2005

An investigation of techniques in deformable object recognition

Gavin S. Page

An Investigation of Techniques in Deformable Object Recognition

Gavin S. Page

gsp8334@cs.rit.edu

Master of Science in Computer Science

Golisano College of Computing and Information Sciences
Department of Computer Science
Rochester Institute of Technology
Rochester, NY

13 December 2005

Abstract

The human's innate ability to process information garnered from a visual scene has no parallel in the digital realm. This task is taken for granted in human cognition, but has not been met by a complete digital solution even following years of research. This difficulty can be explained by the shear complexity of the physology of the visual pathway. Although a complete solution has not been created, there are a number of examples of solutions that address parts of the problem.

The recognition of deformable objects is the area addressed in this work. The specific task researched was the recognition of creatures in structured visual scenes. The focus was on developing a set of features which are able to differentiate between target creature classes. The implications of this research lie in ecoinformatics and field biology with the automated collection and annotation of biological data. The thesis will present a survey of the current literature addressing techniques which have been used to solve similar problems. An algorithm to perform the recognition will be presented and the results discussed. Finally, potential areas for improvement will be described.

Acknowledgements

I would like to thank my committee, especial Dr. Gaborski for his guidance throughout the course of this thesis; everyone in the lab who provided ideas and encouragement, especially Dan, John, Karthik, and Raja; Jim, Leslie, and Nick who helped with my proofreading; and, of course, my family.

Contents

List of Figures

Chapter 1

Introduction

As human beings we can visually identify objects with relative ease. In contrast, teaching our computers to 'see' and recognize objects is a very difficult task. Information from visual scenes is intuitively extracted and processed by humans. There is no such simple solution in the digital realm. Complex algorithms are needed to obtain the target data and derive and represent its descriptive features. These features will be the basis for the operation which distinguishes object types. These algorithms typically follow 3 high-level steps:

1) Segment Image
2) Extract Features
3) Perform Recognition

Depending on the nature of the system one or more of the three steps will take some precedence over the others with respect to complexity. This work focuses on extraction of feature data.

In the first step of the general system, the target object is extracted from the background. The complexity of this step is related to the complexity of the background. When an object is in a natural or cluttered scene, segmentation will exist as an important part of the end algorithm. In contrast, given a strctured and simple scene, segmentation may be a relatively minor component of the overall process.

When extracting features a host of methods can be applied to the segmented objects. The decision regarding which features to use is dependent on the direction which the developer wishes to proceed with implementation. It is possible to base the system on features which describe the shape of the object. These can be both at the macro level and involve the whole object or can be based on a parts based approach where smaller regions of the object are compared. Additionally, features based on the arrangement of pixels can be utilized. These features are classified into the heading of texture. Evaluating the pixel arrangements can be done with a multitude of techniques ranging from statistical measures to wavelet transforms. Some of these techniques will be discussed beginning in Section 2.1.3 on page 11.

The final step involves processing those features and defining their membership in a target class. This is most suitably done using a machine learning paradigm. By utilizing machine learning the system will be able to adapt and respond to the features which are responsible for differentiating the object classes.

1.1. Deformable Object Recognition

Due to the inherent complexity of recognizing objects of non-static shape (deformable objects), human beings are clearly in front of machine based solutions in recognition tasks. This has not limited the desire to create systems that can mimic some human processes in recognition. In this increasingly complex society there is a continual push toward unloading tasks onto computers. This conversion to automation is common for mundane tasks that can be solved algorithmically, for tasks which may put a person at risk, and for tasks which involve an abundance of data to process.

NatureSpy is a project which seeks to assist a biologist in the collection of data from the field. The system is to exist in 3 parts: A visual capture system, an automated recognition system, and database of creatures. It is designed to help resolve some of

the situations discussed previously as reasons for automation. A biologist has to spend significant amounts of time, often in remote or inconvenient locations in order to collect data. While in the field the scientist could be exposed to any number of dangers from the environment or from other unforeseen situations. Additionally, large amounts of data are likely to be collected during a field excursion all of which must be processed and compiled into relevent output. Furthermore, the impact the human has on the behavior of native fauna is unknown. Reducing the human presence in a natural setting could glean new insight into nature behind the scenes. Each of these reasons validates the NatureSpy project.

The contribution outlined in this thesis is limited to the design of the creature classification system. The concept will be illustrated by classifying a small subset of potential objects. This is similar to the work undertaken by Walther et al. [**2**]. While there is no actual recognition of the objects in the scenes, the concept of creature-centric annotation of scenes as 'boring' or 'interesting' is an example of offloading a task typically performed by a person to an algorithmic process. Other work dealing with the detection of animals in visual scenes has been done by Burghardt [**3**] and by Kalafatic [**4**]. These however were centered on the problem of tracking the creatures. This thesis differs from the work of these researchers by annotating creatures in the visual scenes into 3 distinct classes:

- Salamanders
- Snakes
- Toads

For each class of creature, except for snakes, multiple species of each class were collected and classified by the system. The specifics of the creatures involved is discussed further in Section 3.1.2 on page 28. Furthermore, for completeness, it was designed to differentiate natural noise from actual creatures. For this system noise is defined as the set of objects not among the 3 target classes. For experimental purposes a separate class of 'Nature'

objects was utilized. This included natural objects that would likely be found in a field capture system such as leaves, sticks, pinecones, etc.

In the remainder of this document techniques available to leverage a creature classification will be presented along with a system designed to perform the classificiation between the 3 creature types.

Chapter 2

Previous Research

2.1. Object Features

The classification of deformable objects is an important aspect of computer vision and image processing. Features are the most important aspect of this classification. Without robust features the classification cannot be successful. Deformable objects complicate the extraction of features by creating a range of values over which the features will exist. It is very likely that objects, particularly natural objects, will not maintain the same appearance over the course of that object's lifetime. This is especially true of objects that undergo locomotion. The kinematics of the body will cause multiple local deformations across the periphery of the object. These must be expected by the system in order to correctly classify these creatures. The first step to recognizing these deformations is utilizing advanced techniques to represent the salient features of each object.

2.1.1. Color-Based Features. A feature used in work with these sort of conditions (limited objects on a distinctive background) is color. Color indexing has been used since the seminal work of Swain and Ballard in [**5**] where histograms and the distribution of image colors were compared. In this work they showed color has several useful properties when dealing with deformable objects. These include invariance to typical problematic elements such as scale, rotation, and partial occlusion. They also showed the major pitfall of using color as a classifier is its high dependence on illumination.

Color, when represented in the RGB color space is defined by a triple of red, green, and blue values at each pixel location. Representing each pixel value in this way reflects a functional mapping of the intensity as well as the chrominance [**6**]. By combining both luminance and chrominance into one value, pixels which should be chromatically alike are represented by different values due to variations in illumination. This can be mitigated by utilizing a color space which separates the luminance and chrominance components of each pixel. Some examples of these are listed by Ford and Roberts in [**7**] and include YIQ, YUV, YCbCr, and YCC. This has the added benefit of reducing the dimensionality of the problem space from 3 to 2 as the luminance component of each pixel can be discarded. Overall, the reduced dimensionality will speed any computations performed.

Terrillion discusses in [**8**] that normalizing the color space yields the best segmentation results with respect to face detection applications. The process of normalization reduces the sensitivity of the system to illumination changes. Terrillon further hypothesisizes that this method can be extended to other applications requiring color image segmentation. By performing these operations, variance in surface color caused by positional changes affecting incident light levels can be reduced [**9**] providing a more consistent working image representation. That consistency is important for representing images cleanly.

Additionally algorithms have been developed to reduce the degree to which illumination is a factor by reducing the colors in the image by a statistical quantity. This increases the viability of using color as a feature in object recognition and classification problems. These color compensation methods are generally effective in conditions where illumination remains close to a standard quantity [**10**]. There are several algorithms which have been developed for this task. They include Grey World (GW), white-patch retinex, neural net, 2D gamut-constraint, and 3D gamut-constraint. As reported in [**11**] they each have varying levels of success when tested using Swain and Ballard's color indexing method. In the spirit of Swain, these algorithms are utilized in systems where a distinctive chromatic component is required. In [**12**], Chen and Grecos employ the GW

algorithm to leverage a skin region detector. The GW is implemented by assuming that the spatial average of light leaving a surface will be equal to the light incident upon it. This is expressed in equation 1 where S_c is the resultant scale factor for a channel, C_{std} is the standard mean gray value and C_{avg} is the mean value, each for a specific channel.

$$S_C = \frac{C_{std}}{C_{avg}} \tag{1}$$

In the case of this problem color is still not able to be utilized due to subject-based limitations. It is common in nature for creatures to have variable colors or patterns based on phenotype [**13**], diet [**14**], or time of year [**15**]. Due to this inconsistency of coloration it is not possible to depend on creature coloration for anything more than a segmentation medium when compared to the structure background.

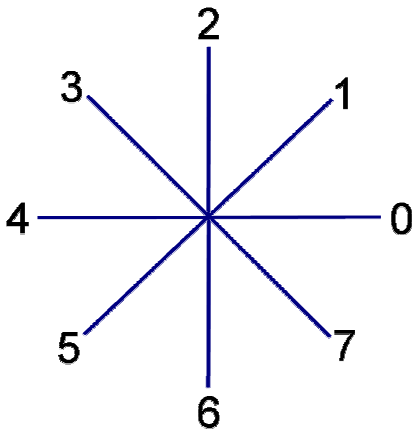
2.1.2. Shape-Based Features. The comparision of shape-based features offers a powerful means of relating and differentiating different classes of objects. While many objects are easily recognizable to most people based on their shapes, this sort of general recognition is not easy to implement in an algorithm. There must be well defined features which can be algorithmically utilized to compare distinctly measurable features on an object to other objects. The most direct means of evaluating shapes is using a point to point comparison as discussed in the original work of Thompson [**16**] or, relatedly, through a direct comparison of an object's bounding contours. It should be noted that these methods are unlikely to work with anything beyond trivial scenarios as they are highly specific and the features must be subjected to multiple transforms in cases of rotation and dilation.

When performing shape-based comparisons of objects, point to point comparisons are too specific to relate anything but exact matches of a source to a sink object. A more advanced and less specific method designed by Freeman [**17**] [**18**] and described in [**19**], is to abstract the curve and represent it by a sequence of line segments on a defined geometric overlay. This method reduces the specificity of the representation and cuts the

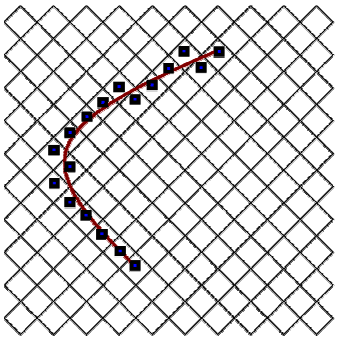
storage overhead inherent with representing individual points. By *linking* segments of the overlay a maximal number of direction changes is represented by the shape instead of maintaining a maximal representation of the entire set of points of the image [**20**]. This representation conversion simplifies contour evaluation to a traversal of the resultant *chain code* making it simple to interpret large deformations in the shape of the object by parsing for deviations in code values. This is illustrated in Figure 2.1. The figure shows how a curve is generalized using Grid-Intersect Quantization. Other methods include Circular and Square Quantization. The methods are named based on geometric overly on which the points are inscribed.

Following the generalization of the representation it is possible to indicate the approximate shape of the objects of interest with the *convex hull*. This by definition is the minimum fit polygon enclosing the object [**21**]. Fitting to the object in this fashion is more general than direct fitting using the contour information. Additionally, it is more specific than the bounding box which simply encloses the entirety of the object without any reflection of the shape. The key disadvantage to using the convex hull is the obvious loss of data caused by this gross abstraction. An image demonstrating a convex hull and a bounding box is in Figure 2.2. While the goal in shape representation is abstraction it is important to not eliminate critical information which will be useful in performing the differentiation.

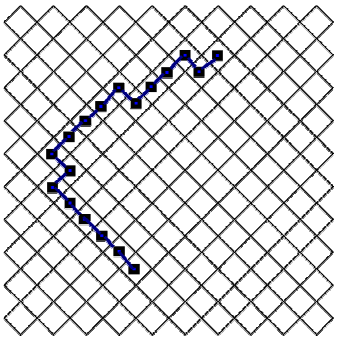
When large deformations are present or there are points of articulation in the target objects, more complex feature representations must be utilized in order to preserve the meaning of the underlying shape information. In [**22**] and [**23**], Cohen and Wu present the idea of utilizing curvature to represent important areas in natural objects. Cohen declares curvatures very useful in matching as they often have a great deal of anatomical significance. Wu takes this further by relating the importance of anatomical structures to recognition of human-like figures. He details that the shape model must have representations which take into account the large, local shape deformations caused by body postures, view changes, and chromatic and textural changes. The literature points to



(a) Chain Code



(b) Smooth Curve



(c) Curve Representation

Figure 2.1. (a) Represents the 8 possible bits available to the Grid Intersect style of chain code representation. (b) Displays a sample curve with the utilized grids marked. (c) Shows the abstraction given by the Chain Code representation. For this example the chain code is *33333131111711171*.

several methods designed to compensate for the high degrees of freedom required of the local shape deformations.

One such method is based on active contours or snakes [**24**] [**25**] [**26**]. An object's contour is defined in [**27**] as a set of radial distance vectors which emanate from a fixed point called the *anchor point*. In the active contour method an initial contour is first selected. It is then adjusted by internal and external forces which help in conforming the contour to the actual boundaries of the image. By utilizing this dynamic apparatus to

Figure 2.2. Shows an example of a bounding box (white) and a convex hull (red).

describe the shape of the target object it is possible to maintain boundary concavities [**25**] this translates into better representation of body deformities. An example of the active contour representation of an object is shown in Figure 2.3.

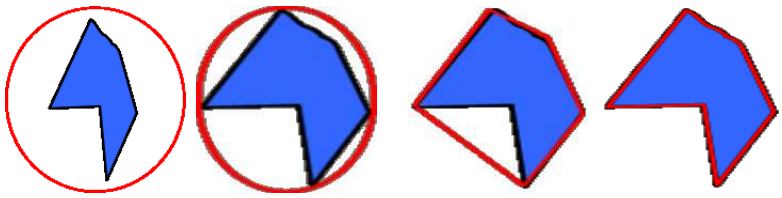


Figure 2.3. This figure illustrates the process undertaken when determining the active contour of an object.

Another less dynamic means of representing contour information is through a concavity graph as shown in work done by Badawy and Kamel [**1**]. The graph is directed

and shows the relationships between objects, concavities, holes, multiple objects, and multiple holes by connecting the nodes of each type. An example of a concavity graph is shown in Figure 2.4. This type of structure is useful in establishing relationships between local deformations such as those created by appendages. This information could be used to relate these appendages in various states to the body. Prior knowledge of shape is required in the work of Chang and Liu [**28**] who use shape information applied to an elastic grid. A match is determined based on energy minimization.

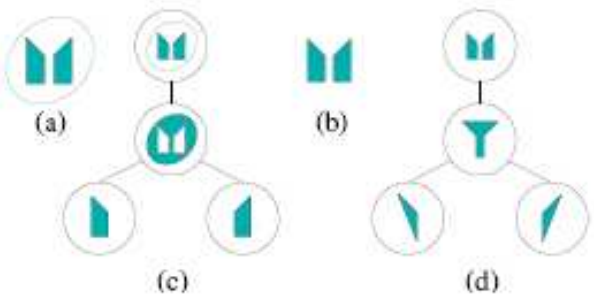


Figure 2.4. Concavity trees and objects that include other objects. The concavity trees in (c) and (d) of the images in (a) and (b), respectively have the same structure. [**1**]

In addition to establishing features to be used in the object classification system contour information can be used in conjunction with heuristics to reject spurious extrema. This is done in [**29**] to reject inconsistencies in words, but can also be leveraged to remove objects from images which could not be a reasonable match for any of the target creatures.

2.1.3. Texture and Frequency-Based Features. Texture is a distinctive, but difficult to define characteristic of image objects. A reasonable, accepted definition is that texture is a function of the spatial variation of pixel intensities [**30**]. These spatial variations are often repetitious across an object or specific regions of an object and exist as a function of resolution. This means a texture is represented differently across different levels of zoom. This can be illustrated by the examples shown in Figure 2.5. In the case of the brick wall, the default resolution is a high level texture of a repeated brick pattern.

Zooming in on a specific region will cause a change in the respresentative pattern. The spatial variations aid in differentiating objects by shape of the texture region and by providing a separate set of quantitative features.

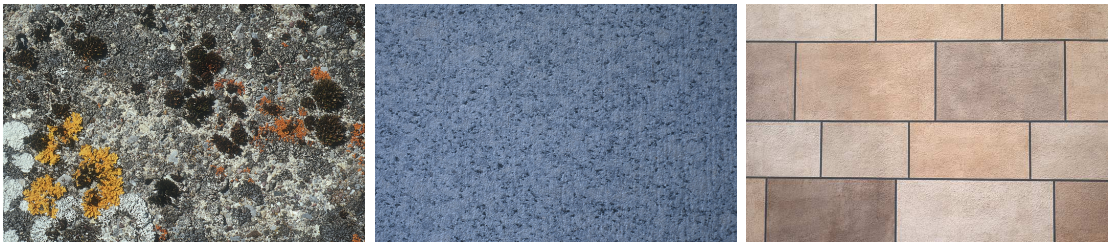


Figure 2.5. This figure illustrates some examples of texture. Notice that each has common, repeated elements. Image Source: http://www.free-pictures-photos.com/texture/

A significant amount of research has been performed with respect to imaging in the medical field. Methods ranging from statistical analysis to structural derivation of texture have been explored to identify and classify liver lesions, neural structures, burns, and others. Imaging in medicine is of critical importantance as it reduces the need for invasive surgery and increases the speed of diagnosis. The resulting image from a medical imaging device (X-Ray, CT, MRI, etc.) is likely to be difficult to interpret due to minimal variations in contrast between structures; making both human and automated analysis difficult. Pitiot describes this situation in [**31**] where he is attempting to identify white matter structures in the brain from a MRI. The target structures feature similar intensity values, therefore, classification had to be based on the fiber orientation, or texture. By applying methods to extract empirical texture data from the images a common arrangement of pixels can be established which can be related using a classifier (discussed in 2.2).

The major difficulty in utilizing texture is developing a set of meaningful features that distinctively represents an image's objects [**32**]. Statistical methods quantify the spatial distribution of gray level values. These come in two general types *first order* and *second order*. The first order statistics are applied directly to the gray level histogram;

specifically, a histogram representation of a region of interest (ROI). Any operations involving the first order statistics are applied directly to the gray level values at each pixel. The second order statistics are performed on a Co-Occurrence (also called Spatial Gray Level Dependence) matrix. This structure is created by counting pairs of pixels at a given position in the image's ROI. Normalizing this matrix allows it to be treated as a probability density function, thereby allowing common statistical measures to be applied. This is discussed in [**33**] and [**34**] and formally described in equation 2.

$$P_d(i,j) = |\{((r,s),(t,v)) : I(r,s) = i, I(t,v) = j\}| \tag{2}$$

Following the extraction of the histogram and construction of the matrix, common statistical methods are applied to describe the shape of the curve. Some first order methods are dicussed in Khan [**35**] and include standard deviation (Equation 3), skew (Equation 4), and kurtosis (Equation 5). With respect to the gray level histogram the standard deviation is a measure of the gray level contrast, the skew quantifies the asymmetrical nature of the probability density, and kurtosis measures the peakedness or flatness of the curve.

$$\delta = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{3}$$

$$\hat{\gamma}_3 = \frac{\sum_{i=1}^{N}(x_i - \hat{m})^3}{(N-1)\hat{\sigma}^3} \tag{4}$$

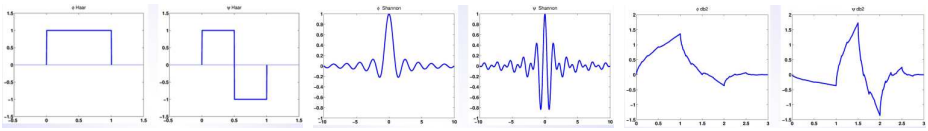
$$\hat{\gamma}_4 = \frac{\sum_{i=1}^{N}(x_i - \hat{m})^4}{(N-1)\hat{\sigma}^4} - 3 \tag{5}$$

Christoyianni discusses the usage of numerous second order statistical measures in his work detecting abnormal features in mammography [**34**]. Among these are the angular second moment (Equation 6) measure of homogeneity for the image. These feature are part of the 14 features originally proposed by Haralick when describing texture [**36**].

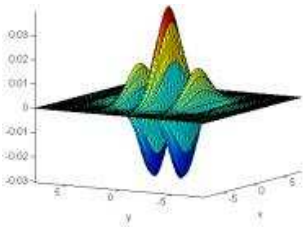
$$AngularSecondMoment = \sum_{i,j} p(i,j)^2 \tag{6}$$

In addition to generalizing texture information using statistical measures it is possible to obtain features using signal analysis techniques. The work of Fourier in the 1800s brought about the representation of functions using the superposition of other functions, namely sine and cosine. These conceptually are refered to as *basis functions* and are also used in wavelet processing. Wavelets are an expansion on the idea of utilizing basis functions in image signal analysis devised by Mallat [**37**]. The major benefit of using wavelets when compared to Fourier analysis lies in their localization in space and frequency as opposed to just frequency [**38**]. More specifically, wavelets describe a limited range of frequencies in a limited range of space, making it possible to use them to describe discrete image regions [**39**].

Another benefit of using wavelets is the multiple basis functions available for use. Since the expansion of the field there have been numerous functions which can be applied depending on the goals of the system. These include Daubechies, Haar, Marr, Sinc, and Gabor wavelets. These functions are applied to a signal attempting to approximate it. During the application the wavelets are scaled and translated to perform the analysis. Since each point in the analysis is a derivation of the original basis function or 'mother wavelet' multi-resolution analysis is possible. Some examples of 'mother wavelets' are shown in Figure 2.6.



(a) Haar Wavelet (b) Sinc Wavelet (c) Daubechies Wavelet



(d) Gabor Wavelet

Figure 2.6. This figure illustrates a few wavelet basis functions. Images (a-c) are from http://cnx.rice.edu/content/m11150/latest/. Image (d) is from http://www.cic.unb.br/docentes/juliana/TesePhD/Chapter%205.html

In the case of a discrete wavelet transform, a set of coefficients are determined which linearly describe the set of wavelets approximating the function. These coefficients become the feature with which the target is described. The result of a decomposition is shown in Figure 2.7. These can be used as a comparison of local edge information or the coefficients themselves can be utilized as a feature vector.

The Gabor wavelet has attracted significant interest with respect to extracting texture as it has been determined that it approximates the function of receptive fields of cortical cells [**40**]. It is a Gaussian curve modulated with a sinusoid and is used in work such as that done by Zhao et al. involving correlating texture images with Gabor elementary functions using CT liver images in [**41**] and in [**42**] analyzing bone texture.

Although wavelets have entered widespread use as of late, standard Fourier analysis is still used in representing texture information in images. This is done in [**43**] by extracting and comparing the Fourier coefficients of neighborhoods of intensity pixels; much the same as is done with wavelets.

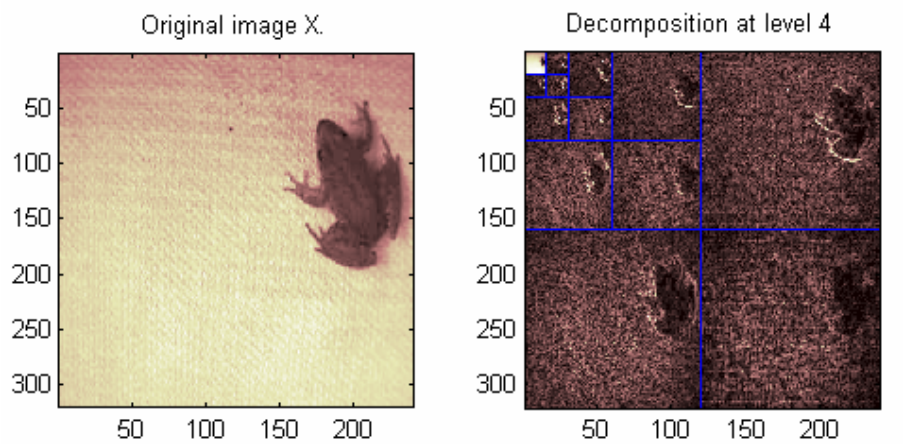


Figure 2.7. Illustrates a sample wavelet decomposition; Specifically a 4th level decomposition using a Debauches wavelet. The image is the coefficients of the 3 levels of detail at each level (horizontal, diagonal, and vertical) and the approximation.

It is also possible to utilize a feature invariant to a subset of geometric transformations such as the Fourier-Mellin Transform (FMT). The FMT is advantageous as it maps the cartesian coordinates of an image into Logarithmic-Polar (LP) coordinates, converting rotation and scale into translational shifts [**44**]. When coupled with a further Fourier transform of the LP-image, scale and rotation invariant features are introduced. The features are not ideal as they are dependent on a fixed point in the original cartesian domain meaning the approach is not position-invariant. Furthermore there is a discontinuity between the effective degree of information within expanding radii of the centroid. An inverse relationship exists in the integral to explain the transform making central features higher weight than those at the periphery [**45**]. The standard Fourier-Mellin transform of f is shown as equation 7 where f is a gray-level image region of real numbers.

$$\forall(k,v) \in \mathbb{Z}x\mathbb{R}, \mathcal{M}_f(k,v) = \frac{1}{2\pi}\int_0^{\infty}\int_0^{2\pi} f(r,\Theta)r^{-iv}e^{-ik\Theta}d\Theta\frac{dr}{r} \tag{7}$$

After the transform has been calculated the results may be concatenated onto a vector of features which can be compared to other vectors based on the phase of the features. Several groups have researched improving the FMT such as Gotze et al. [**44**] who applied a local fast-FMT to a cartesian grid to enable fast computation of features at each image location. Derrode also looked to improve the accuracy and efficiency of the FMT by testing variations in the FMT (Analytical FMT and Inverse Transform) and applying a gray-level shape and shift theorem which converts the original similarity transformation into a multiplication in the Fourier-Mellin domain [**45**].

2.2. Pattern Recognition

The problem of pattern recognition was defined in statistical terms in the 1930's by R.A. Fisher [**46**]. He considered one of the first formulations of the complex mathematical process of recognition; a process which humans are able to do virtually effortlessly. The goal of any pattern recognition engine is to define each element in the *pattern space* as belonging to a particular region of the space called a *class*. Each class corresponds to an end target for the system. In a typical system, each class will be delimited by a complicated boundary called the *decision surface* [**47**]. As will be discussed further in Section 3.2.4 on page 38 the pattern space in this problem is populated with a vector of shape and texture based features obtained from each object.

Numerous devices have been created which divide the pattern space and allow the classification of feature vectors to take place. These include rule based classifiers [**48**] [**49**] [**50**], Support Vector Machines (SVM) [**51**] [**52**] [**53**] [**54**] [**55**], and neural networks [**56**] [**57**] [**58**]. The process by which each image object is automatically annotated varies between these techniques.

2.2.1. Rule Based Techniques. In the application of rule based techniques a series of 'If, Then' style clauses branch out from a root level node and terminate at a result.

By first extracting low level features from a scene and then applying those features to an *a priori* set of rules, it is possible, as was done in [**50**], to develop a high-level semantic decision about a scene. A major setback of this approach is structuring and generating the rule set. The rules require specific targeting for a given application as each application will have target results. Designing an initial rule base is a field among itself as shown by Bischof in [**48**].

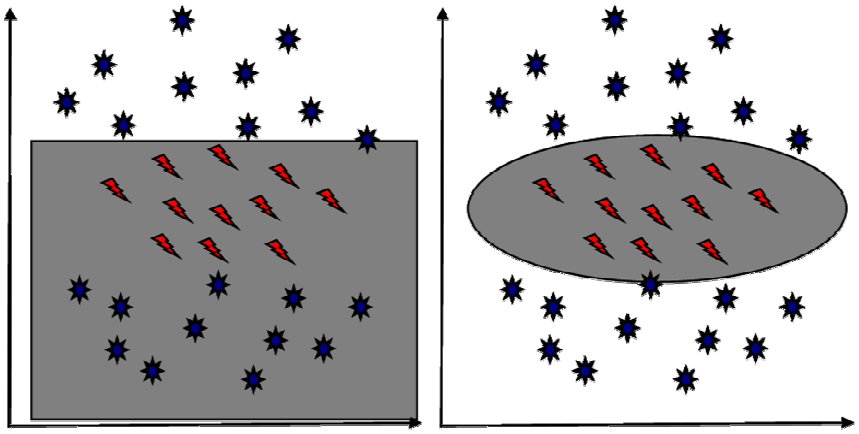
In his work Bischof discusses a system of Conditional Rule Generation (CGR) which builds upon a set of structural pattern information to classify a set of training information. CGR is based on the concept of clustering unary and binary elements of extracted patterns. The clusters define decision boundaries for the system's rules based on a given feature being unique or not. Unique features define a rule while non-unique clusters are evaluated for the new feature space until all of the clusters are resolved or a maximum rule threshold is reached. Following the initial generation of the rule base it is possible to refine it using entropy-based splitting procedures. A suitable boundary is considered found when a minimal state of energy at the point of the partitioning.

Zhou has also looked into the problem of rule refinement. In [**49**] he attempts to remedy the common weaknesses in classifiers (initial classifier chain generation and initial system parameter setting) by employing a Market-based Rule Learning System (MBRL). MBRL is a 3 layer system which verifies the problem can be solved in the first layer, evaluates the performance of the solver in the second layer, and the third layer searches for new ways the learning system may perform its function using a genetic algorithm. This third layer makes it possible for the system to gain experience by adjusting the weights on existing rules and to create new rules .

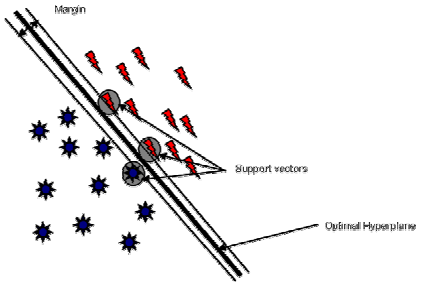
2.2.2. Support Vector Machines. The concept of a SVM fits into the class of solutions broadly referred to as *kernel methods*. As outlined in [**47**] kernel machines are motivated by Cover's theorem which states that a pattern recognition problem can become linearly separable in pattern space of sufficiently high dimension. By utilizing

nonlinear transformations of the pattern space it is possible to select a weight vector in the new feature space Φ that will classify the problem with a minimum error. The kernel machine is advantageous by separating the number of inputs of the machine from the size of the input space; meaning that if the pattern to feature mapping is done using a symmetric function that obeys the Mercer condition, weight calculations in the discriminator function can be computed without ever solving the problem in Φ [**59**].

The ability of an SVM to follow Cover's theorem and make a problem linearly separable makes it possible to utilize many other machine learning methodologies including Multi-Layer Perceptrions (MLP) and Radial Basis Functions at a simpler level. This makes the core methodology highly flexible for target applications. The separation created by discovery of the optimal division between classes, or the *hyperplane*, creates a larger *margin* between neurons when compared to traditional MLP training. This margin is an equidistant space from the *support vectors*, the points nearest the hyperplane as demonstrated in Figure 2.8. Having this margin reduces the error in the test set that generally occurs during MLP training. During MLP training the decision boundary is typically very close to the final point in the training set. Employing the margin system creates a wider region with which to minimize the error. Furthermore, since the mapping and classification are performed using a function, the results of the SVM are reproduceable. Reproduceability is not possible using standard neural networks [**51**] because of the random nature of the training.



(a) Linear (b) Non-Linear



(c) Optimal Hyperplane

Figure 2.8. Subfigure (a) displays the problem that can come about from trying to classify data of a non-linear fit linearly. (b) shows the result after a quadratic fit. Subfigure (c) illustrates how a SVM divides binary classes of data using the optimal hyperplane. Notice that it is equidistant between the support vectors of each class.

Since the work performed by Vapnik [**59**] there has been an increasing interest in utilizing Support Vector Machines in a variety of pattern recognition problems. They have been applied in such tasks as text identification [**54**] [**53**], gene expression data [**55**], and image classification [**52**].

SVMs are deemed highly useful in image classification problems due to the nature of the features typically available during these problems. Images are often diversified

with features that are not linearly separable. In [**52**] Goh shows empirically that SVMs coupled with error reduction were able to increase the class prediction accuracy across each of their binary classifiers.

2.2.3. Neural Networks. Neural networks are a commonly used method in the field of pattern recognition. Their functionality is based on a collection of discrete processing nodes which take a number of inputs and perform basic computations which are combined into a single output [**60**]. These discrete processing elements are designed to mimic the functionality of biological learning. At the initialization of a new neural network, each node is given a random weight. The adjusting of these weights is then dependent on which type of neural network is being used and the training algorithm. For a network architecture that utilizes *supervised learning*, such as a multi-layer perceptron, weights are adjusted to coincide with a set of values for the target classes. A common algorithm for this training is *back-propagation*. *Unsupervised learning* techniques such as those designed by Kohonen [**61**] do not rely on a training set and instead organize neural weights based on the input data.

The general structure of a supervised neural network can be broken down into 3 basic layers. The first is the input layer. This layer is a one-to-one mapping of the quantitative value of each feature to a node which accepts the value. The second, or hidden layer features the collection of processing structures, or neurons, which are trained to produce the correct result in the 3rd or output layer. A neural network is illustrated in Figure 2.9.

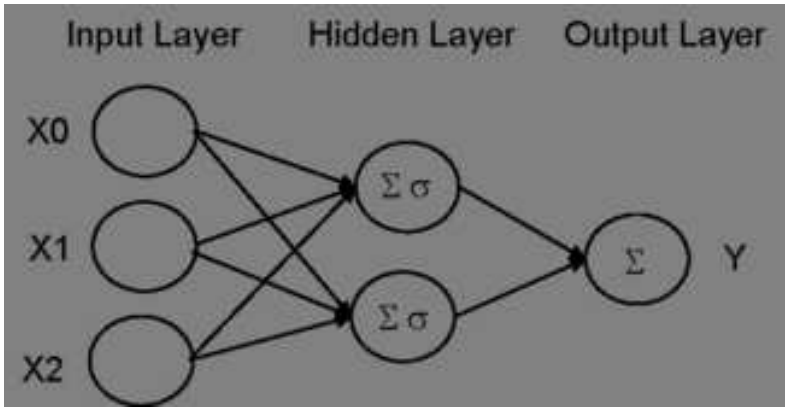


Figure 2.9. This figure shows a feedforward neural network. It has 3 inputs connected to a hidden layer of 2 neurons. The neurons sum the inputs and have a sigmoid transfer function. When the threshold on the transfer function is met the value is passed to the neuron in the output layer.

The benefit of using neural networks is in their adaptability and their ability to learn. A neural network is able to generalize a set of data in such a way that it is possible to recognize an object as being part of a particular class without any specific previous knowledge of that object. This is a clear benefit for pattern recognition [**62**]. Additionally it is possible to classify features using non-linear boundaries with the addition of neurons; another important characteristic given the varied and non-continuous nature of the features extracted from images.

The major drawback with using neural networks is their dependance on the initial or training data. The classification performance will only be as good as the training set. If the system is exposed to an unknown quantity it will not perform successful classification. This means that for every new feature or target object the system will have to be retrained or exposed to a new set of samples so that the weights can be adjusted. Furthermore, since the beginning weights are random, training performance will likely not be repeated each time. This will cause a different result each time the program is retrained and run.

Neural networks have a common usage across the domain of image classification. In [**63**] Khalil and Bayoumi utilize a multilayer, feedforward neural network to perform 2D object recognition using the dyadic wavelet transform. In both [**64**] and [**34**] neural nets are applied to detecting abnormalities in medical imaging. In Turner's work a neural network is used to determine regional image transformations used in document enhancement [**56**]. Finally, in [**57**] and [**58**] neural networks are applied to semantically desribe images; a process similar to the goal of this work where the end result could generate key words for a searchable image library.

Chapter 3

Methods and Implementation

This chapter describes the methods developed in the course of this research project, as well as implementation details for the simulations that produced the results given in Chapter 4 on page 42.

3.1. Data Collection

3.1.1. Evaluation of Equipment. The end goal of this work is be able to field-deploy an image capture system which will transmit data back to a centralized location for further processing and analysis. The initial camera, a D-Link model DCS-2100+ was selected because of the ease it provided these reporting features. The camera was able to function on both a wired and wireless network to automatically email or upload captured images; along with its relatively low cost and motion capture abilities this camera matched the specifications of the project at the time.

This proved to not suit the image processing needs in the early points of the experiment. The camera's depth of field and ability to focus on subject creatures was limited making it difficult to use anything but the simplest of algorithms. It was determined that it would be best to leave more algorithmic options available and utilize a different camera for image capture. While data from this camera turned out to not be useful in the processing aspect it did provide experimental verification of setup.The configuration of the capture system is outlined in Figure 3.1.

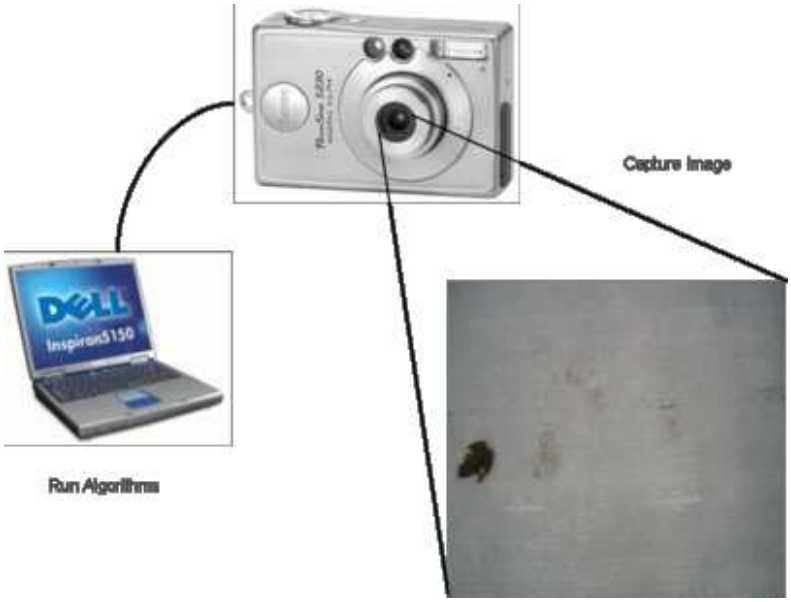


Figure 3.1. Figure shows the arrangement of hardware used to perform the image capture. The system is arranged in 3 parts: The static background over which the creatures pass, the fixed-position camera, and the computer used to do the processing.

A point of discussion of the classification of the creatures was their ideal orientation. Numerous images at different distances and orientations were captured in an attempt to capture the position where each creature was best represented. It is necessary to capture the subject of the image in as uniform an environment as possible. This was done to optimize the amount of surface area exposed in the image. A greater area provides an increased amount of data from which to extract features.

Each creature was placed within an environment that remained uniform across image capture sessions. This included the background which will be discussed later and the positioning of the other system elements. Consistent measurements were maintaned governing the positioning of the camera and its distance from each creature. This standard measure provided a uniform scale across the experimental images.

It was determined that a structured background would be used. This would make segmentation a simpler task and allow the focus to be on experimentation with features for the classifier. Backgrounds were tested of various materials and colors to identify which would be the ideal surface for creature segmentation while still providing a comfortable setting for wild animals. The goal for the end system is to be able to perform observations without having an overly intrusive human element distrupting the normal habits of fauna. This made material selection important. It had to be something that an animal would feel comfortable with in the wild, while still fitting with the goal of a high-contrast background. The material also had to be durable and able to withstand the abuse characteristic of being outdoors. To fill this role standard outdoor carpet of different colors was selected.

The mats provided two surfaces, a textured carpet and a rubber backing, with which to experiment. Evaluation was done to determined which material and color combination worked the best given the available options. The options included:

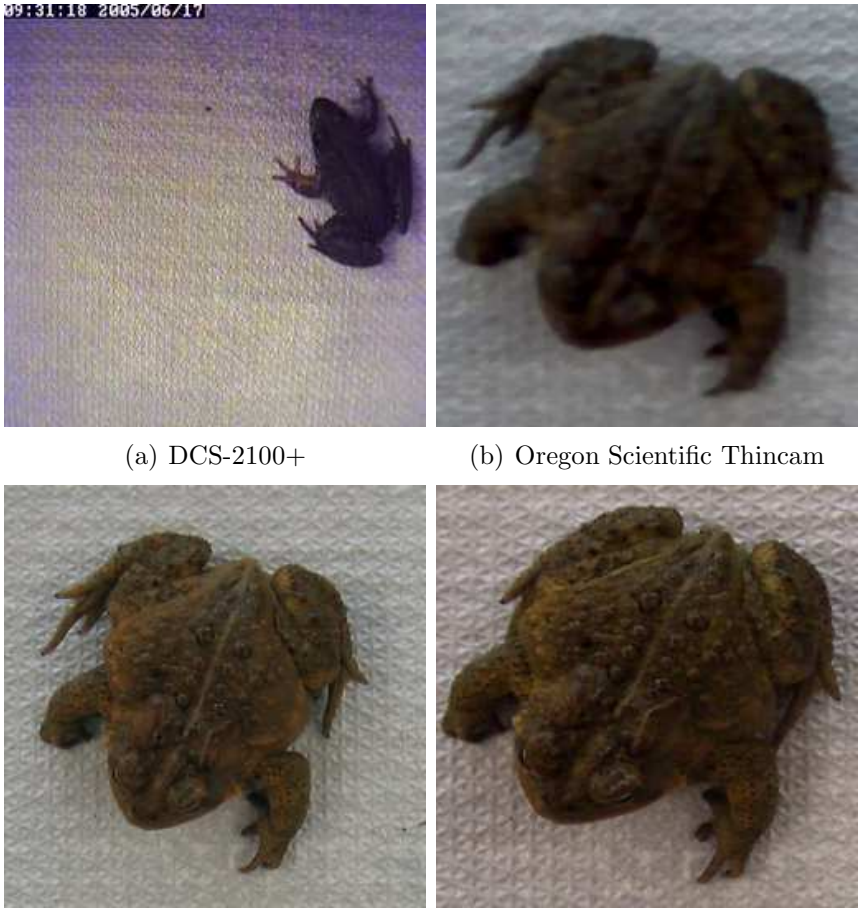
- Alternating Gray/White Carpet
- Solid Gray Carpet
- Solid Green Carpet
- Solid Black Rubber
- Solid White Rubber

The best background proved to be white painted rubber. This is obvious from an image processing standpoint as it provides a high contrast surface to segment a typically dark creature. The surface also has a distinctive texture which should prevent falsely classifing a creature from a backgroud segment given texture based features. From a biological standpoint the creatures did not seem to have a problem navigating across the surface.

In order to select an optimal level of detail to be captured several cameras with different features were evaluated. The cameras utilized were the 1.3 Megapixel Oregon Scientific Thincam, the 3.2 Megapixel Canon Powershot S230, and the 5.0 Megapixel

Pentax Optio S5i. The Oregon Scientific offering captured images only fractionally better than the D-Link camera. The resultant images did have increased color representation, but not of a suitable factor to justify its use. Furthermore, the camera had poor focusing abilities, reducing its ability to capture fine featured texture information information.

The Canon and Pentax cameras showed significant increases in quality in their captured images. The representation of color and texture increased visibly over the previously tested cameras. The focus of these cameras was also an improvement. It was determined that the Canon would be the best to use as the primary image capture device. The principle goal of this testing was to determine a minimally satisfactory camera for capturing images used to differentiate creature type. While the testing done was not extensive it revealed that 3.2 MegaPixels provided suitable image depth and clarity for extraction of features. A comparison of images is shown in Figure 3.2.



(a) DCS-2100+ (b) Oregon Scientific Thincam

(c) Canon Powershot S230 (d) Pentax Optio S5i

Figure 3.2. The above images illustrate the differences in capture quality between the 4 test cameras. Note: (a) is an image of *Rana clamitans* while (b-d) are of *Bufo americanus*. The difference is due to the creatures available at different points in the experiment.

Camera position was also evaluated. Sample images were taken from several angles around the test creatures in order to determine the angle which captures the maximal amount of useful information from the creature. The obvious decision was the top-down view. From this position significant information can be collected including the position of the appendages, locomotive deformations, and dorsal patterns.

3.1.2. Collection of Creature Images. Creatures were encouraged to traverse the mat in the field of view of the camera. Multiple images of each creature were taken to attempt to capture each creature in a variety of poses both for training and testing purposes. Dealing with live animals was problematic in cases of obtaining various species during a given time of year and with guiding them into the field of view.

Figure 3.3 lists the creatures utilized in this work. Each was captured by Dr. Paul Shipman[1]. In addition to the Latin and common names, the capture site is listed. This will be helpful in future advances of the system which could possibly involve detection of features that may vary between regions.

Latin Name	Common Name	Collection Site(s)	Images
Ambystoma maculatum	Spotted Salamander	South Hemlock	40
Bufo americanus	American Toad	Oatka Creek	36
Desmognathus fuscus	Northern Dusky Salamanders	Oatka Creek	5
Eurcyea bislineata	Northern Two Lined Salamander	South Hemlock/Oatka Creek	23
Plethodon cinereus	Red Backed Salamander	Oatka Creek	14
Rana clamitans	Green Frog	South Hemlock/Black Creek/Oatka Creek	35
Rana palustris	Pickerel Frog	South Hemlock	64
Thamnophis sirtalis	Common Gartersnake	Oatka Creek	6

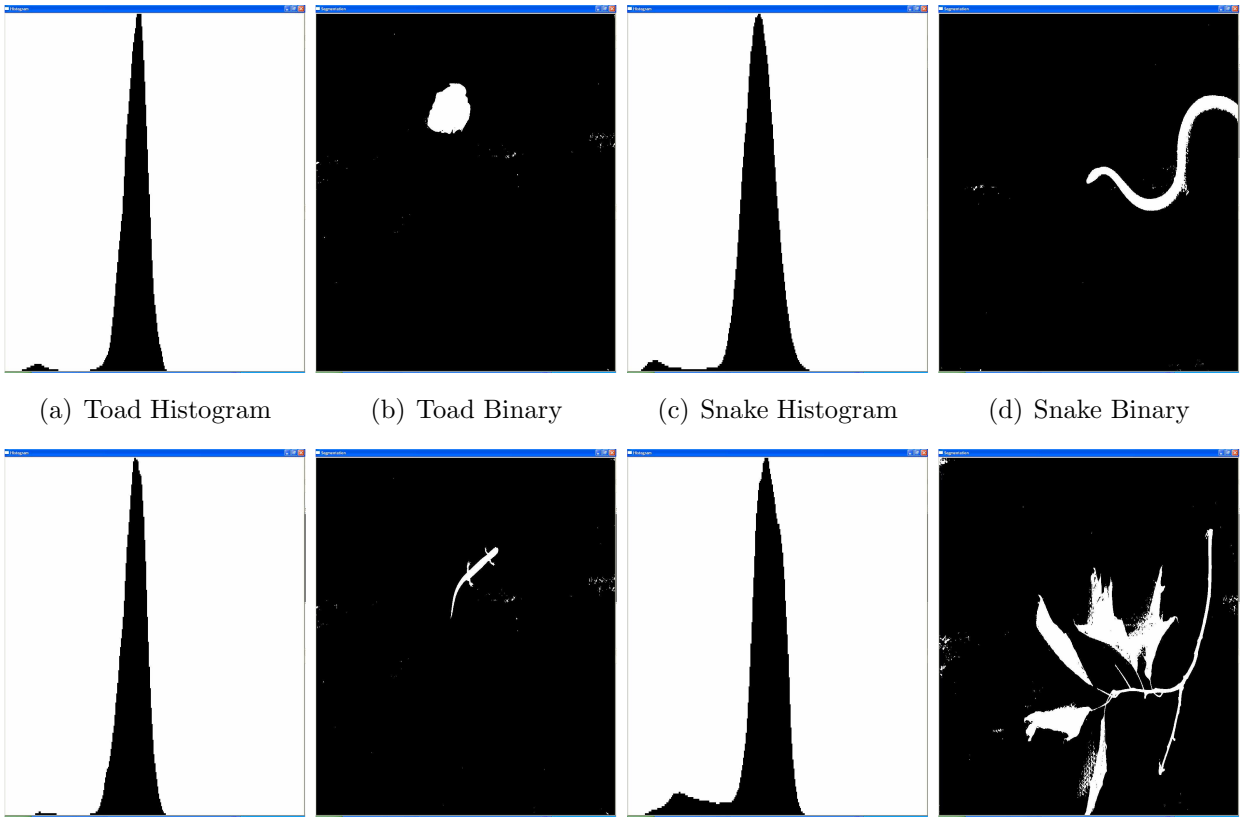
Figure 3.3. Creatures utilized in study and their point of capture.

3.2. Algorithm

3.2.1. Pre-Processing. Prior to extracting and evaluating the features of objects in an image, detection and segmentation must first be performed. Due to the structured background this is a relatively straightforward process. The dark object on the light background creates a bimodal histogram based on the intensity image. By dividing the image at the global minimum it is possible to establish a threshold which can be

[1]Department of Biological Sciences, RIT

used to perform foreground/background segmentation. Performing a gaussian smoothing operation to the image prior applying the threshold results in a significantly cleaner result object for extracting features. Some samples of object segmentation are shown in Figure 3.4.



(a) Toad Histogram (b) Toad Binary (c) Snake Histogram (d) Snake Binary

(e) Salamander Histogram (f) Salamander Binary (g) Branch Histogram (h) Branch Binary

Figure 3.4. The above figure displays the histogram information along with the binary image resulting from the segmentation.

3.2.2. Shape-Based Feature Extraction. The principle means of classification is done in the spirit of Cohen [**22**] and Wu [**23**] mentioned previously. This will involve abstractly representing the curvature information using a set of minimally deformable features(MDFs) extracted from each object. In order to qualify as a minimally deformable feature the characteristic must be one which is present in every object (or whose absence is itself a classifier). The feature must also fit into a set of bounds which can abstractly

differentiate that object class from at least one other object class. Due to the significantly different appearances in Figure 3.4 it is clear that there is a shape-based approach which is able to classify these objects. The features that were determined would make the classification possible are based on the objects' contour information.

After segmentation, each object's contours contours are broken down into concise clusters of points for which several parameters are calculated. The first step in extracting features from the objects is obtaining the deformation curves from the contour. These deformations are areas where the convex hull and the object's contours intersect. The starting point for each of these curves is saved to an array. These points are then clustered based on an object variant distance parameter. Significant information can be extracted from these clusters. Intuitively it can be observed that characteristics of clusters are prevalent given specific types of features. For example, points of high curvature are represented through long chains of points and appendages are typically represented by single points or tight groupings. The MDFs are to be representative of these sort of cluster-based abstractions. This is shown algorithmically in Figure 3.5 and is graphically illustrated in Figure 3.6.

Algorithm 3.2.1: Extraction Of MDFs(f)

$contour \leftarrow ImageContours$
$defects \leftarrow ConvexityDefects$
for each $deformity \in defects$
  do $pArray \leftarrow StartingDeformationPoint$
comment: Cluster the points.

for each $point \in pArray$
  do **while** $point \in DistanceThreshold$
    do $clusterVector \leftarrow point$
comment: Perform the feature extraction.

for each $cluster \in clusterVector$
  do $centroid \leftarrow$ calculateCentroid($contour$)
  $averageClusterVal \leftarrow AverageXandYvalues$
  CalculateAngle($averageClusterVal, centroid, prev$)
  DistanceToCenter($averageClusterVal, centroid$)

Figure 3.5. Algorithm for determining MDFs.

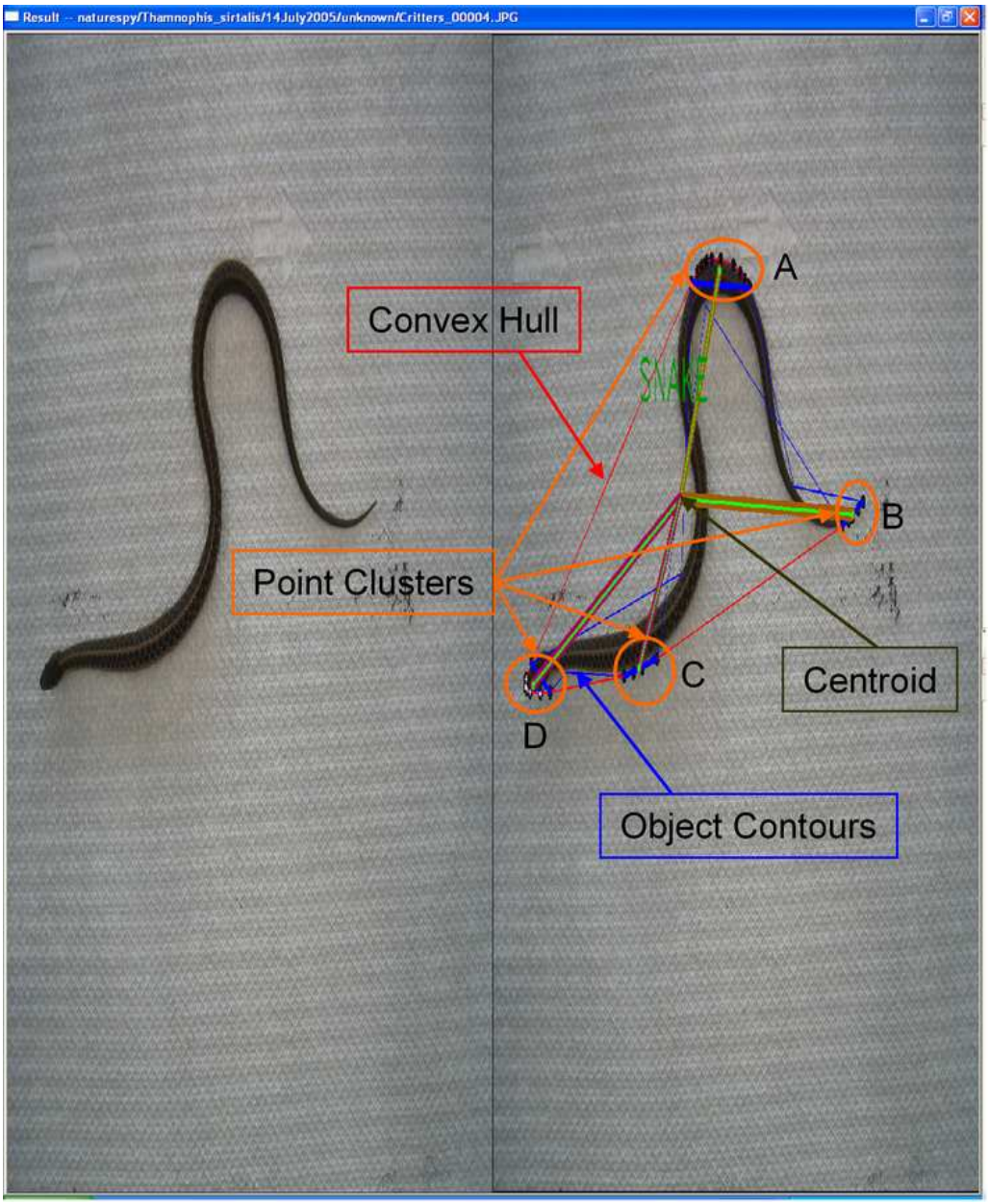


Figure 3.6. This figure provides a visual description of the MDFs. The basis of the features are the Point Clusters (A-D) located at the intersection of the convex hull and the object contours and the centroid. Each of the point clusters is indicative of a point of high curvature. These high curvature points help to anatomically describe the shape of a target object. The centroid is the center of mass of the image region. It is itself used as a feature and also is useful in relating the positions of the Point Clusters.

The calculated values are displayed graphically on the image (this was shown in 3.6). This helps to visualize the differences in features between object classes. Figure 3.7 illustrates an example of each type of creature after it has had its features labeled. The labels are described as follows:

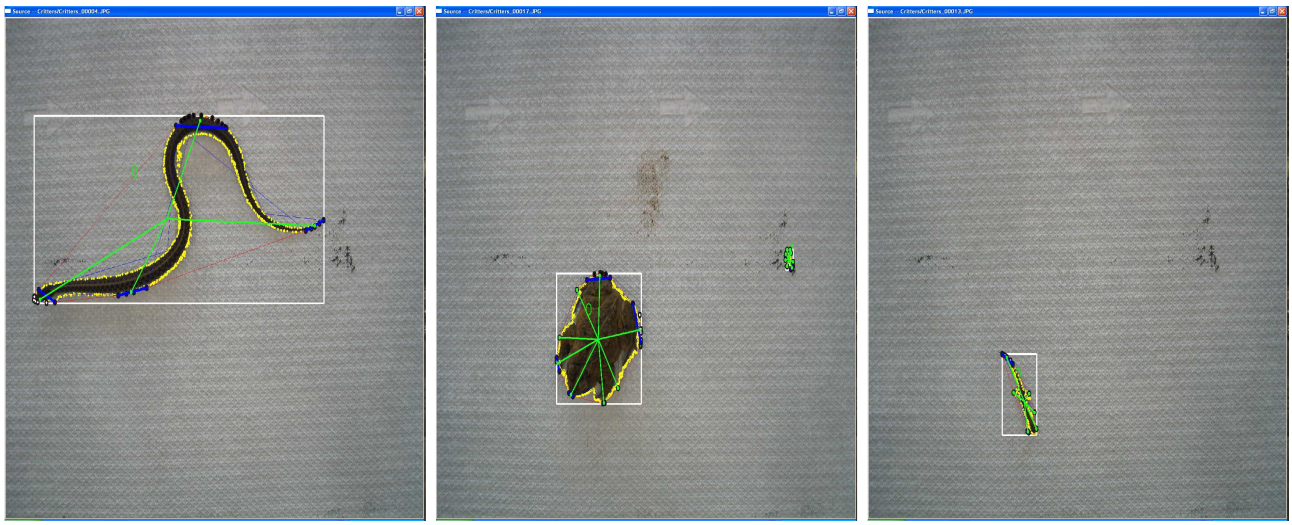
- BLACK: Deformation points.
- BLUE (thick): Line drawn between start and end of cluster.
- BLUE (thin): Convexity defects.
- GREEN: Line drawn between average cluster point and centroid.
- RED: The Item's convex hull.

These labels correspond to feature data to be used by the classifier. Figures 3.8 and 3.9 outline the information gathered visually from the labeled images of the target creatures and the nature objects.

The first feature to be addressed is the position of the object's center of mass or centroid. Since a frog is roughly circular object, the centroid will always be central to the frog's mass. This will not be the case in the highly deformable snake and salamander. While the centroid of a salamander is usually very close to the body portion of the creature, the centroid of a snake image is generally not located on the object at all. The position of the centroid is directly related to the segment intersection of the object. This 'segment intersection' is the quantity of the line drawn between the cluster average of each segment to the centroid, that intersects with the object of interest. In the case of the frog there should be complete interesection in most cases. This is intuitive when imagining the generally round shape of a frog. The salamander, except in cases where the posterior has a high curvature, will have a high quantity of intersection. The snake will have a very low intersective quantity.

The segments themselves also provide information about the underlying creature. In the case of frogs the size of the segments are roughly equal. For the snake this is almost

never the case as its deformations in shape often lead to major variations in segment length. Salamanders often have several points which are equidistant-the appendages-creating a useful discriminator from their appendage less brethren. The position and number of the clusters which create these segments are useful in that they generally map to the same regions on a creature, defining similar angles between segments. In frogs and toads the points are focused on the appendages and the anterior portion of the creature. The angles between the segments are generally acute with lateral segments fanning out from the centroid to the appendages in the ideal case. The salamander has the same type of clustering scheme where points are centered anteriorly and at the appendages. The posterior, tail region becomes more important in a salamander as it is often in a deformed position generating more points. The snake has the fewest number of point clusters in most cases. It can reliably be assumed that a cluster will be centered on both the head and tail region with other points being dependent on the degree of deformation of the snake. The angles between these points are often obtuse.



(a) Snake (b) Toad (c) Salamander

Figure 3.7. Illustrates the labelling performed upon completion of the feature extraction and calculation algorithm.

Animal	Centroid Location	Segment Intersection	Segment Length	Segment Angle	Number of Deformed Clusters
Frog-Toad	Central To Frog Mass	Complete	Roughly Equal	Ideal-Lateral Groupings	Few-Some (Anterior, Posterior, Appendages)
Salamander	On or Near Body Mass	Majority	Some Equidistant(also Major Axis)	Ideal Bisections	Few-Many(Head, Feet, Tail)
Snake	Often Not On Body	Minimal	Varies	Often Obtuse	Few(Head, Tail, Major Curves)

Figure 3.8. Displays the differences in shape based features extracted using cluster deformation points and centroid data from target creatures.

Object	Centroid Location	Segment Intersection	Segment Length	Segment Angle	Number of Deformed Clusters
Acorn	Central to Mass	Complete	Roughly Equal	Circular Groupings	Few-Some
Leaf	Central to Mass	Varies with Leaf Type	Varies with Leaf Type	Varies with Leaf Type	Few-Some
Pine Cone	Central to Mass	Complete	Roughly Equal	Ideal-Lateral Groupings	Few
Stick	Often Not On Mass-High Variation	Minimal-High Variation	Varies	Often Obtuse	Few-Many

Figure 3.9. Displays the differences in shape based features extracted using cluster deformation points and centroid data from nature items.

The descriptions of deformable features in Figure 3.8 have been implemented so that they can be accepted as input by the classifier. The first feature, *Centroid Location*, is a simple boolean value to declare if the object centroid intersects with the mass of the creature's body (1) or if it intersects with the background (0). The *Segment Angle* is the mean value of the angle formed by each of the segments. The *Number of Deformed Clusters* is done as a count which is fed into one of the input neurons. The final two features listed in the figure were deemed empirically to be detrimental to the performance

of the system. The segment intersection and the segment length were implemented as the average quantity of intersection and the standard deviation of the segment length respectively. More details of why they were not utilized in the final system will be discussed in the results section.

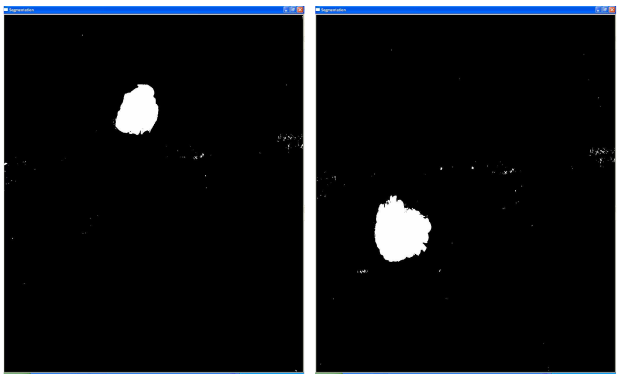
The remaining shape based features are not based on Minimally Deformable Features and are instead part of the more classical set of image processing values used for classification. These are the axis ratio and the compactness. The axis ratio is simply the ratio between the major and minor axis and is shown in 8. Compactness (or Circularity) is a measure which quantifies the external irregularities of an object. Its calculation is shown in 9.

$$AR = MajorAxis/MinorAxis \tag{8}$$

$$C = Area/Perimeter^2 \tag{9}$$

Each of these features is equally scaled and fed into the classifier. More on this is discussed in 3.2.4.

3.2.3. Texture-Based Feature Extraction. The incorporation of texture as a feature was done to aid in the differentiation of objects which have a high degree of shape correlation but need to be differentiated. This was particularly problematic with regard to comparing objects of type nature to certain creature types and configurations. This is illustrated in Figure 3.10.



(a) Toad Image (b) Pine Cone Image

Figure 3.10. Illustrates the problem that can come about when depending exclusively on shape-based features for classification. The segmented toad and the segmented pine cone appear virtually identical.

To extract the texture information, the mean value of each point cluster is again utilized. A gray-scale patch is extracted from the region around each point. The size of each patch was experimentally determined to be 15x15 pixels. Example texture patches are shown in Figure 3.11. This patch is then subjected to the first and second order statistical operations discussed in Section 2.1.3 on page 11.

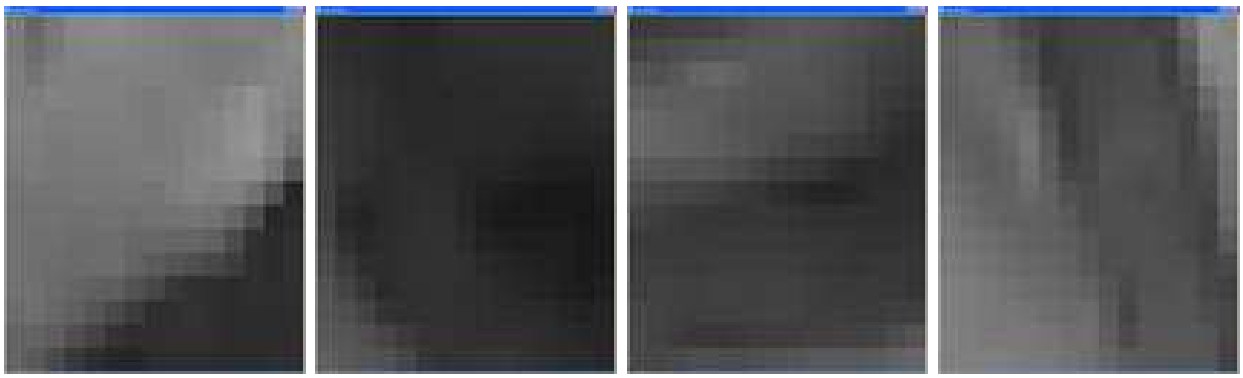


Figure 3.11. Sample Texture Patches

3.2.4. Classifier. The classification is performed with a feed-forward neural network using a library designed by Allesandro Presta[2] and released under the GNU Lesser General

[2]http://nnf.sourceforge.net/

Public License. The library provides a high speed implementation of the neural network that is configurable and able to be integrated into existing frameworks.

The classifier features an input neuron for each feature, an output for each target class, and a hidden layer that was adjusted during training to create an optimal solution. This is shown graphically in Figure 3.12. Each of the input neurons is scaled equally to reduce the input parameter to be compatible with the logistic activation function.

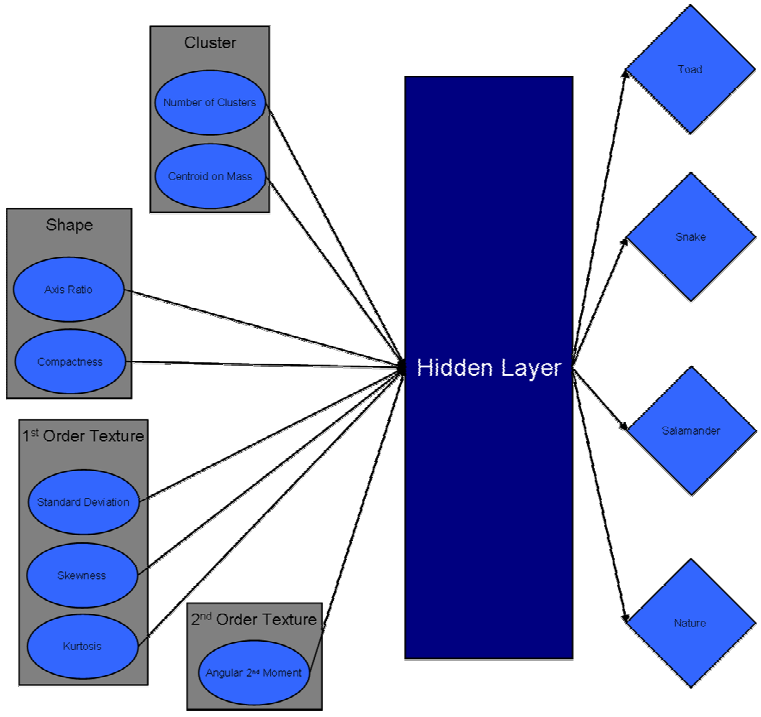


Figure 3.12. Illustrates the structure of the classifier utilized in the experiment. Each feature received is assigned an input neuron. This in turn is passed to the computational neurons of the hidden layer where the weighted values are calculated. Finally the output layer features a node for each candidate. A match is determined when a threshold is crossed for the maximal output value, else the object is classified 'Undefined.'

The final structure of the classifier was determined through continual experimentation and adjusting of the parameters. To determine the best combination of values of features and parameters Mean Square Error (MSE) calculations were performed and graphed and match percentages were calculated and tracked for the parameters. The

equation used to perform the MSE is shown as 10. This is a comparison between the value obtained by the network and the desired value for a particular object. As the value approaches zero the neural network's accuracy is improving due to its learning of the features of the training set. A sample set of results of the calculation are shown in Figure 3.13.

$$MSE(t) = \frac{1}{n}\sum_{i=1}^{k} f_i(x_i - t)^2 \tag{10}$$

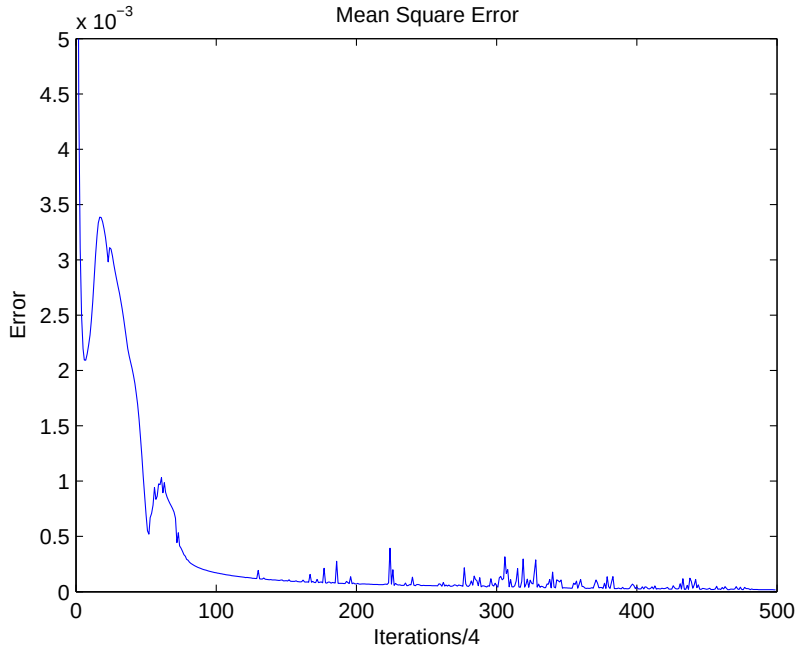


Figure 3.13. The above figure represents the graphed results of the Mean Square Error calculation obtained during training of the classifier during 20,000 epochs. Notice the graph approaches zero illustrating the improvement gained by training.

MSE was just part of the refining of the classifier. End match percentages were compared for different combinations of training data, training length, and neural network configurations. This measure is the most obvious means of validating the system's performance. A match percentage close to 100% classification for each of the 4 classes

reflects ideal output values for the system. Common refinements included adjusting the number of layers and neurons in the hidden layer and the adjustment of the training data.

The current parameters of the hidden layer are 2 layers of 4 neurons each. This seems to give good classification while still being general enough to handle creature images on which it was not trained. The set of training images was another point of continual refinement. Since a neural network is only able to recognize objects on which it has been trained, it has required multiple adjustments to obtain the right balance in classification accuracy between the classes. The most difficult aspect of the classification deals with distinguishing 'Nature' objects from the 3 creature types. More on this will be discussed later in the results section.

Chapter 4

Results

4.1. Training Data Summary

Due to the utilization of a neural network based classifier the results of the system are highly dependent on the training data. During final system evaluations a balanced amount of data for each creature type was presented to the system for training. Following the training, new images were utilized as a test set. By using images that were previously unseen by the system it was possible to verify the classifier's generalization of the data[1].

4.2. Intra-Target Training

Experiments were initially performed with only creature data in the training set. Each creature was given a balanced representation. This was typically 5 images per creature. Varied representation in pose was attempted. If possible, images of each creature at multiple sizes was utilized. The results of this type of training proved to be very positive.

The classification that resulted from training data representative of only the target information resulted in a high overall match percentage. This is shown in the table in Figure 4.1. The performance varied across training instances due to the random initialization of the neural network. This classification was performed using only the shape-based classifiers. The distinctiveness of the cluster-based features derived from the

[1]Unfortunately this was not possible with snake images due to the small number of sample creatures

contour information of the 3 creatures was enough to provide strong decision boundaries across the feature space.

	Salamander	**Snake**	**Toad**
Salamander (72)	61 (84.7%)	0	11 (15.3%)
Snake (9)	1 (11.1%)	7 (77.8%)	1 (11.1%)
Toad (86)	0	0	86 (100.0%)

FIGURE 4.1. This table illustrates the power of the system when dealing with only the target classes. The left axis represents the actual creatures as they are presented to the system. The top axis is the results of the classification. Note: it is difficult to accurately judge the performance for the 'snake' class due to its under representation.

Misclassification was typically manifested when the creature, usually a salamander, was in a position that caused the clusters to be in a radial pattern around the the centroid. The radial pattern is more representative of a frog or toad. The position reflected a minority of the data collected and is likely representative of posing the creature in front of the camera.

4.3. Target and General Training

In order to improve the realism, non-target images were be tested on the system. This was leveraged by the introduction of a separate 'Nature' category. This category included anything that did not fit into the 3 targets and generally involved leaves, sticks, and anything else commonly found on the ground in a wooded area.This greatly increased the difficulty of the problem and required more training data to compensate for the new targets.

Introduction of the nature objects inspired the evaluation of texture data. As was discussed on Section 2.1.3 on page 11, texture was incorporated in order to differentiate objects which had a high degree of shape correlation. As it is common for flora to feature

enclosed shapes with equidistant protrusions, this caused a default 'Toad' classification for many leaves based on their shape-based features. Adding the textural features increased the abilitiy of the system to differentiate between creatures and objects in the general nature category.

In order to maintain a positive balance between correctly classifying the creatures and distinguising creatures from the 'Nature' category, training data was carefully selected. The best results were achieved representing each creature with 5 examples in the training set. 'Nature' was trained using 2 samples of a general object target. These general object targets are attempts at abstracting the overarching category to which they belong (such as leaf or pine cone).

Not maintaining the balance among training data categories causes severe problems in the results of the classifier. Under-representing the 'Nature' category caused overclassification of creature objects. As a result of the low representation, the classifier was more likely to match a creature, something that had been presented more frequently, then declare a match to one of the generic nature targets. This is shown in the table in Figure 4.2. Over-representing the nature class had the opposite effect. This effectively caused a subversion of the target creature classes causing almost all classifications to be 'Nature.' This is illustrated in Figure 4.3. The difference in Figures 4.2 and 4.3 is an increased representation of misclassified nature classes. The addition of this training data causes a reformulation of the classifier's decision boundaries reducing the classification generality for the 3 target classes.

	Salamander	**Snake**	**Toad**	**Nature**
Salamander (72)	62 (86.1%)	0	9 (12.5%)	1 (1.4%)
Snake (9)	1 (11.1%)	7 (77.8%)	0	1 (11.1%)
Toad (86)	0	0	82 (95.3%)	4 (4.7%)
Nature (93)	31 (33.3%)	0	33 (35.5%)	29 (31.1%)

Figure 4.2. This table shows the problem that arose when a 'Nature' class was introduced. The training data used to build the 4th class was limited in this example. The accuracy numbers for the target classes deviated only slightly, but the new class performed poorly.

	Salamander	Snake	Toad	Nature
Salamander (72)	64 (88.0%)	0	1 (1.4%)	7 (9.7%)
Snake (9)	1 (11.1%)	7 (77.8%)	0	1 (11.1%)
Toad (86)	2 (2.3%)	0	36 (41.8%)	48 (55.8%)
Nature (93)	42 (45.1%)	4 (4.3%)	12 (12.9%)	46 (49.5%)

FIGURE 4.3. This table shows the result of increasing the training data associated with the 'Nature' class. There is a marginal increase in classification ability with respect to the new class, but there is a significant performance reduction in the classification of toads.

Although the shape-based classifiers should ideally create a general classifier for each of the creature types, the system had difficulty recognizing species with which it had no prior experience. This was likely due to the incorporation of textural information. This made it necessary to train on each species. An interesting corollary to this is that it was impossible to differentiate between frogs and toads. It was hypothesized that difference in skin texture would create a suitable difference between the two classes. Given the inclusion of this textural data it was theorized that the natural progression would be to differentiate between those similar classes. This left the final system with the 3 creatures classes {frogs/toads, salamanders, snakes} and the suplemental 'Nature' class. Sample positive classification results of creature data are shown in Figures A.1, A.2, and A.3. Correctly matched 'Nature' objects are shown in Figure A.4.

In the case of misclassification it is most common for the creature objects to be placed into the 'Nature' category. This is due to the relatively close decision space between each of the creatures and that of 'Nature'. Due to the high classification rate of Intra-Target objects using shape-based features it is likely that there is a large separation between the 3 target spaces, detailing a high degree of generality in the classifier. Furthermore, due to the high variability of the 4th category, it is probable that given the proper set of conditions a creature will 'look like' another piece of nature. This demonstrates the problem with working with objects which are naturally inclined to avoid detection. Some examples of this are shown in Figure A.5.

4.4. Elimination of MDFs

Two minimally deformable features discussed in the algorithm implementation were eliminated from the final system. This was done due to the reduction in match accuracy caused by the inclusion of segment intersection and segment length. When part of the system these parameters caused an abundance of creatures to be classified as 'Nature.'

Chapter 5

Conclusion

5.1. Final Thoughts

Shape-based features are a powerful means of classification among objects which are recognized by distinctive protrusions. The task at the onset of this research was to create a system to distinguish between the 3 creature types. As was shown in the initial experimentation these creatures were distinct enough to be classified with a high degree of regularity between each other. This makes the overall goal of the research successful. The addition of natural noise into the system reduced the overall effectiveness of the system. This was partially offset by the inclusion of textural features. These features made it possible to distinguish objects which had similar shape characteristics, but did not provide the level of discriminating ability as was seen between only the 3 target classes.

5.2. Future Work

5.2.1. Improved Classifier Methodology. Neural networks offer a highly configurable classifier. There are a number of neural network subtypes which could be explored which may yield better classification. Additionally, due to the configurability of the feed-forward network utilized there are multiple possibilities available refining this solution.

These range from adjusting the activation function on the neurons, to tweaking the learning rate, to refining the structure of the hidden layer. Although some experimentation was done in these capacities, it surely was not exhaustive.

The use of a neural network as the classifier brought about the problem of reproduceability in results. A newly trained network was likely to yield different results each time. A potential way to address this is to utilize Support Vector Machines in the classifier. This adjustment would linearly separate categories using a standardized function. The standardization would create reproducible classifiers.

5.2.2. Addition and Alteration of Features. The attempted classification of the nature data proved to not have the best marks with regard to performance. Ideally the percentages should have been closer to the 80% to 90% achieved for the creatures. It was originally assumed that the simple textural features would be enough to distinguish these elements, but as was discussed previously this was not the case. In future work other advanced techniques for extracting texture data, such as wavelets, should be reviewed and possibly incorporated into the system.

5.2.3. More Data. When dealing with pattern recognition problems there is the issue of how much data to work with. An increased sample size will reflect a better cross section of the potential population. In the case of this experiment there were clear deficiencies in the quantity of some of the data types. This is particularly apparent with the snake images. Only 6 were available because of the difficulty in obtaining the creatures during hot summer months. This meant that that the classifier was reviewing images it had been trained upon to gather final conclusions. This is clearly not conclusive means of testing.

5.2.4. System Expansion. In the future this system will have to distinguish between a significantly larger sample of creatures. This will likely require improvements in

each of the aforementioned areas: New features will have to be reviewed along with the structure of the classifier itself.

References

[1] Ossama El Badawy and Mohamed Kamel. *Shape Representation Using Concavity Graphs.* In *ICPR (3)* pages 461–464 2002.

[2] Dirk Walther, Duane R. Edgington and Christof Koch. *Detection and Tracking of Objects in Underwater Video.* In *CVPR (1)* pages 544–549 2004.

[3] Tilo Burghardt, Janko Calic and Barry Thomas. *Tracking Animals in Wildlife Videos Using Face Detection.* In *European Workshop on the Integration of Knowledge, Semantics and Digital Media Technology* October 2004.

[4] Z. Kalafatic. *Model-based tracking of laboratory animals.* In *EUROCON 2003. Computer as a Tool. The IEEE Region 8* Band 2 pages 175–178 vol.2 2003.

[5] Michael J. Swain and Dana H. Ballard. *Color indexing.* Int. J. Comput. Vision **7** (1), 11–32 (1991).

[6] Rafael C. Gonzalez, Richard E. Woods and Steven L. Eddins. *Digital Image Processing Using MATLAB.* Prentice-Hall, Inc. Upper Saddle River, NJ, USA 2003.

[7] Adrian Ford and Alan Roberts. *Colour Space Conversions* 1998.

[8] J. Terrillon and S. Akamatsu. *Comparative Performance of Different Chrominance Spaces for Color Segmentation and Detection of Human Faces in Complex Scene Images* 2000.

[9] Graham D. Finlayson, Steven D. Hordley and Paul M. Hubel. *Color by Correlation: A Simple, Unifying Framework for Color Constancy.* IEEE Trans. Pattern Anal. Mach. Intell. **23** (11), 1209–1221 (2001).

[10] Jure Kova Peter. *Illumination Independent Color-Based Face Detection.*

[11] Brian Funt, Kobus Barnard and Lindsay Martin. *Is Machine Colour Constancy Good Enough?* Lecture Notes in Computer Science **1406**, 445–?? (1998).

[12] Lihui Chen and Christos Grecos. *Fast skin color detector for face extraction.* Band 5671 pages 93–101. SPIE 2005.

[13] Erica Bree Rosenbaum, Hopi E. Hoekstra and Michael W. Nachman. *Adaptive Reptile Color Variation and the Evolution of the MC1R Gene*. pages 1794–1808 2004.

[14] Kevin J. McGraw, Emiko A. Mackillop, James Dale and Mark E. Hauber. *Different colors reveal different information: how nutritional stress affects the expression of melanin- and structurally based ornamental plumage*. J Exp Biol **205** (23), 3747–3755 (2002).

[15] Kevin J. McGraw. *Winter Plumage Coloration in Male American Goldfinches: Do Reduced Ornaments Serve Signaling Functions in the Non-Breeding Season?* Ethology **110** (9), 707–715 (2004).

[16] D'Arcy Thompson. *On Growth and Form*. Cambridge University Press 1917.

[17] H. Freeman. *On the encoding of arbitrary geometric configurations*. IRE Transactions (2), 260–268 (1961).

[18] H. Freeman. *H. Boundary encoding and processing*. Academic Press New York 1973.

[19] Rafa Absar and Junaed Sattar. *Freeman Chain Coding: A Short Tutorial*.

[20] H. Freeman and R. Shapira. *Determining the minimum-area encasing rectangle for an arbitrary closed curve*. Commun. ACM **18** (7), 409–413 (1975).

[21] M.M. McQueen and G.T. Toussaint. *On the Ultimate Convex Hull Algorithm in Practice*. PRL **3**, 29–34 (1985).

[22] Isaac Cohen, Nicholas Ayache and Patrick Sulger. *Tracking Points on Deformable Objects Using Curvature Information*. In *ECCV '92: Proceedings of the Second European Conference on Computer Vision* pages 458–466 London, UK 1992. Springer-Verlag.

[23] Ying Wu, Ting Yu and Gang Hua. *A Statistical Field Model for Pedestrian Detection.* In *CVPR (1)* pages 1023–1030. IEEE Computer Society 2005.

[24] M. Kass, A. Witkin and D. Terzopoulos. *Snakes: Active Contour Models*. International Journal on Computer Vision **1** (4), 321–331 (1988).

[25] Jerry L. Prince Chenyang Xu. *Snakes, Shapes, and Gradient Vector Flow*. IEEE Transactions on Image Processing **7**, 359–369 (1998).

[26] *Tracking deformable objects with the active contour model.* In *ICMCS '97: Proceedings of the 1997 International Conference on Multimedia Computing and Systems (ICMCS '97)* page 608 Washington, DC, USA 1997. IEEE Computer Society.

[27] J.S. Kim, K.C. Koh and H.S. Cho. *A Novel Contour Model for Locating Objects in Images.* In *Intelligent Robots and Systems* pages 64–69 1999.

[28] Tien-Lung Chang and Tyng-Luh Liu. *Detecting Deformable Objects with Flexible Shape Priors.* In *ICPR (4)* pages 155–158 2004.

[29] S. Madhvanath, G. Kim and V. Govindaraju. *Chaincode Contour Processing for Handwritten Word Recognition.* IEEE Trans. Pattern Anal. Mach. Intell. **21** (9), 928–932 (1999).

[30] Mihran Tuceryan and Anil K. Jain. *Texture analysis.* pages 235–276 (1993).

[31] A. Pitiot, A. Toga, N. Ayache and P. Thompson. *Texture based MRI segmentation with a two-stage hybrid neural classifier.* In *World Congress on Computational Intelligence / INNS-IEEE International Joint Conference on Neural Networks WCCI-IJCNN'02* 2002.

[32] A. Ahmadian, A. Mostafa, M.D. Abolhassani and N. Riahi Alam. *A method for texture classification of ultrasonic live images based on Gabor Wavelet.* page 971=974 2004.

[33] C. H. Chen, L. F. Pau and P. S. P. Wang, Editoren. *Handbook of pattern recognition & computer vision.* World Scientific Publishing Co., Inc. River Edge, NJ, USA 1993.

[34] I. Christoyianni, E. Dermatas and G. Kokkinakis. *Neural Classification Of Abnormal Tissue In Digital Mammography Using Statistical Features Of The Texture.* IEEE ICECS **2**, 117–120 (1999).

[35] Fyzodeen Khan and Ying Sun. *Morphological Templates for Extracting Texture Information in X-Ray Mammography.* In *14th IEEE Symposium on Computer Based Medical Systems* pages 375–380 2001.

[36] R. M. Haralick, K. Shanmugam and I. Dinstein. *Textural Features for Image Classification.* **3** (6), 610–621 (November 1973).

[37] S. G. Mallat. *A Theory for Multiresolution Signal Decomposition: The Wavelet Representation.* IEEE Trans. Pattern Anal. Mach. Intell. **11** (7), 674–693 (1989).

[38] Amara Graps. *An Introduction to Wavelets*. IEEE Comput. Sci. Eng. **2** (2), 50–61 (1995).

[39] R. R. Brooks, L. L. Grewe and S. S. Iyengar. *Recognition in the wavelet domain: a survey*. Journal of Electronic Imaging **10**, 757–784 (jul 2001).

[40] Moshe Porat and Yehoshua Y. Zeevi. *The Generalized Gabor Scheme of Image Representation in Biological and Machine Vision*. IEEE Trans. Pattern Anal. Mach. Intell. **10** (4), 452–468 (1988).

[41] C.G. Zhao, H.Y. Cheng, Y.L. Huo and T.G. Zhuang. *Liver CT-image retrieval based on Gabor texture*. In *Engineering in Medicine and Biology Society, 2004. EMBC 2004. Conference Proceedings. 26th Annual International Conference of the* Band 1 pages 1491–1494 Vol.2 2004.

[42] Yongqing Xiang, Vanessa Yingling, Jonathan Silverberg, Mitchell B. Schaffler and Theodore Raphan. *Quantification of trabecular bone mass and orientation using Gabor wavelets*. In *SAC '03: Proceedings of the 2003 ACM symposium on Applied computing* pages 183–188 New York, NY, USA 2003. ACM Press.

[43] Hamzah Arof and Farzin Deravi. *One-dimensional Fourier transform coefficients for rotation invariant texture classification*. Band 2908 pages 152–159. SPIE 1996.

[44] N. Gotze, S. Drue and G. Hartmann. *Invariant object recognition with discriminant features based onlocal fast-Fourier Mellin transform*. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on* Band 1 pages 948–951 vol.1 2000.

[45] Stéphane Derrode and Faouzi Ghorbel. *Robust and Efficient Fourier-Mellin Transform Approximations for Gray-Level Image Reconstruction and Complete Invariant Description*. Computer Vision and Image Understanding: CVIU **83** (1), 57–78 (2001).

[46] R. A. Fisher. *The Use of Multiple Measurements in Taxonomic Problems*. Annals of Eugenics **7** (2), 179–188 (1936).

[47] Jose C. Principe, Neil R. Euliano and W. Curt Lefebvre. *Neural and Adaptive Systems: Fundamentals through Simulations with CD-ROM*. John Wiley & Sons, Inc. New York, NY, USA 1999.

[48] Walter F. Bischof and Terry Caelli. *Scene Understanding by Rule Evaluation.* IEEE Trans. Pattern Anal. Mach. Intell. **19** (11), 1284–1288 (1997).

[49] QingQing Zhou and Martin Purvis. *A market-based rule learning system.* In *CRPIT '04: Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation* pages 175–180 Darlinghurst, Australia, Australia 2004. Australian Computer Society, Inc.

[50] Wensheng Zhou, Asha Vellaikal and C. C. Jay Kuo. *Rule-based video classification system for basketball video indexing.* In *MULTIMEDIA '00: Proceedings of the 2000 ACM workshops on Multimedia* pages 213–216 New York, NY, USA 2000. ACM Press.

[51] Kristin P. Bennett and Colin Campbell. *Support vector machines: hype or hallelujah?* SIGKDD Explor. Newsl. **2** (2), 1–13 (2000).

[52] King-Shy Goh, Edward Chang and Kwang-Ting Cheng. *SVM binary classifier ensembles for image classification.* In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management* pages 395–402 New York, NY, USA 2001. ACM Press.

[53] Evgeniy Gabrilovich and Shaul Markovitch. *Text categorization with many redundant features: using aggressive feature selection to make SVMs competitive with C4.5.* In *ICML '04: Proceedings of the twenty-first international conference on Machine learning* page 41 New York, NY, USA 2004. ACM Press.

[54] Yuki Tanaka, Hiroya Takamura and Manabu Okumura. *Extraction and classification of facemarks.* In *IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces* pages 28–34 New York, NY, USA 2005. ACM Press.

[55] Balaji Krishnapuram, Lawrence Carin and Alexander J. Hartemink. *Joint classifier and feature optimization for cancer diagnosis using gene expression data.* In *RECOMB '03: Proceedings of the seventh annual international conference on Research in computational molecular biology* pages 167–175 New York, NY, USA 2003. ACM Press.

[56] Yaguang Yang, Kristen Summers and Mark Turner. *A text image enhancement system based on segmentation and classification methods.* In *HDP '04: Proceedings*

of the 1st ACM workshop on Hardcopy document processing pages 33–40 New York, NY, USA 2004. ACM Press.

[57] Bertrand Le Saux and Giuseppe Amato. *Image recognition for digital libraries.* In *MIR '04: Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval* pages 91–98 New York, NY, USA 2004. ACM Press.

[58] Chih-Fong Tsai, Ken McGarry and John Tait. *Image classification using hybrid neural networks.* In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval* pages 431–432 New York, NY, USA 2003. ACM Press.

[59] Corinna Cortes and Vladimir Vapnik. *Support-Vector Networks.* Machine Learning **20** (3), 273–297 (1995).

[60] Cemil Kirbas and Francis Quek. *A review of vessel extraction techniques and algorithms.* ACM Comput. Surv. **36** (2), 81–121 (2004).

[61] Teuvo Kohonen, Editor. *Self-organizing maps.* Springer-Verlag New York, Inc. Secaucus, NJ, USA 1997.

[62] Richard P. Lippmann. *An introduction to computing with neural nets.* pages 36–54 (1988).

[63] Mahmoud I. Khalil and Mohamed M. Bayoumi. *Invariant 2D object recognition using the wavelet transform and structured neural networks.* Band 3723 pages 330–340. SPIE 1999.

[64] Dionisis Cavouras, Panos Prassopoulos, Gregory Karangellis, Maria Raissaki, Lena Kostaridou and George Panayiotakis. *Applications of a Neural Network and Four Statistical Classifiers in Characterizing Small Focal Liver Lesions on CT.* pages 1145–1146 1996.

Appendix A

Result Images

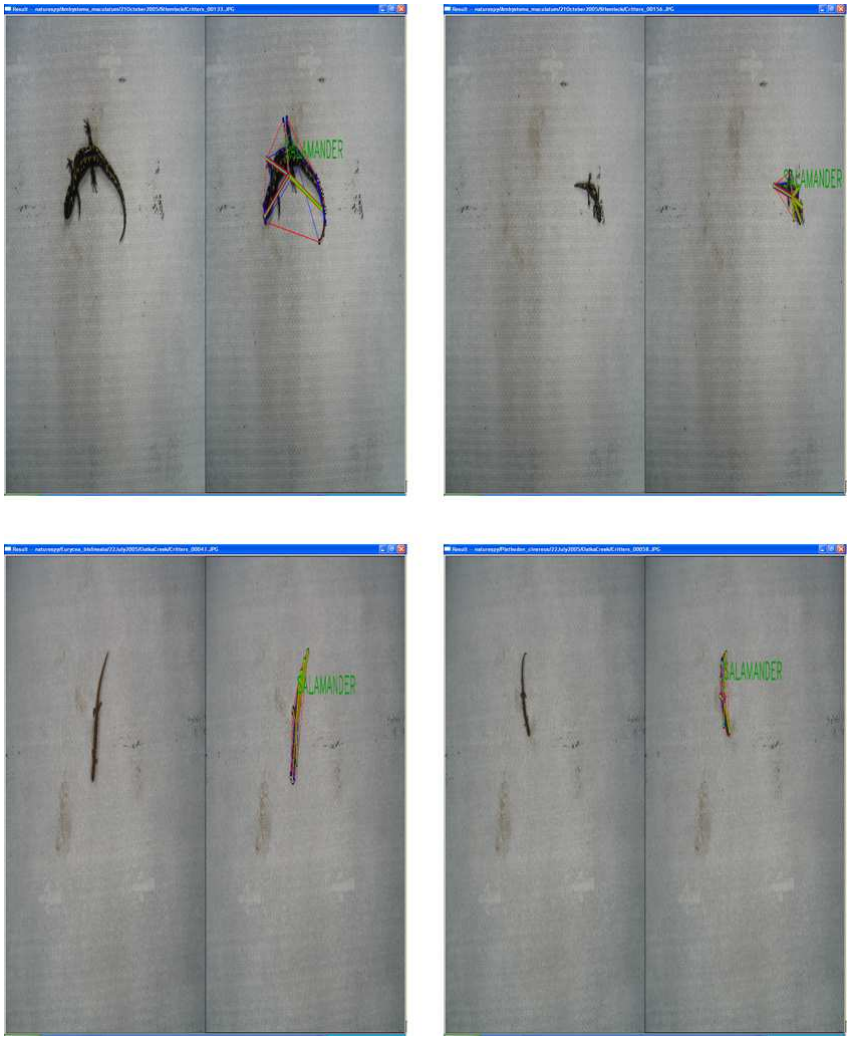


Figure A.1. Figure showing the results of correct classification of salamanders.

Figure A.2. Figure showing the results of correct classification of snakes.

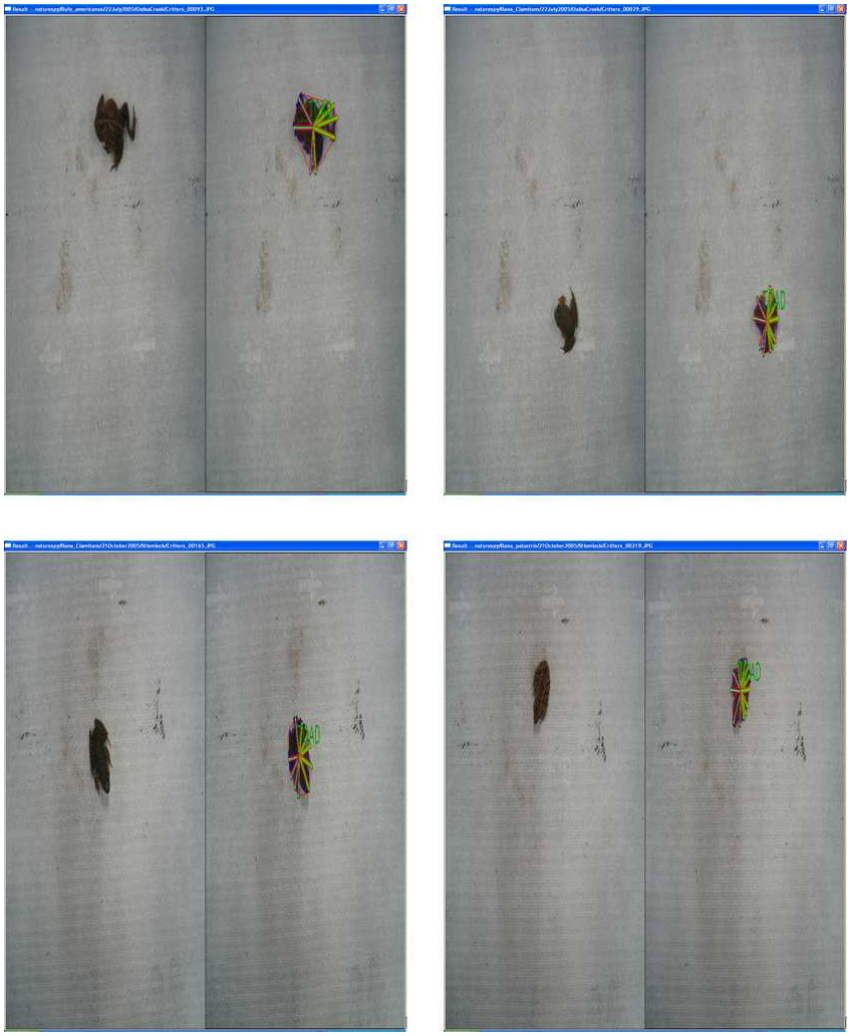


Figure A.3. Figure showing the results of correct classification of frogs and toads.

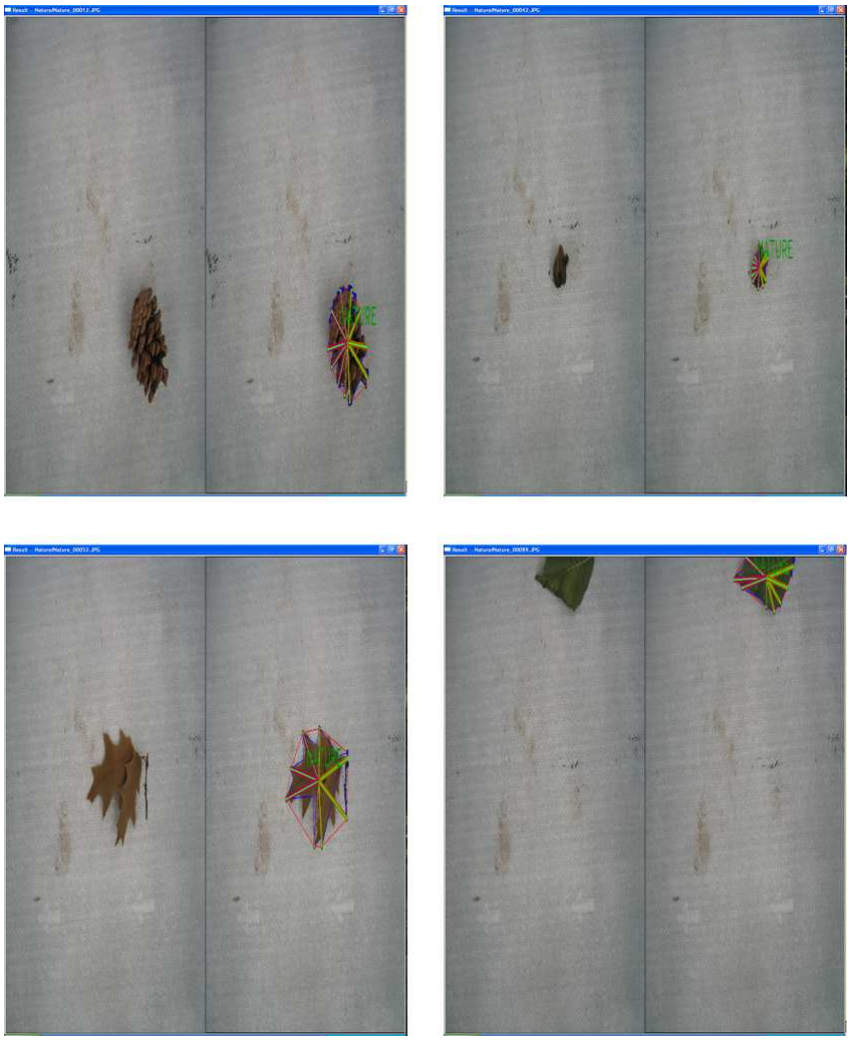


Figure A.4. Figure showing the results of the correct classification of non-specific nature objects.

Figure A.5. Figure showing the results of incorrect classification of creatures.