix is set aside for training. In our case we use 15% of the data to feed into the neural network. The output of the neural network is weights and bias that can be used to classify target objects. The percentage of each object is also computed using the test subset.

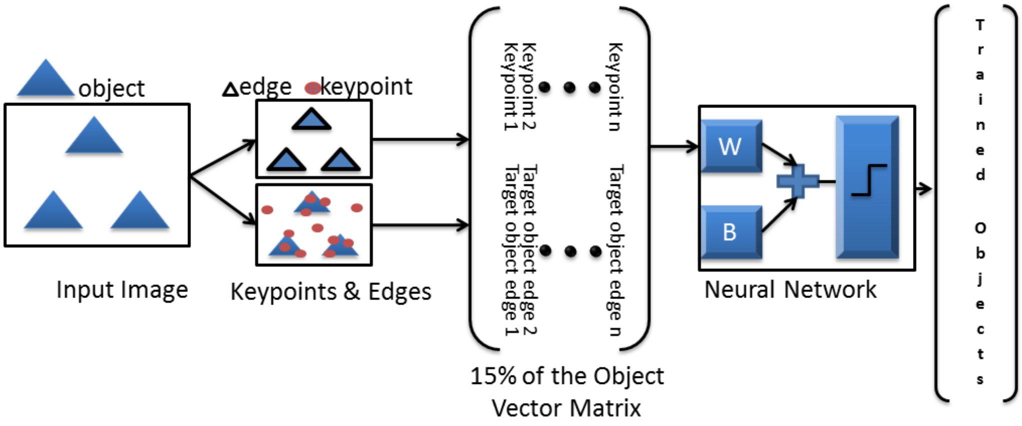


Fig. 23. Training Flow for a 3 object pattern recognition

5.3.1.1 Training of a 3 Object Pattern Recognition

Training data are fed from the object matrix into the network so that the weights and bias are set according to the lowest mean square error. The goal of training is to ensure the weights and biases are stable to the point where the network behavior will correctly classify known objects and results in supervised learning. Validation data are used to halt training when the mean square error stops improving and provides a set point for our final weights and bias that will be used to classify patterns. Ideally, validation error should decrease with initial training inputs and then when over fitting occurs the minimum level has been reached. Testing is not associated with training or validation and is an independent measure of network performance, i.e., how well patterns are being recognized. Using the mean squared error provides an average squared difference between outputs and targets using (5.5) which allows us to track the error to determine if it increase, decreases, or does not change.

Once training is completed, pattern recognition can be executed. Shown in Fig. 24, the rest of the object matrix is fed into the neural network. The mean square error is calculated and is the measure of performance of the pattern recognition system. The mean square is represented by 5.5,

$$mse = \frac{1}{N}\sum_{n=1}^{N}(t_n - a_n)^2 \qquad (5.5)$$

where N is the number of keypoint-edge pairs fed into the network. n is the specific row that contains the keypoint-edge input #, a_n is the input into the network from 5.3, and t_n is the respective output.

A plot of the mean square error from input identifies the level at which the error from consecutive training inputs was minimized and validated with consecutive increase in the mse for the validation partition of data. Because the mse is quadratic it will either be a minimum or no minimum so that consecutive increases are an observable trigger that we have found the minimum mse. The network weight and bias has reached a stable set so the patterns can be detected and result in the desired network behavior. The lowest error of validation set of data represents the level, also known as epoch in neural network literature, at which pattern recognition has occurred. The result is the output of patterns which can be normalized using the Simulink Neural Network toolbox statistical functions to calculate the desired network performance, the percentage, and number of patterns recognized using a truth table known as a confusion matrix in neural network literature

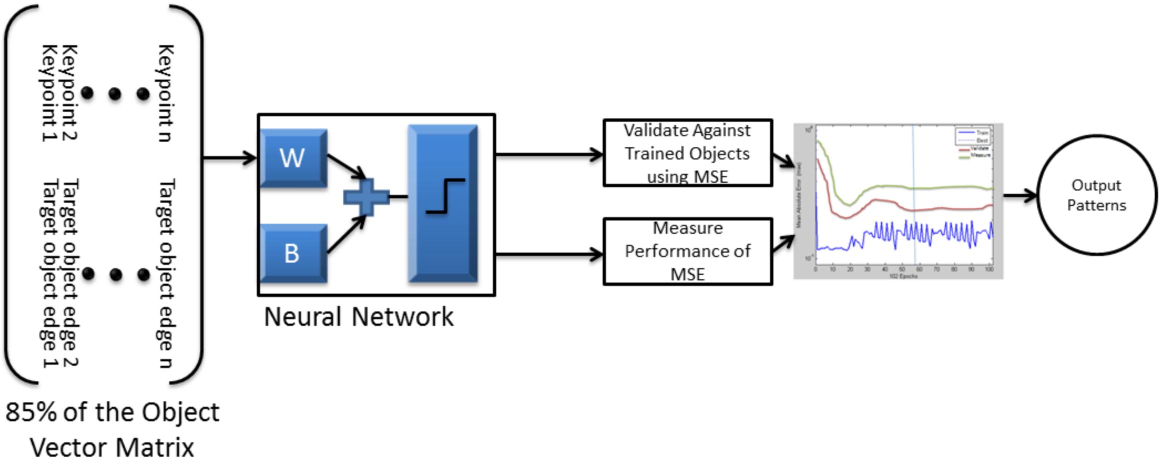


Fig. 24. Execution Flow of a 3 object pattern recognition

5.3.1.2 Validation of a 3 Object Pattern Recognition

Validation subset data is fed into the network simultaneously with the training data. Training data are used to get the gradient and finalize the weights using back propagation to assist in evaluating the accuracy of the network. Validation data error is tracked until the minimum error is detected. Since the goal is to have a network that has decreasing error to assist in evaluating network accuracy, the validation error should also decrease until it reaches a minimum. At the point when the network begins to over fit the training data the validation error increases. After six consecutive increases in the mean square error of the validation data, the network ceases to train. The gradient of the mse is computed and the weights and bias set based on the level (epoch) at the min validation error.

As shown in Fig. 25 our training data are paired with the validation checks to improve the chances of correct classification using a technique called early stopping. Early stopping shows that the minimum error level (va fail) occurs at an epoch of 17. The mean square error, at the va fail level, stops improving because the error starts to increase. The gradient is 0.00018559. The purpose of va fail and gradient is to determine the set point for the weight and bias final at the minimum mse level. Using the gradient is computationally advantageous because it stops changing when the mse is at a min. Then validation checks over and over until we are confident we have found values of weight and bias that allow confident classification.

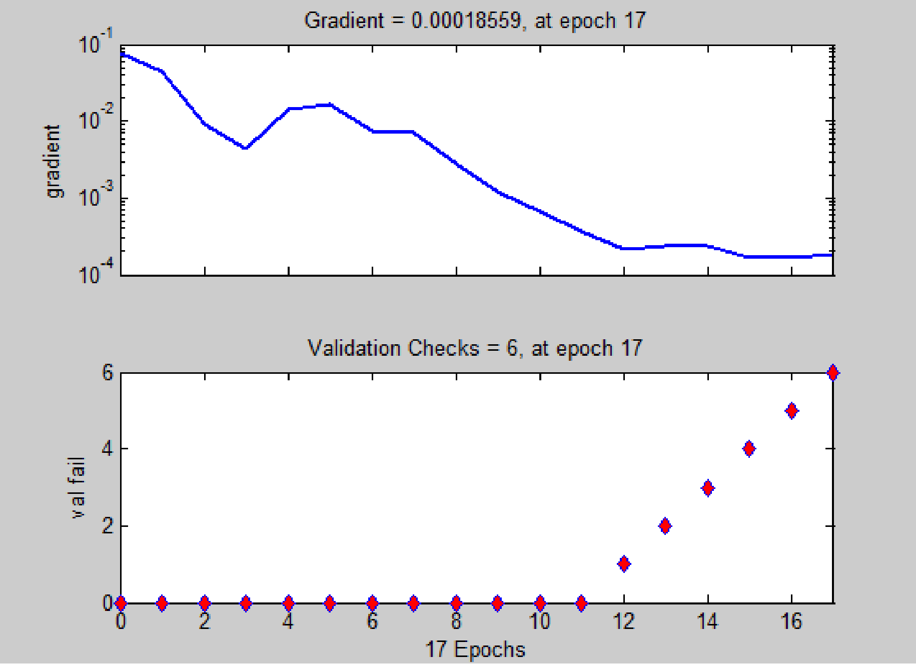


Fig. 25. Training Status for 3 Object Classification

The error histogram buckets the error associated with training, validation, and test and shows a mean centered on zero error from (5.5) as shown in Fig. 26.

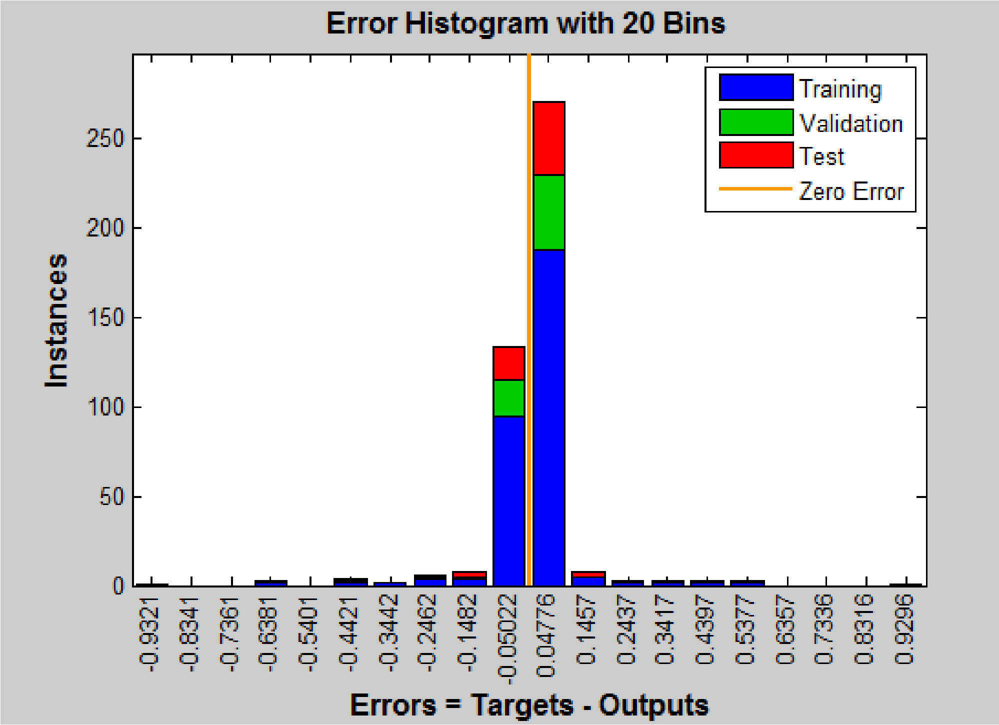


Fig. 26. Error Histogram for 3 Object Classification

From the performance data in Fig. 27, we observe best validation performance is at a mean square error of 0.019913 at epoch 11, with ultimate stability after six consecutive validation checks occurring at epoch 17. The means that we have three separate measures that account for setting the final weights and bias set for the neural network. This provides accurate classification and we can re-plot the mse vs. the level of convergence to see if data does min out as the gradient in Fig. 25 suggests.

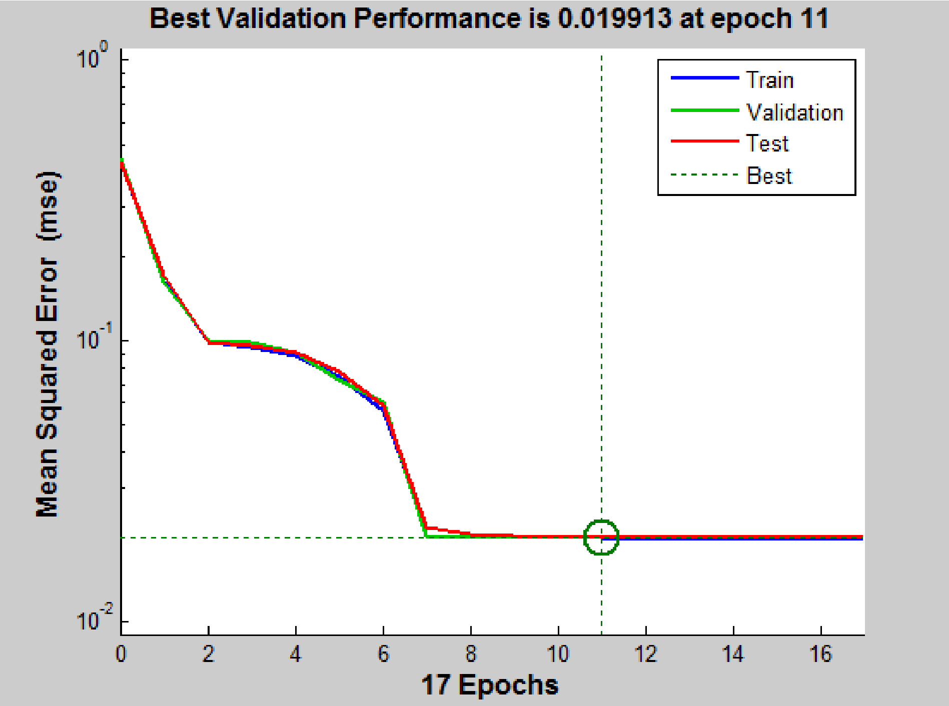


Fig. 27. Performance Plots for 3 Object Classification

5.3.1.3 Testing of a 3 Object Pattern Recognition

Testing is not associated with training or validation and is an independent measure of network performance, i.e., how well patterns are being recognized. Testing tells us how reliably we are recognizing patterns given the rest of the inputs and noise associated with them. Using the mean squared error of each input and corresponding output we can calculate the mse as an average squared difference between outputs and targets (keypoint edge pairs) using (5.5). We use the difference to minimize the average sum of these two values for each of the inputs.

$$mse = \frac{1}{N}\sum_{n=1}^{N}(t_n - a_n)^2 \quad (5.5)$$

Each object from Fig. 21 is assigned to a class based off of the edges from the object matrix to distinguish it from the other objects. Class 1 is the triangle. Class 2 is the hexagon. Class 3 is the square. Upon further investigation of the receiver operating characteristic (ROC), a neural network measure for how well classification occurred based on the changing weights and bias, for each phase (training, validation, and test), shown in Fig. 28, we observe true positive rates for each phase hover near 1. ROC curves that converge to the left and top edges of the plot mean better the classification occurred versus curves that do not show this type of convergence and drop away from the upper left corner of the ROC plot.

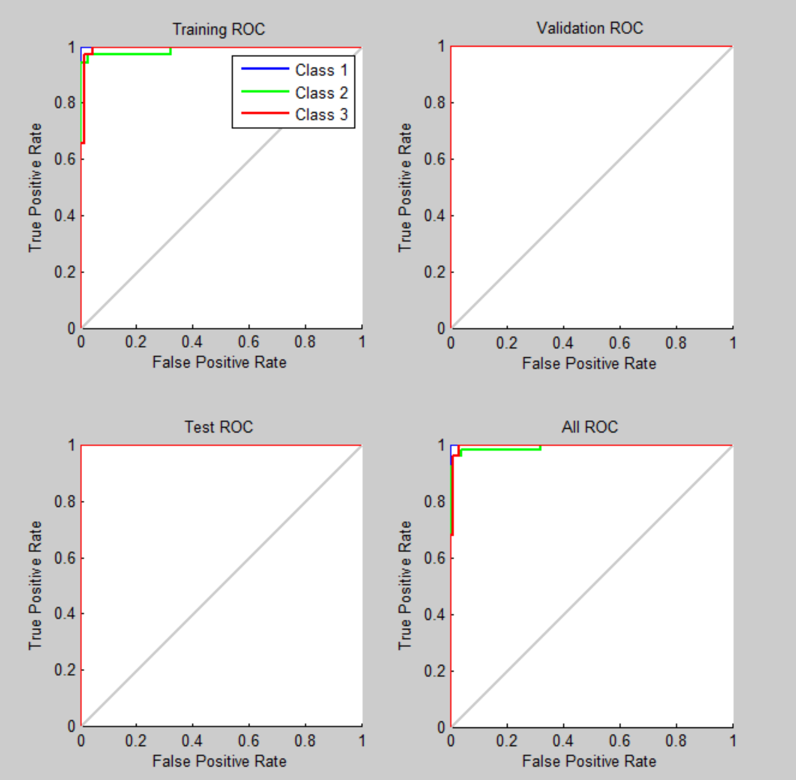
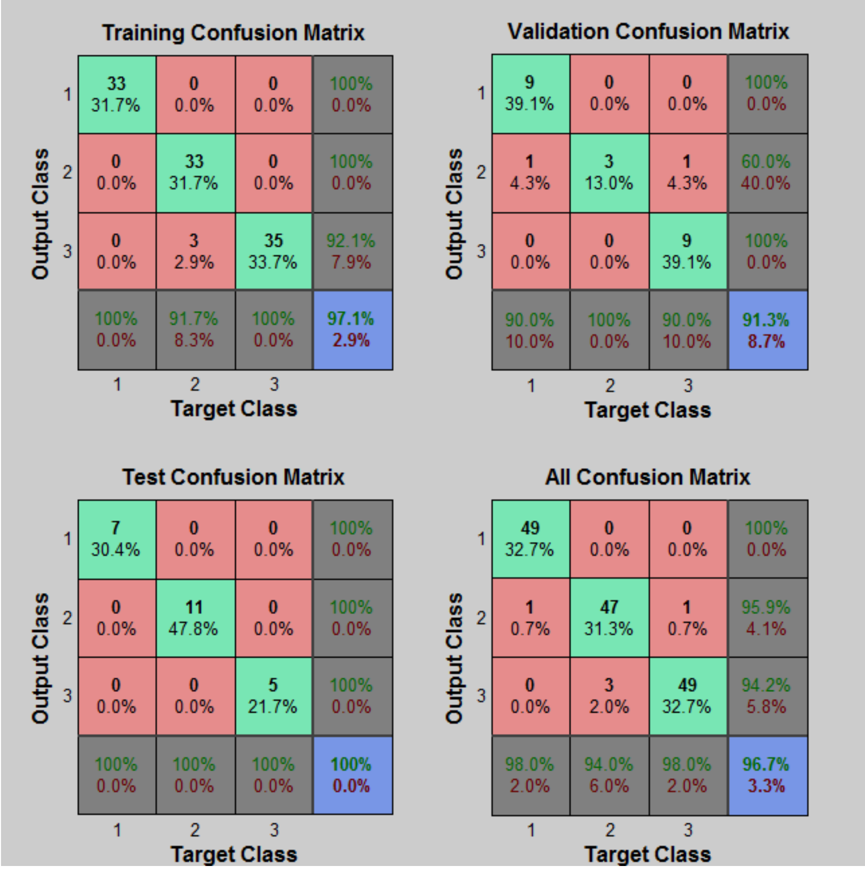


Fig. 28. Receiver Operating Characteristic Plots for: training (upper left), validation (upper right), test (lower left), and comined plot for all phases (lower right)

Next, we analyze confusion matrices shown in Fig. 29. The confusion plots give the percentage of each object recognized for all phases of network execution similar to the ROC plots. The most important phase, which is the test phase, result in zero error associated with our 3 object pattern recognition and that 100% of the test data classified correctly to one of the three input objects. This means that the weights and biases need no more adjustments because successful classification occurred.



Training Confusion Matrix

Output Class				
1	33 31.7%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	33 31.7%	0 0.0%	100% 0.0%
3	0 0.0%	3 2.9%	35 33.7%	92.1% 7.9%
	100% 0.0%	91.7% 8.3%	100% 0.0%	97.1% 2.9%
Target Class	1	2	3	

Validation Confusion Matrix

Output Class				
1	9 39.1%	0 0.0%	0 0.0%	100% 0.0%
2	1 4.3%	3 13.0%	1 4.3%	60.0% 40.0%
3	0 0.0%	0 0.0%	9 39.1%	100% 0.0%
	90.0% 10.0%	100% 0.0%	90.0% 10.0%	91.3% 8.7%
Target Class	1	2	3	

Test Confusion Matrix

Output Class				
1	7 30.4%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	11 47.8%	0 0.0%	100% 0.0%
3	0 0.0%	0 0.0%	5 21.7%	100% 0.0%
	100% 0.0%	100% 0.0%	100% 0.0%	100% 0.0%
Target Class	1	2	3	

All Confusion Matrix

Output Class				
1	49 32.7%	0 0.0%	0 0.0%	100% 0.0%
2	1 0.7%	47 31.3%	1 0.7%	95.9% 4.1%
3	0 0.0%	3 2.0%	49 32.7%	94.2% 5.8%
	98.0% 2.0%	94.0% 6.0%	98.0% 2.0%	96.7% 3.3%
Target Class	1	2	3	

Fig. 29. Confusion Plots for: training (upper left), validation (upper right), test (lower left), and combined plot for all phases (lower right)

5.3.2 Generalization Confirmation

We have demonstrated that successful classification using the final weights and bias is possible using our model test subset. Next, system generalization will show the effectiveness of our model to classify other images. The images are a subset of the objects that were trained. They are placed with other objects in arbitrary positions and orientations in the scene as shown in Fig. 30. Then, without retraining, we run the algorithm on these images. Once the weights and bias are established, objects of the same class are tested in different positions and orientations along with untrained objects. The purpose of this experiment is to test, using the final weights and bias from our original training, work in recognizing the same objects in a general setting. For a c classes of objects technical details we expect are as follows for the general cases of classification using network weights and bias using (5.6) and (5.7). I is the number of objects, j is the number of samples of keypoint-edge pairs, w is final weight from the aforementioned training phase, and b is the final bias from the aforementioned training. P is the result of object (Obj) times the weight plus the bias for each object.

Class c

$$\begin{bmatrix} Obj^{1}{}_{s1} & Obj^{1}{}_{s2} & Obj^{1}{}_{s3} & Obj^{1}{}_{s4}...Obj^{1}{}_{sj} \\ Obj^{2}{}_{s1} & Obj^{2}{}_{s2} & Obj^{2}{}_{s3} & Obj^{2}{}_{s4}...Obj^{2}{}_{sj} \\ Obj^{3}{}_{s1} & Obj^{3}{}_{s2} & Obj^{3}{}_{s3} & Obj^{3}{}_{s4}...Obj^{3}{}_{sj} \\ Obj^{4}{}_{s1} & Obj^{4}{}_{s2} & Obj^{4}{}_{s3} & Obj^{4}{}_{s4}...Obj^{4}{}_{sj} \\ . & & & \\ . & & & \\ . & & & \\ Obj^{i}{}_{s1} & Obj^{i}{}_{s2} & Obj^{i}{}_{s3} & Obj^{i}{}_{s4}...Obj^{1}{}_{sj} \end{bmatrix}_{ixj} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ . \\ . \\ . \\ w_j \end{bmatrix}_{jx1} + \begin{bmatrix} 1_1 \\ 1_2 \\ 1_3 \\ 1_4 \\ . \\ . \\ . \\ 1_j \end{bmatrix}_{jx1} b = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ . \\ . \\ . \\ p_j \end{bmatrix}_{jx1} \qquad \textbf{(5.6)}$$

Class 1

$$\begin{bmatrix} Obj^1{}_{s1} & Obj^1{}_{s2} & Obj^1{}_{s3} \end{bmatrix}_{1x3} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}_{3x1} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}_{3x1} b = \begin{bmatrix} p_1 \\ 0 \\ 0 \end{bmatrix}_{3x1} \quad \textbf{(5.7a)}$$

Class 2

$$\begin{bmatrix} Obj^2{}_{s1} & Obj^2{}_{s2} & Obj^2{}_{s3} \end{bmatrix}_{1x3} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}_{3x1} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}_{3x1} b = \begin{bmatrix} 0 \\ p_2 \\ 0 \end{bmatrix}_{3x1} \quad \textbf{(5.7b)}$$

Class 3

$$\begin{bmatrix} Obj^3{}_{s1} & Obj^3{}_{s2} & Obj^3{}_{s3} \end{bmatrix}_{1x3} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}_{3x1} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}_{3x1} b = \begin{bmatrix} 0 \\ 0 \\ p_3 \end{bmatrix}_{3x1} \quad \textbf{(5.7c)}$$

The first image contains the triangles we initially trained on within a different scene that includes part of a phone, a flashlight, and a tape measure.

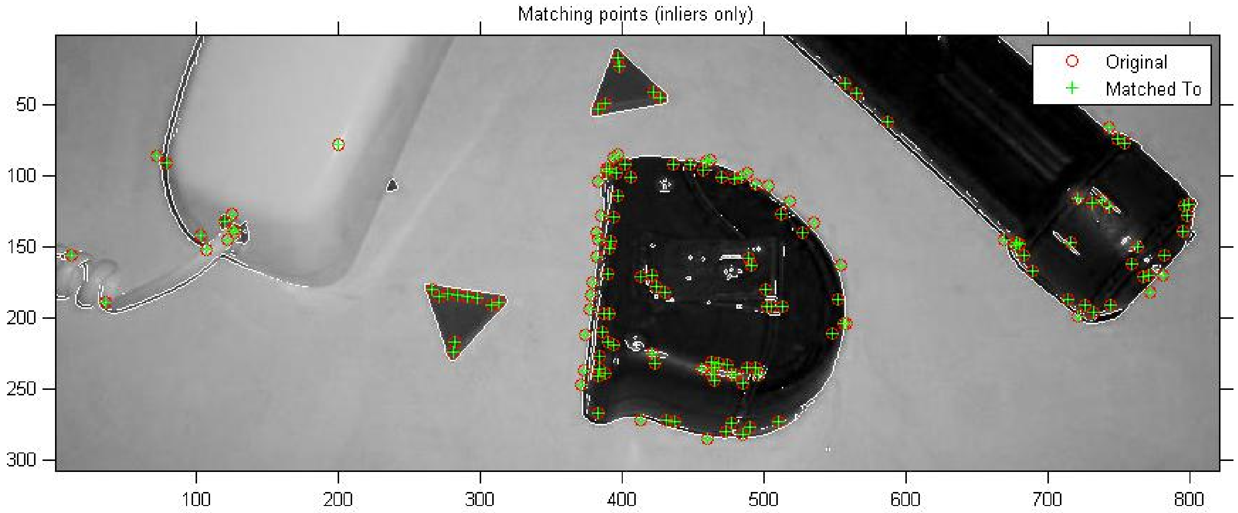
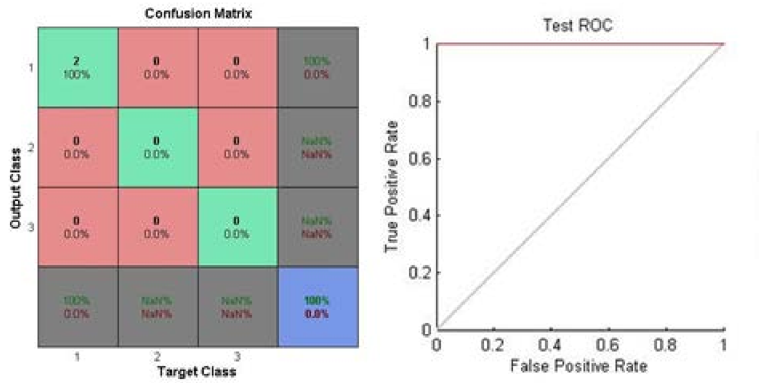


Fig. 30. Classification of a Triangle within a Scene

We remake the test confusion and ROC, shown in Fig. 31, for the newly added input data from the image scene above for the triangles. The triangle object corresponds to the first output class in the matrix below and we observe a repeatable result in accordance with (5.7a) that of the first run with the triangle being correctly classified and improved pattern recognition from our test data. The ROC also confirms correct classification.


Confusion Matrix

Output Class	1	2	3	
1	2 100%	0 0.0%	0 0.0%	100% 0.0%
2	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
3	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
	100% 0.0%	NaN% NaN%	NaN% NaN%	100% 0.0%

Fig. 31. Confusion Matrix and ROC results of a Triangle with a Scene

The second image (Fig. 32) contains the squares we initially trained on within a different scene that includes a phone handset, part of a flashlight, and a tape measure.

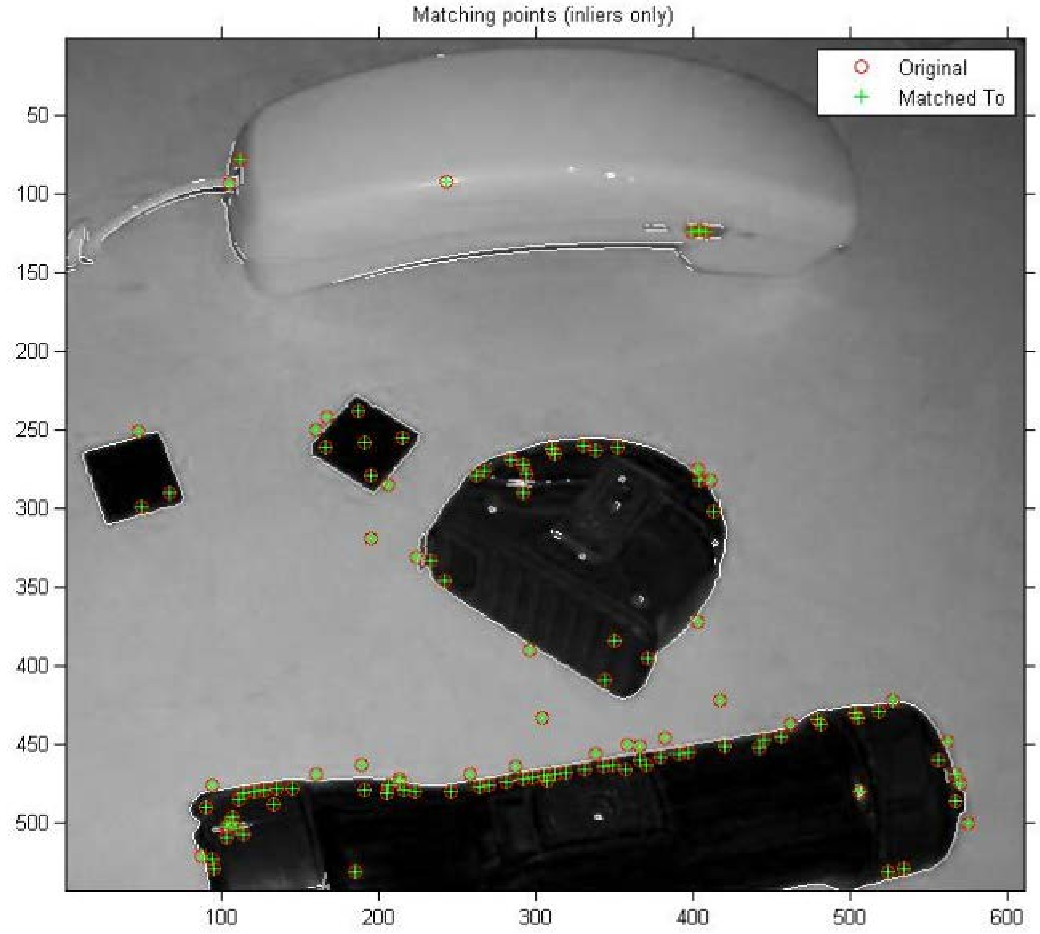
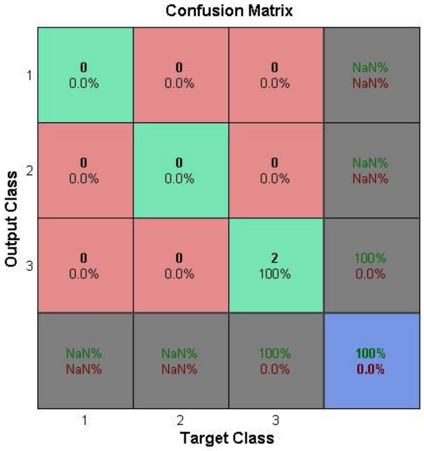


Fig. 32. Classification of a Square within a Scene

As with the untrained triangles above, we observed a repeatable classification result with the square (Class 3) in accordance with (5.7c) to that of the first run shown in Fig. 33.



Output Class \ Target Class	1	2	3	
1	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
2	0 0.0%	0 0.0%	0 0.0%	NaN% NaN%
3	0 0.0%	0 0.0%	2 100%	100% 0.0%
	NaN% NaN%	NaN% NaN%	100% 0.0%	100% 0.0%

Fig. 33. Confusion Matrix of a Square within a Scene

Our system validation shows the effectiveness of our model to classify objects in other images correctly. The images were a subset of the objects (square, triangle, and hexagon) that were trained. The squares were placed with other objects in arbitrary positions and orientations in the scene. Then, without retraining, we ran the algorithm on these images which should classify objects correctly based on the initial training images. Our next set of experiments will go another step in pattern recognition to extensibility to multi-object recognition.

5.3.3 Perform Pattern Recognition on Several Objects

We used Simulink's Neural Network Toolbox to build our HILB pattern recognition model shown in Fig. 34. HILB is represented by a two layer feed forward network that uses a sigmoid transfer function, which from the literature review, is a plausible way in which humans recognize patterns. Based on previous work, a hidden layer can be used to simulate additional neurons to provide classification of the objects in our images.

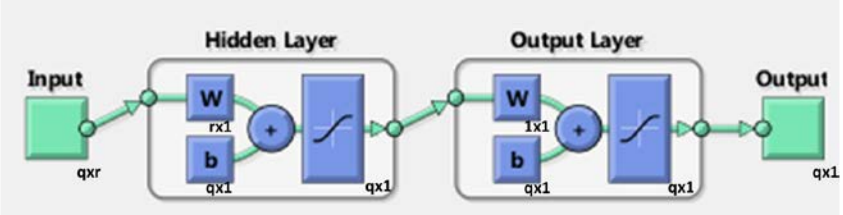


Fig. 34. Two Layer Feedfoward Network Diagram from Simulink [57]

The next experiment extends pattern recognition to a cluttered scene with multiple objects to test if the method is extensible. The goal is to prove statistical significance if images with 30 objects can be classified successfully. Pattern recognition utilizes the rest of the data using neural network hidden nodes—ways of adding extra neurons like we believe humans do. The number of neurons can be adjusted on the fly once the object vector with keypoints and edge target objects is known, something not done in prior use of neural networks. Fig. 35 is an example of our pattern recognition model. Starting at the top level system overview where inputs, keypoints and edge target objects shown in Fig. 36, are fed into the neural network shown as Input 1 (x{1} for each target input) and the output is patterns (y{1}).

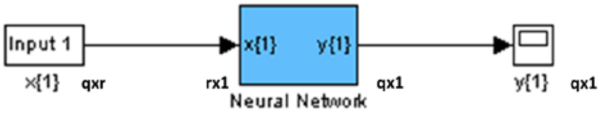
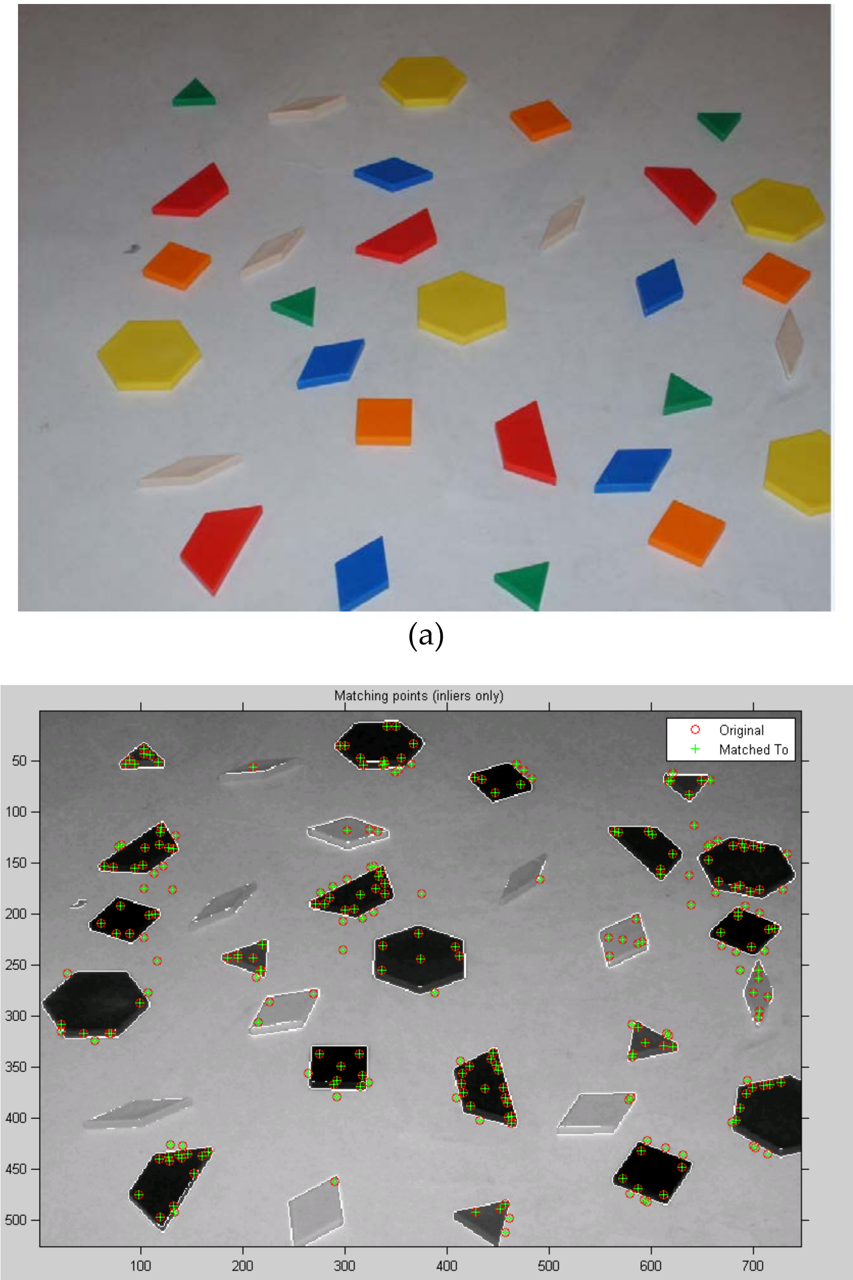


Fig. 35. Top Level View-Neural Network of a 3 object pattern recognition



(a)

(b)

Fig. 36. (a) Original Image Capture (b) Edge and Keypoint Overlay to Original Image

The second level of the neural network is the two layers and shown in Fig. 37. The first layer processes the input x{1} into a format p{1} input into the first layer where the weight IW{1,1} is added to the bias b{1} via netsum and sent to the transfer function , shown in the exploded view of layer 1 in Fig. 37. This results in the output of the first layer—a classification vector a{1}. The first classification vector, a{1}, is fed into a second layer where the process is repeated. The output is classification, a{2}, of data that become the memory cells used for training, y{1} and a decreased probability of false pattern recognition since two layers refines data.

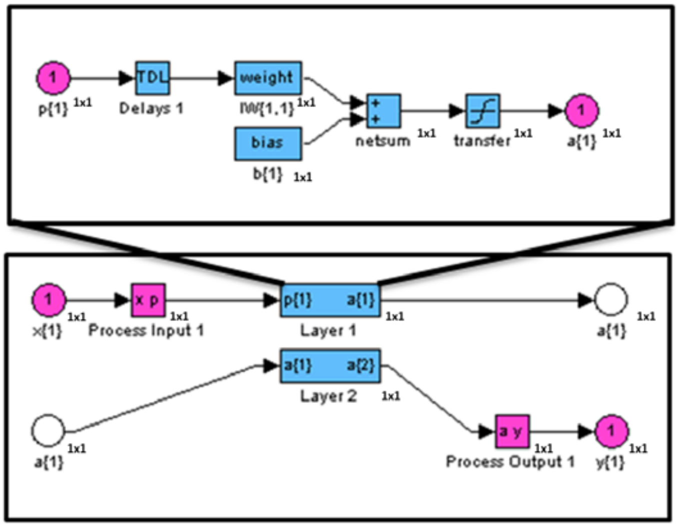


Fig. 37. Second Level View-Two Layers of a multi object pattern recognition neural network with Layer 1 exploded view

5.3.3.1 Training on Several Objects

Once training is completed and the weights and bias established the rest of the data are fed through the neural network and the mean square error is computed on the test data to complete the pattern recognition.

5.3.3.2 Validation on Several Objects

As shown in Fig. 38 our training data are paired with the validation mean square error checks and reaches a minimum at 31 levels (epoch) and the mean square stops over fitting and gradient of 0.0071904.

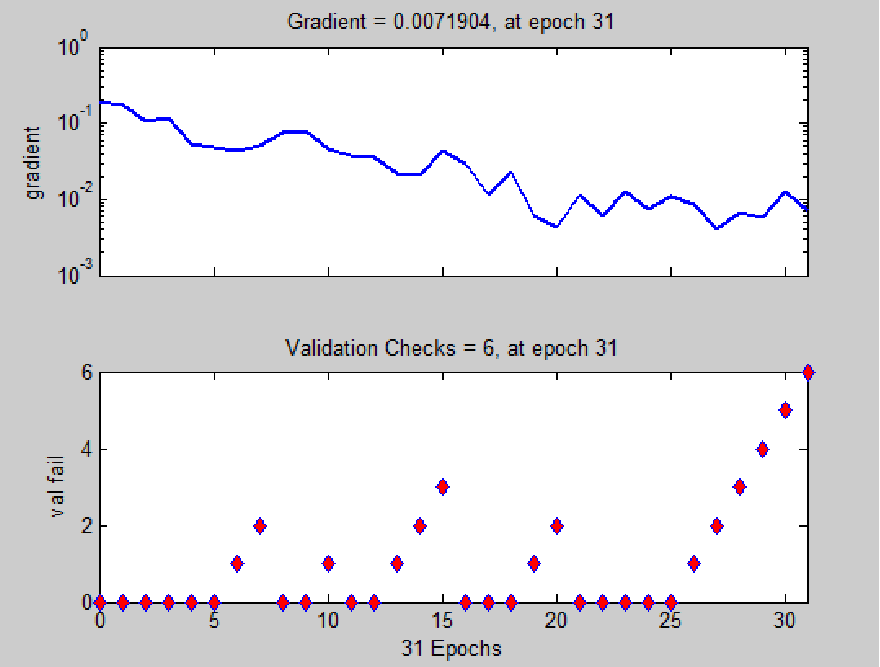


Fig. 38. Training Status on Several Objects

We can see from the validation check that for additional objects have more mean square variation before stability is confirmed with six consecutive rises in error compared to the 3 object recognition experiment earlier.

The error histogram buckets the error associated with training, validation, and test and shows a mean centered near zero error as shown in Fig. 39. This is our first indication that classification is successful and performance data should be analyzed.

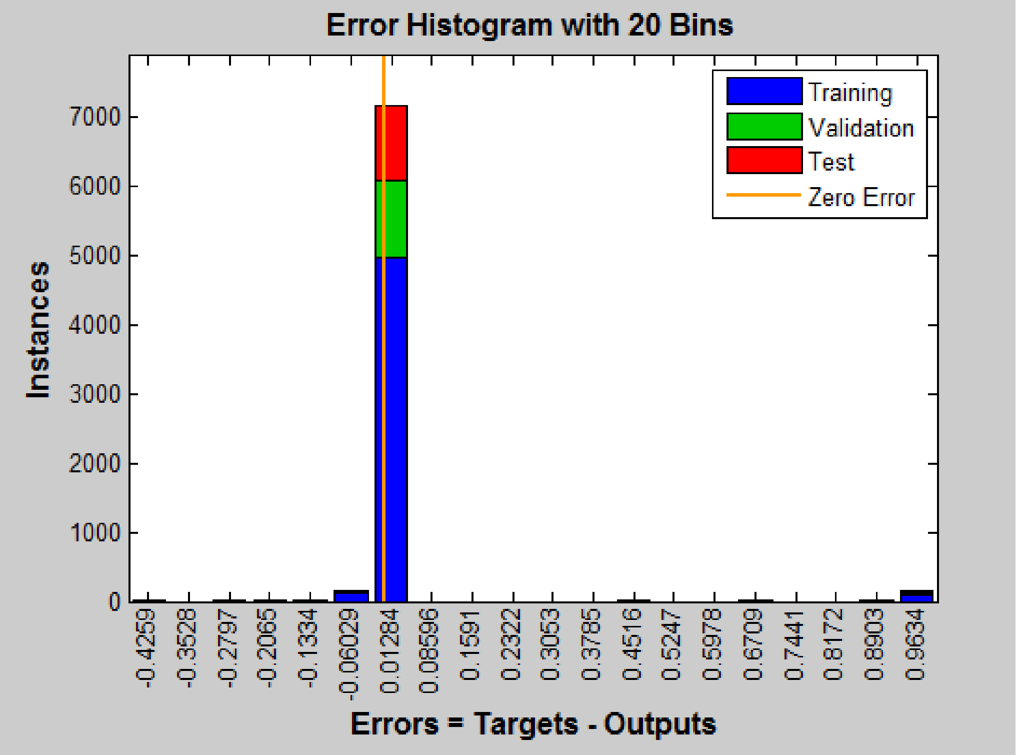


Fig. 39. Error Histogram **on Several Objects**

5.3.3.3 Testing on Several Objects

From the performance data in Fig. 40, we observe best validation performance is at a mean square error of the test data 0.019617 at epoch 25, with ultimate stability occurring at epoch 31.

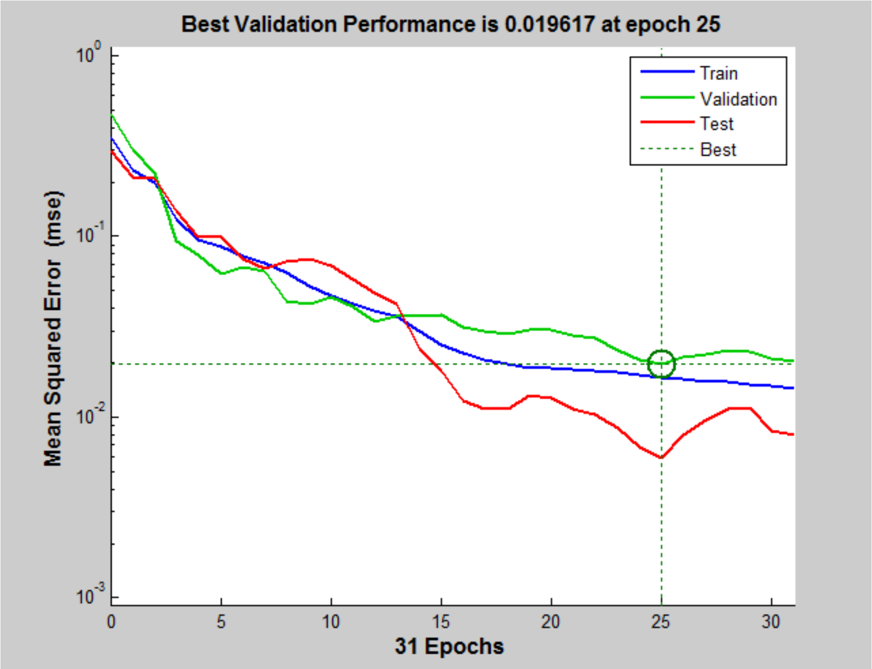


Fig. 40. Performance Plots on Several Objects

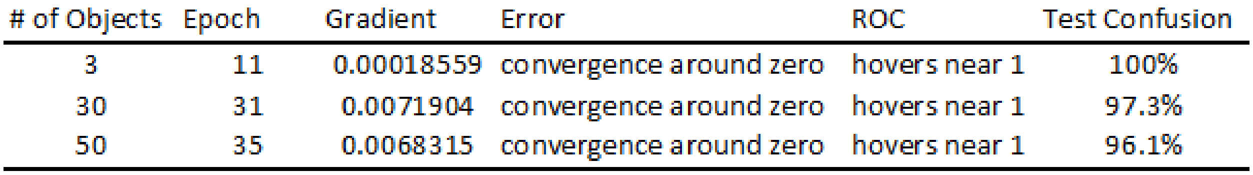
Similar to Fig. 28 plots, the receiver operating characteristic (ROC) for each phase (training, validation, and test) shows true positive rates that hover near 1. The ROC results mean better the classification is occurring.

The test phase shows that we have zero error associated with our 30 object pattern recognition and 97.3% of the test data could be classified based on the desired behavior learned from our training data. This means that from the training and validation phase the weights and biases were adjusted sufficiently to achieve successful classification.

5.4 Extensibility Comparison

The next experiment extends the number of objects by 20 to create a 50 object pattern analysis problem. The results are shown in Table 8. Consistent with previous runs we observed comparable gradient figures with an error that centers around zero. The ROC hovers near the upper left, i.e., near 1.0, with a test confusion showing 96.1% of the data being able to be classified. The result is that for each of the object runs we conclude there is no significant difference between the number of objects with respect to gradient, error, ROC, and test confusion-all key metrics in neural network performance.

Table 8. Pattern Recognition Summary



# of Objects	Epoch	Gradient	Error	ROC	Test Confusion
3	11	0.00018559	convergence around zero	hovers near 1	100%
30	31	0.0071904	convergence around zero	hovers near 1	97.3%
50	35	0.0068315	convergence around zero	hovers near 1	96.1%

5.5 Conclusion

A human inspired approach was used to create a CMAC based neural network capable of pattern recognition. The analysis used a two layer feed forward network to calculate the mean square error. Experiments were run on 3 object, 30 object, and 50 object scenes using edge and optimized SIFT based features as inputs and produced extensible results from 3 to 50 objects with successful classification performance.

The classification results prove that we achieve a high level of pattern recognition from few objects to many objects that ranged from 96.1% to 100%.

HILB pattern recognition model is capable of locally based classification without the need for global models. Global models are reliant solely on what data it has been trained against. Global models typically do not show extensibility from a small number of objects to cluttered scenes with a large number or objects.

Chapter 6: CONCLUSION

6.1 Summary

The goal of the proposed research was to develop a pattern recognition model that consists of a cerebral framework that locally trains, classifies, and recognizes patterns based on local invariant keypoints extracted from image scenes.

This was accomplished based on local information within a scene. We trained via the implementation of an algorithm based on SIFT keypoints and edge data as inputs into a neural network. The network was inspired by the Cerebellar Model Articulation Controller (CMAC) neural network as well as what is perceived to be in the human visual cortex.

SIFT based parameters were optimized to extract invariant keypoints and establish a HILB model of pattern recognition. The HILB model is a framework that links the current global models of pattern recognition to that of human based pattern recognition by being able to process scene information relying solely on localized image data extraction - a departure from current algorithms.

Currently algorithms are task specific and work well if a target object and all its attribute variations are already trained, but struggle to correctly classify new objects if they are not part of an externally trained database. Recognition is difficult due to input sensory data that is not necessarily invariant and susceptible to noise. Our localized model establishes pattern recognition within a scene based on invariant information that will work with any number of objects within a scene.

Chapter 7: REFERENCES

[1] J. S. Albus, "A New Approach to Manipulator Control: The Cerebellar Model Articulation Controller (CMAC)," *Journal of Dynamic Systems, Measurement and Control,* 1975.

[2] J. S. Albus, *Brains, Behavior and Robotics*. New York: BYTE/McGraw-Hill, 1981.

[3] J. S. Albus, "A Theory of Cerebellar Function," *Mathematical Biosciences,* vol. 10, pp. 25-61, 1971.

[4] J. S. Albus, "Outline for a Theory of Intelligence," *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 21, 1971.

[5] J. S. Albus and A. M. Meystel, *Engineering of Mind: An Introduction to the Science of Intelligent Systems*. New York, 2001.

[6] H. Bay, T. Tuytelaars, and L. Gool, "SURF: Speeded Up Robust Features," in *Computer Vision – ECCV 2006*. vol. 3951, A. Leonardis, H. Bischof, and A. Pinz, Eds., ed: Springer Berlin Heidelberg, pp. 404-417, 2006.

[7] A. J. Bell and T. J. Sejnowski, "The "Independent Components" of Natural Scenes are Edge Filters," *Vision Res,* vol. 37, pp. 3327-3338, 1997.

[8] I. Biederman, "Recognition-by-Components: A Theory of Human Image Understanding.," *Psychological Review,* vol. M, pp. 115-147, 1987.

[9] J. Canny, "A Computational Approach to Edge Detection," *IEEE Transaction on Pattern Analysis and Machine Intelligence,* vol. 8, 1986.

[10] F. Cao, Morel, J., and Muse, P., "A Theory of Shape Identification," *Springer Lecture Notes in Mathematics,* p. 257, 2008.

[11] G. Capi, "Multiobjective Evolution of Neural Controllers and Task Complexity," *Robotics, IEEE Transactions on,* vol. 23, pp. 1225-1234, 2007.

[12] W. Cheung and G. Hamarneh, "N-SIFT: N-DIMENSIONAL SCALE INVARIANT FEATURE TRANSFORM FOR MATCHING MEDICAL IMAGES," in *Biomedical Imaging: From Nano to Macro, 2007. ISBI 2007. 4th IEEE International Symposium on*, pp. 720-723, 2007.

[13] S. Chikkerur, T. Serre, C. Tan, and T. Poggio, "What and where: a Bayesian inference theory of attention," *Vision Res,* vol. 50, pp. 2233-47, Oct 28 2010.

[14] C. Cortes and V. Vapnik, "Support-Vector Networks," *Mach. Learn.,* vol. 20, pp. 273-297, 1995.

[15] R. O. Duda, Hart, P.E., *Pattern Classification and Scene Analysis*. New York, NY: John Wiley & Sons, 1973.

[16] H. R. Eghbalnia, Assadi, A., and Townsend, J. , "Symmetry, Features, and Information. Handbook of Geometric Computing: Applications in Pattern Recognition," in *Computer Vision, Neural computing, and Robotics . Part I. ,* ed, pp. 31-65, 2005.

[17] J. Farndon, *Human Body: The Ultimate Guide to How the Body Works*: Dolphin Books, 2006.

[18] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 2, pp. II-264-II-271, 2003.

[19] J. H. Friedman, "Flexible Metric Nearest Neighbor Classification," D. o. S. a. S. L. A. Center, Ed., ed. Stanford University , pp. 1-32, 1994.

[20] M. Grabner, H. Grabner, and H. Bischof, "Fast Approximated SIFT," in *Computer Vision – ACCV 2006*. vol. 3851, P. Narayanan, S. Nayar, and H.-Y. Shum, Eds., ed: Springer Berlin / Heidelberg, pp. 918-927, 2006.

[21] A. K. Jain, Duin, R., Mao, J., "Statistical Pattern Recognition: A Review," *IEEE Transaction On Pattern Analysis and Machine Intelligence,* vol. 22, pp. 4-36, 2000.

[22] K. Jonnalagadda, R. Lumia, G. Starr, and J. Wood, "Viewpoint selection for object reconstruction using only local geometric features," in *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*vol 2,, pp. 2116-2122 , 2003.

[23] S. Kalpakjian and S. R. Schmid, *Manufacturing Engineering and Technology*, 5th ed.: Pearson Education Canada, 2006.

[24] S. W. Kim and B. J. Oommen, "Enhancing prototype reduction schemes with LVQ3-type algorithms," *Pattern Recognition,* vol. 36, pp. 1083-1093.

[25] R. P. Lippmann, "Pattern classification using neural networks," *Communications Magazine, IEEE,* vol. 27, pp. 47-50, 1989.

[26] D. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision,* vol. 60, pp. 91-110, 2004.

[27] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol.2, pp. 1150-1157, 1999.

[28] R. Lumia, "ME 586: L11 Taguchi Design of Experiments," ed University of New Mexico, pp. 1-13, 2008.

[29] D. Maestas, R. Lumia, G. Starr, and J. Wood, "Scale Invariant Feature Transform (SIFT) Parametric Optimization Using Taguchi Design of Experiments," in *Intelligent Robotics and Applications*. vol. 6424, H. Liu, H. Ding, Z. Xiong, and X. Zhu, Eds., ed: Springer Berlin Heidelberg, pp. 630-641, 2010.

[30] D. Marr, "A Theory for Cerebral Cortex," in *Proceedings of the Royal Society of London*, pp. 161-234, 1970.

[31] D. Marr, "A Theory of Cerebellar Cortex," *Journal of Physiology,* vol. 202, pp. 437-470, 1969.

[32] D. Marr, "Philosophical Transactions of the Royal Society of London," in *Biological Sciences*, Simple Memory: A Theory for Archicortex, pp. 23-81.

[33] L. Montesano, M. Lopes, A. Bernardino, and J. Santos-Victor, "Learning Object Affordances: From Sensory--Motor Coordination to Imitation," *Robotics, IEEE Transactions on,* vol. 24, pp. 15-26, 2008.

[34] N. Neave. *Biological Bases of Behaviour*. Available: http://evostudies.org/modules.html

[35] B.-J. Park, W. Pedrycz, and S.-K. Oh, "Polynomial-based radial basis function neural networks (P-RBF NNs) and their application to pattern classification," *Applied Intelligence,* vol. 32, pp. 27-46, 2010.

[36] H. Paugam-Moisy, "Spiking Neuron Networks," ed. IDIAP Research Institute, Martigny, Switzerland and Ecole Polytechnique Fdrale de Lausanne (EPFL), Switzerland, 2006.

[37] S. M. Peeling and R. K. Moore, "Isolated digit recognition experiments using the multi-layer perceptron," *Speech Commun.,* vol. 7, pp. 403-409, 1988.

[38] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 12, pp. 629-639, 1990.

[39] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex," *Nat Neurosci,* vol. 2, pp. 1019-25, Nov 1999.

[40] F. Rosenblatt, "The Perceptron: A Probalistic Model For Information Storage and Organization In The Brain," *Psychological Review,* vol. 65, 1958.

[41] F. Rosenblatt, "The perceptron: A theory of statistical separability in cognitive systems," ed. Buffalo: Cornell Aeronautical Laboratory, Inc. Rep. No. VG-1196-G-1 1958.

[42] A. Rosenfeld and M. Thurston, "Edge and Curve Detection for Visual Scene Analysis," *Computers, IEEE Transactions on,* vol. C-20, pp. 562-569, 1971.

[43] H. A. Rowley, S. Baluja, and T. Kanade, "Neural Network-Based Face Detection," *IEEE Trans. Pattern Anal. Mach. Intell.,* vol. 20, pp. 23-38, 1998.

[44] R. K. Roy, *A primer on the Taguchi method*: Van Nostrand Reinhold, 1990.

[45] T. Serre, L. Wolf, and T. Poggio, "Object Recognition with Features Inspired by Visual Cortex," presented at the Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02, 2005.

[46] C. Siagian and L. Itti, "Biologically Inspired Mobile Robot Vision Localization," *Robotics, IEEE Transactions on,* vol. 25, pp. 861-873, 2009.

[47] G. Taguchi, S. Chowdhury, and Y. Wu, *Taguchi's quality engineering handbook*: John Wiley, 2005.

[48] T. Tuytelaars and K. Mikolajczyk, *Local Invariant Feature Detectors: A Survey*: Now Publishers, 2007.

[49] A. Vedaldi, and Fulkerson, B. . (2008). *VLFeat: An Open and Portable Library of Computer Vision Algorithms*. Available: http://www.vlfeat.org/

[50] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on,* vol.1 pp. I-511-I-518, 2001.

[51] G. Wallis, Rolls, E.T., "A Model of Invariant Object Recognition in the Visual System," *Prog. Neurobiol,* vol. 51, pp. 167-194, 1996.

[52] A. Witkin, "Scale-space filtering," *International Joint Conference on Artificial Intelligence,* pp. 1019-1021, 1983.

[53] T. Ying-li, "Recognizing Action Units for Facial Expression Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 23, pp. 97-115, 2001.

[54] W. Zhang, Yang, H., Samara, D., and Zelinsky, G. , "A Computational Model of Eye Movements during Object Class Detection.," *Advances In Neural Information Processing Systems,* pp. 1609-1616, 2006.

[55] (2003). *Neuroscience: Science of the Brain - An Introduction for Young Students. The British Neuroscience Association.* Available: www.bna.org.uk/publications/brain_sci.html

[56] "JMP, A Business Unit of SAS," Version 9.0.2 ed: SAS Campus Drive. Cary, NC 27513.

[57] "Matlab," Version 7.13, Mathworks, Inc, 2011b.