Towards a Versatile System for the Visual Recognition of Surface Defects

by

Miroslav Koprnicky

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Systems Design Engineering

Waterloo, Ontario, Canada, 2005

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Automated visual inspection is an emerging multi-disciplinary field with many challenges; it combines different aspects of computer vision, pattern recognition, automation, and control systems. There does not exist a large body of work dedicated to the design of generalized visual inspection systems; that is, those that might easily be made applicable to different product types. This is an important oversight, in that many improvements in design and implementation times, as well as costs, might be realized with a system that could easily be made to function in different production environments.

This thesis proposes a framework for generalizing and automating the design of the defect classification stage of an automated visual inspection system. It involves using an expandable set of features which are optimized along with the classifier operating on them in order to adapt to the application at hand. The particular implementation explored involves optimizing the feature set in disjoint sets logically grouped by feature type to keep search spaces reasonable. Operator input is kept at a minimum throughout this customization process, since it is limited only to those cases in which the existing feature library cannot adequately delineate the classes at hand, at which time new features (or pools) may have to be introduced by an engineer with experience in the domain.

Two novel methods are put forward which fit well within this framework: cluster-space and hybrid-space classifiers. They are compared in a series of tests against both standard benchmark classifiers, as well as mean and majority vote multi-classifiers, on feature sets comprised of just the logical feature subsets, as well as the entire feature sets formed by their union. The proposed classifiers as well as the benchmarks are optimized with both a progressive combinatorial approach and with an genetic algorithm. Experimentation was performed on true colour industrial lumber defect images, as well as binary hand-written digits.

Based on the experiments conducted in this work, it was found that the sequentially optimized multi hybrid-space methods are capable of matching the performances of the benchmark classifiers on the lumber data, with the exception of the mean-rule multi-classifiers, which dominated most experiments by approximately 3% in classification accuracy. The genetic algorithm optimized hybrid-space multi-classifier achieved best performance however; an accuracy of 79.2%.

The numeral dataset results were less promising; the proposed methods could not equal benchmark performance. This is probably because the numeral feature-sets were much more conducive to good class separation, with standard benchmark accuracies approaching 95% not uncommon. This indicates that the cluster-space transform inherent to the proposed methods appear to be most useful in highly dependant or confusing feature-spaces, a hypothesis supported by the outstanding performance of the single hybrid-space classifier in the difficult texture feature subspace: 42.6% accuracy, a 6% increase over the best benchmark performance.

The generalized framework proposed appears promising, because classifier performance over feature sets formed by the union of independently optimized feature subsets regularly met and exceeded those classifiers operating on feature sets formed by the optimization of the feature set in its entirety. This finding corroborates earlier work with similar results [3, 9], and is an aspect of pattern recognition that should be examined further.

Acknowledgements

Firstly I would like to thank my supervisors, Dr. Ahmed and Dr. Kamel, for their contributions and continued support throughout this endeavor, for which I am very grateful. Thanks also to my readers, Dr. Tizhoosh and Dr. Freeman, whose time and input has proved invaluable.

Funding provided by my supervisors, the Province of Ontario, and the University of Waterloo through the Ontario Graduate Scholarship (OGS) program must also be acknowledged, without which my work would surely have been rendered impossible.

Thanks are also due to all of my friends, who consistently encouraged me in my studies, and made sure I would not end them prematurely.

Lastly, I must thank my family, and my parents in particular, not only for all of their kind words and constant support, but mainly for the many sacrifices they made, especially early in the course of my life, almost exclusively for my benefit.

Thank you both so much, and thank you all.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

1.1 General

Automated defect inspection is a critical step in most manufacturing plants, and often represents a significant percentage of total production costs [2]. In most cases, non-destructive, non-tactile automated defect inspection is accomplished through the computerized inspection of digital images captured at some point of production. The majority of the costs associated with such a system reside in the design and implementation of the sophisticated technology required for the task [1]. Usually, this setup is very product-centric, and cannot be easily ported to other applications.

It is therefore very desirable to devise a system which can easily be made to function with different product types, with little or no input from an engineer or operator, in order to minimize design and deployment costs. Because the defect recognition stage of automated visual inspection is the only one that could reasonably be thus generalized with our present level of technology, this thesis deals primarily with the formulation of a framework for this very purpose.

Two novel methods which fit within this framework are also proposed, and are compared to standard benchmark classifiers for performance analysis on a task in lumber defect recognition, as well as numeral recognition for an indication of their performance in another domain.

1.2 Thesis Organization

This thesis discusses many aspects of the automated visual inspection research area. It is organized as follows:

- Chapter 2 – Background. This chapter discusses the various aspects of defect inspection, and compares and contrasts traditional human inspection with the emerging automated inspection systems that are becoming more prevalent across industries each year. Some of the drawbacks of automated inspection are outlined, as well as the state of its publicly available research. The research goals of this thesis are summarized at the end of the chapter.

- Chapter 3 – Automated Visual Inspection. Here a detailed look into the process of AVI is offered, organized by the usual order of processing: image acquisition, segmentation, feature extraction, feature selection, and classification. The chapter closes with a summary of two works which present a systems approach to AVI design.

- Chapter 4 – Proposed Approach. This chapter identifies which components of an AVI system might readily be generalized for cross-product use. A framework is proposed for the automation of this generalization process. Two novel methods are put forward which satisfy this framework, and their parameters are discussed.

- Chapter 5 – Experimental Setup and Results. In this chapter the combinatorial optimization of the proposed methods is outlined, and results are contrasted with similarly

optimized benchmark methods. Experiments are repeated for features sets that have been pruned with a selection method. Lastly, a simultaneous search for an optimal feature-space and classifier parameters is attempted with a genetic algorithm.

- Chapter 6 – Conclusions and Future Work. Here the results obtained in chapter 5 are discussed, and conclusions are drawn. Some promising areas in which further work would benefit are outlined.

Chapter 2
Background

This chapter begins with an introduction to defect inspection, and the reasons for its importance to current manufacturing processes. The major limitations of human-based inspection systems are outlined, and ways in which automated inspection counter said limitations are discussed. Lastly, some of the difficulties associated with the research and implementation of automated defect inspection systems are highlighted, as well as the method by which this thesis attempts to address them.

2.1 Defect Inspection

Defect inspection is any process which attempts to identify errors in a product, or (in a larger sense) in the manufacturing system responsible for its creation, assembly, or handling. It is usually a critical component in any production companies' quality assurance system.

Defect inspection takes many forms depending on the product type, and its method of production. The most common method is visual inspection, which is often enough to satisfy most requirements

for quality. If surface observation is insufficient, and the product requires internal inspection, there exist ultrasonic, x-ray, and magnetic resonance methods by which defects not accessible to surface inspection might be observed [2]. Occasionally a product must be exposed to physical duress to ascertain its physical cohesion, but such forms of quality control are usually referred to separately as *testing tasks*[1].

This thesis is primarily concerned with automatic visual inspection of products through digital imaging. This method is widely employed in industry and is appropriate for a great variety of products, such as web material (paper, films, print, labels, etc), plastics, glass, steel, circuit boards, etc. [13]. It is possible however, that the method proposed here might be applied to other forms of imaging, as long as the inspection method generates two-dimensional intensity images, and even to other problem domains, since the specifics of the proposed methods reside within the larger research area of pattern recognition.

2.2 Motivation

It is detrimental and often unacceptable for a company to ship defective product to consumers; upon receiving a defective batch a customer will likely consider the option of finding a more reliable source for their needs in the future. Because of constantly increasing demands (and indeed expectations) for quality in commercial product, most producers are so intent on delivering only the highest grade of product to consumers, that it is not uncommon to see a large percentage (often in the range of 10% or greater) of overall production costs reside in defect inspection [1]. In fact, since inspection is regarded as a non value-added activity, it can easily represent the largest single cost of the entire manufacturing process [2]. Thus there is a great economic incentive for any deliverable product to meet the expected quality level.

Visual defect inspection has long been the most dominant form of quality inspection in a great variety of industries. This is largely because humans are visual beings; it has been estimated that over 90% of our knowledge of the world is acquired visually. The other main advantage of visual inspection is that it is non-destructive and non-invasive; often no tactile disturbance is required of the product, and it is thus left in a pristine state. Some product types require more than visual assessment, and occasionally warrant invasive or destructive testing, but it is avoided if possible, because obviously, product is compromised as a result, and 100% inspection cannot be achieved – a sufficient sample size is used to obtain significant statistical assurance of that particular batch's quality.

2.2.1 Drawbacks of Human Inspection

Visual Inspection as traditionally performed by human inspectors is usually undesirable for one of several reasons. Performance issues are first and foremost; production rates can easily exceed even a team of humans' capacity to fully inspect each product. In fact, sometimes even inspection of a sample of statistically sufficient size is simply not feasible by people [2].

Another key issue is consistency. Human inspectors have often been shown to perform unreliably; different inspectors assign different grades to one class of product, and in some cases, the same inspector will assign differing quality grades to the same *instance* of a product when encountered twice within a negligible amount of time (sometimes in the same shift) [1]. This is because human inspection is, perhaps even more so than other forms of labour, susceptible to issues of subjectivity, emotion, fatigue, and stress [12].

Safety can also be a key factor in deciding whether to automate inspection. There are many environments in a production process that simply cannot be inspected by a human without unnecessary - if not grave - risk, such as metal casting, power line inspection, nuclear plants, etc.

Temperature, electrical voltage, ph balance, atmospheric pressure, and radioactivity are just some of the common sources of hostility in real environments in which visual checks must be performed, rendering human inspection impractical, if not impossible [1].

Lastly, a not insignificant consideration is cost - human inspection in general can often prove quite expensive [2]. Inspectors must obviously be compensated for their labour, and are entitled to company benefits and incentives, as are all employees. This is one of the main reasons for the high cost percentage of inspection with respect to the overall production process mentioned earlier.

As an aside, the move to install an automated inspection system specifically to address this last concern can be looked upon as an issue of some moral concern. This is not a new development, as many technologies have been viewed in an unfavourable light in the past. Although it is easy to depict emerging manufacturing technologies as usurpers of the blue-collar workforce, there are other social considerations to take into account. Namely, that quality inspection is often regarded as tedious and undesirable work [12], and as such, inspectors are often lacking in motivation and career fulfillment, two factors which likely also play a role in the aforementioned issues of emotion, fatigue, and ultimately reliability. Also, one must take into account that the introduction of new technologies has consistently stimulated the creation of new areas in the educated workforce, and that historically one of the greatest causes of unemployment has been the lack of investment in emerging technology [13].

Automated inspection suffers from none of the aforementioned drawbacks. Performance can be gauged much more quantitatively than with human inspection, and has been found to be significantly more consistent over time [2]. Computers and sensory equipment certainly do not suffer from issues of subjectivity or fatigue, unless one factors in the small percentage of down time required for refitting or repairing components as necessary. Electronic components function equally well at the

end of an eight hour production cycle as at its beginning, and there certainly doesn't exist the need to replace them at the end of said shift. Safety is seldom a concern when it comes to automated systems, as they are designed specifically for the hazardous operating environment in which they are intended to operate. Lastly, costs are limited to initial system design, development and deployment, along with regular upkeep. As well, it is worthwhile to note that automated systems' costs are constantly dropping, mainly as a result of the continual decrease in cost of the complex electronic components required for their construction [2], but also because as the companies involved in manufacturing automated inspection systems continue to expand their expertise and existing resource bases, design and development costs decrease as a result. This certainly cannot be said of the costs associated with human inspection, as inspector pay rates must necessarily increase proportionately with rises in workforce inflation.

These are the primary reasons why automatic inspection has enjoyed such a boom in many industrial niches lately, and why more and more companies are investigating the possibility of automating their inspection processes [13].

2.3 Difficulties

Despite all the advantages of automated defect inspection noted above, there are some existing difficulties with the technology and its development that should be noted.

2.3.1 Research Base

Automated visual inspection (AVI) is obviously a very complex, and yet very relevant and application-specific technology. Any significant and practical advances in the field are likely very profitable, and as a result, much of the leading edge research is performed in-house by companies planning on applying the technology on a commercial basis [1]. This has left publicly available

research sources in AVI in a state of mild disarray, as companies often compete in parallel on many different approaches to the problem, while maintaining somewhat of a veil of secrecy around their work, in the hope of being the first to develop a new and marketable idea. As a result, there is much overlap in the subjects of work being performed, and a novice to the area may find themselves rereading many similar and basic concepts, while the state of the art is left guarded in company knowledge bases.

Another problem of a similar nature is that of a researcher trying to understand the source and nature of a particular product's defects. It is often the case that not only will AVI companies remain secretive surrounding the technology employed by their defect inspection system, but the companies which use said technology will also be quite hesitant to release information concerning the type, frequency, severity, and costs associated with defects of their particular product. This is understandable when one considers that it could be quite detrimental for a production company to let their customers and competitors know exactly how, why and when their manufacturing technology fails. It poses a quandary to researchers however, in that it can often be difficult to understand the true natures of said defects, and as a result, to adequately formulate a system to treat them.

Another failing of the AVI research area, is a lack of work performed on the applicability of systems to deal with multiple product types [1]. Generalizing an inspection system is an interesting and certainly profitable problem to investigate, in that it stands to reason that if a system could be devised which could adequately deal with multiple, and indeed, new product types, a major savings would result in the design, development, and deployment costs associated with said system.

Lastly, a major issue with the research area of automated visual *inspection* is a lack of sufficient effort in the interesting sub-task of visual *recognition*. Many papers treat the two terms as

interchangeable, but for the purposes of this thesis, it would be beneficial to form a clear distinction at this point.

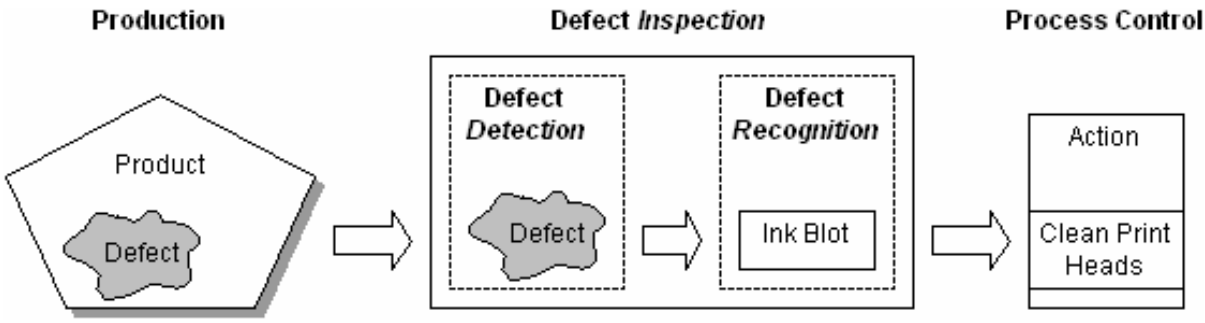


Figure 2.1: Defect inspection as an aggregate of separate defect *detection* and defect *recognition* modules.

There exist a great many papers dealing with the defect detection portion of inspection on a specific product type, but not nearly as many that try to address the issue of defect recognition, which is odd since most, if not all, interesting functionality available to process control are direct results of the ability of the defect recognition module to differentiate defect types.

Consider for example a production process whose visual inspection capability is limited simply to defect detection; we are left with products being labeled as either of quality or possessing a defect. The only actions available to process control at this point then are to let the item continue on the production path (quality product), or to eliminate it or reroute its path (defective or inferior product). Another course of action would be to have a human inspector decide what sort of defect has been identified, and consequently, what to do with the product in question, but then we are left again with the need for human inspectors.

By contrast, when it can be said with certainty not only that a defect exists (defect detection), but also to what class of defect it belongs (defect recognition), and possibly its severity, a whole new set of options in process control present themselves. The production system can in this case still pass,

eliminate or reroute the product, but it can also do it in a more logical and cost-effective manner. For example, a defect's severity can dictate whether production must cease completely for a refit of components due to a severe production error, or whether the defect is simply a minor aberration and production can continue indefinitely. Also, possibilities are raised in regards to automating process control, and the further reductions of costs and increases in efficiency that can be achieved as a result.

The lack of research available in Defect Recognition can in some cases be attributed to researchers treating detection and recognition as one and the same problem in classification; it is sometimes the case that a defect detection module is composed of several components, each hunting individually (and ideally in parallel), for defects of a specific type [14]. Therefore, if a defect is detected by an individual component, its type is automatically known (that of the type that the component was searching for). Also, lack of interest in defect recognition can be somewhat attributed to difficulties in implementation, which are discussed further in the next section.

2.3.2 Implementation

Because AVI is a very application specific domain, it is constrained by its applicability. For example, real-time implementation is usually a very high priority for any inspection system. If the system is unable to cope with the production rates of its target production process, then it is rendered virtually useless (except for theoretical study), regardless of its sophistication, novelty, or accuracy. Real-time performance is also a particular concern because it is often the case that a production process will improve in quality and/or in throughput rate (as most constantly are [1]), thus requiring flexibility and the ability to increase performance in its respective AVI system, in either dealing with different types of defect and/or dealing with a greater input rate.

A further result of AVI's application-specific nature, is the need to function properly within an industrial environment. It is often the case that a production process' environment is several orders of magnitude more hostile than that of a lab; lighting concerns, dust, dirt and other contaminants, as well as vibration are very real criteria to be considered when designing AVI for a particular plant [12]. A robust AVI system is always the goal, because as mentioned before, stability and reliability are two of the main demands made of any quality control system, whether automated or not.

Another constraint on an AVI system is its cost. In the past, this has been the main reason companies stayed with human inspectors: the cost of the complex equipment and electronics involved for an automated inspection refit was simply too great, and companies with insufficient production rates, or net profit margins simply could not afford having an automated system designed and installed [1]. As mentioned before, these costs are declining at a tremendous rate, largely as a result of the current decline in electronic components' prices. Whether to install an AVI system is now more often a question of viability, in that AVI is still inferior to humans at inspecting certain product types, but this is often the result of three dimensional concerns, or the specifics of the manufacturing environment [13], which are beyond the scope of this thesis. Nevertheless, before installation of an automated vision inspection system, a company must make certain to perform the appropriate cost-benefit analysis, in order to ascertain whether production throughput is sufficient to balance the initial investment required for the design and installation of the system, which can range in price from the relatively cheap to the exorbitantly expensive [12].

Further constraining AVI systems development are their (seemingly) inherent specificity [1]. For example, an excellent system for inspecting lumber would perform quite poorly when applied to the problem of glass inspection. There are simply too many variables for successful system crossover,

such as the nature of the products and defects involved, the physical characteristics of the production process itself, the type of lighting required, etc.

Lastly, another detriment to AVI systems research and implementation is the prevalence of rudimentary systems which are currently used for defect inspection. Production technicians and quality assurance experts are used to the existing systems, and cannot easily see beyond their limitations. A perfect example is newspaper production. Currently most newspaper presses use human inspection on samples of issues to determine process control actions. A technician will arbitrarily choose a finished newspaper every few minutes and examine it for certain visual cues as to its quality. A special hand-held instrument for the measurement of colour called a densitometer is used manually to check ink jet levels, to ascertain that the correct mix is used for optimum colour reproduction, and a simple magnifying glass is used to check marks for press alignment which is adjusted manually if necessary while production continues [22]. Depending on how severe the colour and alignment discrepancies are judged, it is decided whether or not to discard the affected copies.

Likewise, each page of the sample issue is checked for scuff, which is an ink blot that occurs repetitively on the same page in each issue, as a result of a foreign object (often a small wad of paper) stuck on one of the press rollers [22]. By the time this problem is located, and the press stopped for cleaning, hundreds of copies are often rendered useless. These are perfect candidates for automated inspection, but because the existing system has served the newspaper business for so long, there is no perceived need to go to superior systems, and thus no need to fund research for better systems.

2.4 Thesis Goals

It is the aim of this thesis to address some of the existing difficulties with AVI systems research described in the previous sections, specifically those of AVI system specificity, and the lack of

research in defect recognition, as opposed to defect detection. In fact, the two problems are dealt with in tandem, with the research question specifically being addressed as: can a system of defect recognition be devised which could be used effectively across different production platforms, and differing product types? Naturally, real-time applicability should always be kept in mind when devising new methods for AVI, but for the purposes of this work we will put aside this issue, and focus more on the theoretical concerns as formulated by the research question.

Chapter 3
Automated Visual Inspection

This chapter is a survey of some of the AVI techniques present in the literature today. It is organized roughly by the order of processing performed by nearly every AVI system, namely image acquisition, detection and segmentation, feature extraction and selection, and finally recognition. Most work has focused on one or two of these facets of AVI, to the exclusion of the others. The last section of the chapter examines two examples of works that propose complete framework approaches that include many if not all of the aforementioned subsystems, and their design considerations.

3.1 Image Acquisition

Image Acquisition is the foundation on which all AVI systems reside. It is of the utmost importance to properly calibrate the lighting and camera conditions to suit the application at hand, since without the ideal image capturing conditions, the resulting data is usually useless. It is often the case that proper care taken to adjust image acquisition conditions can lead to the avoidance of many costly image pre-processing operations that would otherwise be required [20].

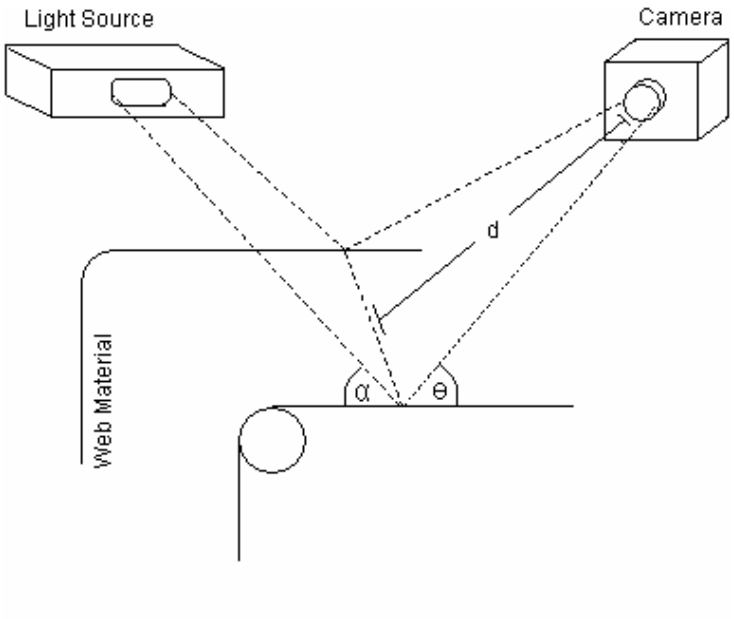


Figure 3.1: A typical line-scan camera setup for visual web inspection.

At the minimum, it must be decided at what resolution to operate, and the proper type, distance and intensity of illumination. Resolution is a key tradeoff, because the higher the resolution, the finer the details one can acquire, and the smaller the defects one can recognize. However, with an increase in resolution comes a proportionate increase in data throughput, which the system must be able to cope with. In general, it is desirable to operate at the lowest resolution capable of discerning the smallest defects one wishes to locate; any surplus in resolution is simply a burden to the rest of an AVI system.

Illumination types and methods can range from simple incandescent lighting, to more esoteric solutions for specific needs, such as specular reflection, polarized light, circular arrays of LEDs, and synchronized stroboscopic lights [2]. The method proposed here might be employed with any of these, but for our purposes it might be easier to imagine a simple overhead lighting setup, which would be adequate for most applications.

If the material is one in which surface texture plays an important determining factor in quality assessment, then camera and illumination angles (θ and α in Figure 3.1, respectively) should also be considered, to cast an appropriate amount of shadow, thereby highlighting the surface texture as required [2].

3.2 Detection & Segmentation

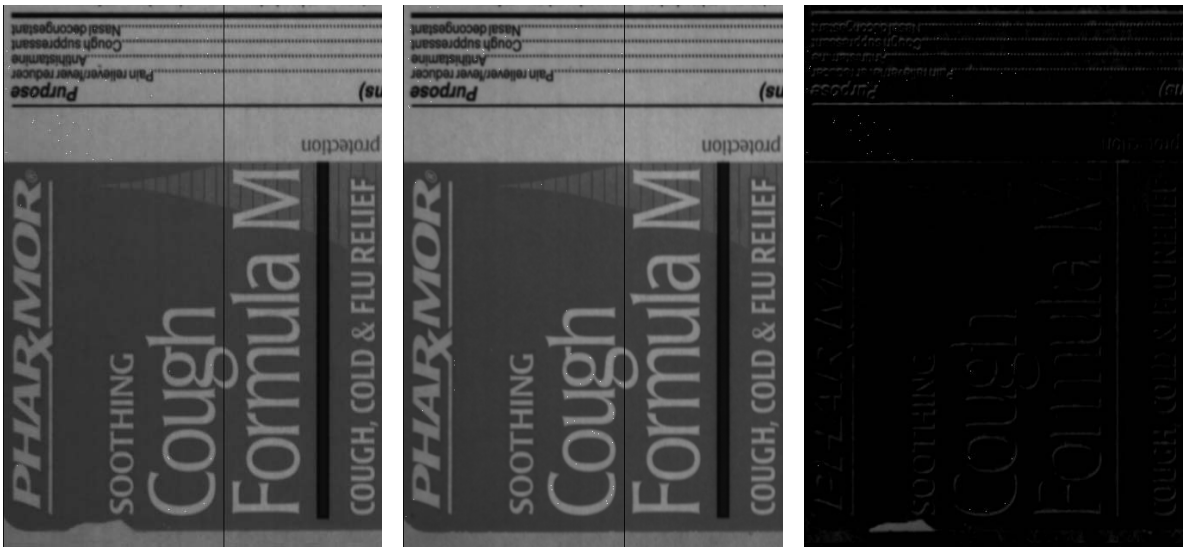
Defect detection can be thought of as an entire classification process in and of itself. Here the task at hand is to recognize and isolate defective regions of the product, and if necessary, to pass these on for further processing to analyze more specifically its type, and how it might be treated.

This segment of AVI has garnered a lot of attention, as it is the one which can most significantly impact an AVI system's effectiveness through initial design. Most products have large differences in their optical characteristics, and it is a non-trivial matter to design and implement a successful segmentation routine; it must be tailored to the specific texture and histogram properties of the product at hand, as well as to the amount of variance expected of these properties from product to product and indeed region of product to region. Because of these considerations, a successful generalized segmentation routine has not yet been put forward, nor is one expected in the near future.

Sometimes it is the case that detection and segmentation can be handled by something as trivial as static image intensity thresholding [1]. This is possible when the image of the product is relatively well behaved, that is to say that there is not much local variation in pixel intensity. This can be the case with smooth monotone products such as metals or plastics, in which defects such as holes or cracks show up quite readily on product images as much lighter or darker regions [1].

Another product type which is readily segmented with thresholding techniques is one whose precise appearance is known beforehand, and can be verified quickly thorough template matching

(image subtraction) techniques. Products such as circuit boards or labels are an example of this type, where the desired pixel intensity is known beforehand at each position of the image, and variances in these are clear indicators of a problem. Template matching brings up additional technical problems that need to be resolved, such as image registration, in which the template and defect image must be precisely aligned before image subtraction can be employed, but this is a separate concern.



a) b) c)

Figure 3.2: a) Close-up of pharmaceutical label with tear, b) its "golden template" stored in memory, c) enhanced result of image subtraction. The visible text and graphic "ghosts" are the result of imperfect image registration.

The static thresholding techniques mentioned are also readily expandable to include adaptive methods as well, which are somewhat more robust, and less prone to false alarm errors. In adaptive thresholding, dynamic cutoff values are determined by the characteristics of the local region being examined.

Sometimes detection and recognition can be bundled together into one module, in that the means to identify the defect from the background is sufficient to determine its type. Such an approach is put forward in [14], where a series of Finite Impulse Response (FIR) filters are used in parallel to isolate defects on textile materials.

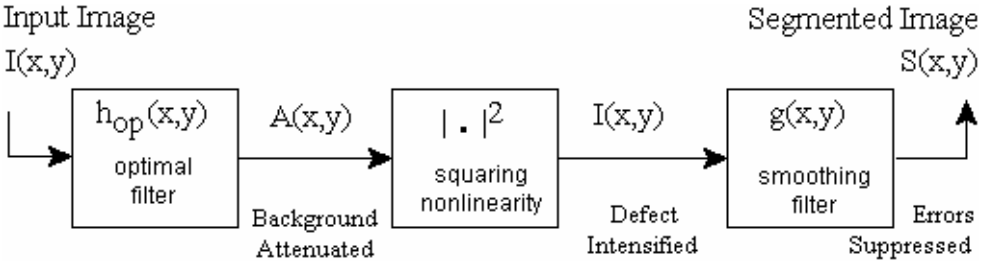
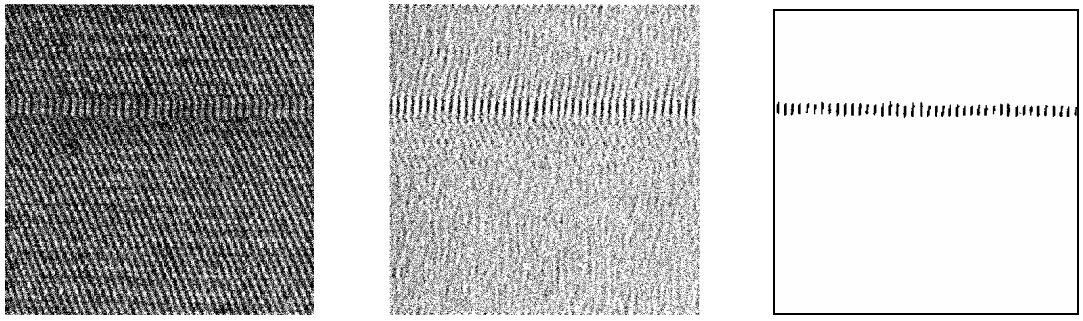


Figure 3.3: Using custom FIR filters and some simple image processing to segment a specific defect type [14].

Because it was found that each defect type tends to have predictable texture characteristics (as modeled by autocorrelation functions), FIR filters could be successfully attuned to a particular defect. By convolution with the image, a series of intensity images (one for each defect type) is then created, in which the background textures are attenuated. These images are then enhanced through a squaring non-linearity, and finally smoothed with a low pass filter to eliminate errors.



a) b) c)

Figure 3.4: a) fabric sample with defect mispick, b) attenuated output of optimal FIR filter, c) segmented output after nonlinear squaring and threshold processing.

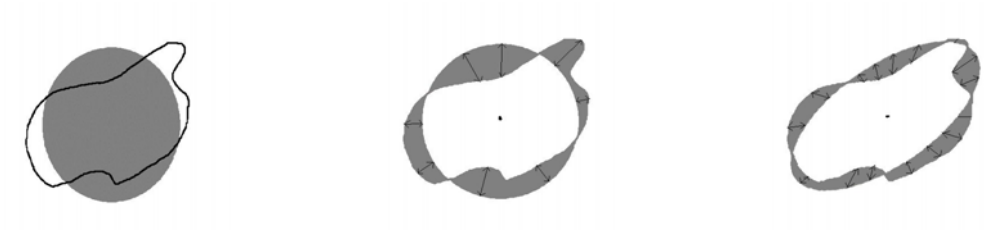
3.3 Feature Extraction

Perhaps the most important design choice for any pattern recognition system is the set of features used to describe the objects to be classified. The sophistication of the classifier to be used is rendered irrelevant if the feature set simply does not contain adequate information to separate the classes. Because of the importance of the features used for classifier performance, and because of the proposed method's delineation of features into subsets by type, we will examine some of the more popular features used for defect classification by the type of measurements they quantify.

3.3.1 Shape Description

There are many ways of describing shape, such as statistical methods, structural methods, and those based on examining an image's frequency-domain characteristics, such as analysis of its Fourier or Wavelet transform [20]. Statistical methods are quite popular in AVI, because they tend to be computationally efficient, and combinations thereof can be very effective at delineating defect classes [17].

a) A defect's contour b) Convexity c) Principal axis ratio



d) Compactness e) Circular Variance f) Elliptical Variance

Figure 3.5: Statistical shape features utilized for the recognition of paper defects in [17].

Convexity is the ratio of an object's convex hull to its perimeter. It is an indicator of any cavities in its contour.

$$Conv = \frac{P_{ConvexHull}}{P_{Defect}} \tag{3.1}$$

An object's primary axis ratio is a good measure of its elongatedness, and increases as an object strays from compact shapes into more longitudinal ones.

$$PAR = \frac{c_{yy} + c_{xx} - \sqrt{\left(c_{yy} + c_{xx}\right)^2 - 4\left(c_{xx}c_{yy} + c_{xy}^2\right)}}{c_{yy} + c_{xx} + \sqrt{\left(c_{yy} + c_{xx}\right)^2 - 4\left(c_{xx}c_{yy} + c_{xy}^2\right)}} \tag{3.2}$$

It can be calculated as a function of a contours 2-D covariance matrix C, which is defined as:

$$C = \begin{pmatrix} c_{xx} & c_{xy} \\ c_{yx} & c_{yy} \end{pmatrix} \tag{3.3}$$

Compactness is the ratio of the perimeter of a circle of equal area to the defect's perimeter. It approaches unity as the defect tends toward circularity.

$$Comp = \frac{P_{circle}}{P_{Defect}} = \frac{2\sqrt{A\pi}}{P_{Defect}} \tag{3.4}$$

Circular variance is a similar measure to Compactness, but is less sensitive to small discrepancies, in that it is a measure of the pointwise variance from circularity, and is calculated for every point in the defect's contour.

$$Circ.Var = \frac{1}{N\mu_r{}^2}\sum_i \left(\left\|\vec{p}_i - \vec{\mu}\right\| - \mu_r\right)^2 \tag{3.5}$$

Here N is the number of pixels in the defect's contour, μ_r is the defect's mean radius, $\vec{p}_i$ is a point on the defect's contour, and $\vec{\mu}$ is the defect's centroid.

Elliptical variance is similar to circular variance, in that it is a pointwise measure of a contour's variance from that of an ellipse with equal 2-D covariance matrix.

$$Ell.Var = \frac{1}{N\mu_{rc}{}^2}\sum_i \left(\sqrt{\left(\vec{p}_i - \vec{\mu}\right)^T C^{-1}\left(\vec{p}_i - \vec{\mu}\right)} - \mu_{rc}\right)^2 \tag{3.6}$$

Elliptical covariance is somewhat more computationally intensive in that it depends on the term shown below, where C^{-1} is the inverse of the defect's contour's covariance matrix, but both measures are still O(n) operations, because they depend only on the perimeters of the contours being measured.

$$\mu_{rc} = \frac{1}{N}\sum_i \sqrt{\left(\vec{p}_i - \vec{\mu}\right)^T C^{-1}\left(\vec{p}_i - \vec{\mu}\right)} \tag{3.7}$$

The five shape features mentioned above were found to be quite attractive in [17] because they were computationally inexpensive, and because they proved effective at describing paper defects when used in tandem. An additional characteristic of the above features is that their outputs are all bounded in the range [0,1], which can be quite useful when dealing with classifiers that expect input data in ranges know a priori.

In [18], 8 new features were proposed that were even cheaper computationally, and proved as good supplements to the ones already mentioned towards constructing descriptive feature sets.

a) Circular Differences b) Elliptical Differences

Figure 3.6: Efficient features detailed in [18].

They are calculated very simply as operations on the sets of pixels encompassed by the defects and the reference circles and ellipses with equal areas which are superimposed at their centers of mass.

Outer and inner differences are a measure of the defect's area which fall outside and inside of the reference shapes, respectively.

$$\Delta_O = \frac{area(S \cap \overline{R})}{area(S)} \tag{3.8}$$

$$\Delta_I = \frac{area(\overline{S} \cap R)}{area(S)} \tag{3.9}$$

Absolute differences are the sum of pixels lying in either of these regions, and can be computed very quickly as an XOR or image subtraction operation.

$$\Delta_A = \frac{area(S \oplus R)}{area(S)} = \frac{area|S - R|}{area(S)} \tag{3.10}$$

Relative differences are an indicator of whether the shape tends to reside outside or inside the reference shape's boundaries.

$$\Delta_R = \Delta_O - \Delta_I = \frac{area(S \cap \overline{R}) - area(\overline{S} \cap R)}{area(S)} \tag{3.11}$$

These measurements are all normalized by the area of the shape so that they too, are bounded on [0,1], with the exception of the relative differences, which can take on values on [-1,1]. This is easily remedied with a simple linear transform, however.

3.3.2 Texture Description

Features based on co-occurrence matrices are quite popular in AVI, because they have been thoroughly researched, and have been shown effective in a wide variety of recognition tasks [13]. Co-occurrence matrices are a compact representation of a region's second-order statistics, which have been shown in some instances to align with human texture recognition [23].

Co-occurrence matrices are populated by counting the number of times specific combinations of pixel intensities occur at preselected distances and angles.

$$I(x,y) = \begin{bmatrix} 2 & 2 & 1 & 1 \\ 2 & 1 & 0 & 0 \\ 3 & 2 & 2 & 1 \\ 0 & 2 & 1 & 1 \end{bmatrix} \qquad C_{(1,0)}(i,j) = \begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 2 & 1 & 1 & 0 \\ 1 & 4 & 4 & 0 \\ 1 & 4 & 4 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

a) b)

Figure 3.7: a) A 4x4 2-bit image and b) its corresponding co-occurrence matrix formed using a horizontal distance of 1 ([1,0] in vector form).

Through the application of some mathematical operands on a region's co-occurrence matrix, it is possible to create a set of descriptive features for that region. Some common examples include energy (or angular second moment), which corresponds approximately to a region's homogeneity.

$$Energy = \sum_i \sum_j C_d(i,j)^2 \tag{3.12}$$

A region's contrast is its co-occurrence matrixes moment of inertia about its main diagonal. It can be said to give an indication of the amount of local variation present in the region.

$$Contrast = \sum_i \sum_j (i-j)^2 C_d(i,j) \tag{3.13}$$

Entropy is often shown to be an indicator of a texture's complexity.

$$Entropy = -\sum_i \sum_j C_d(i,j) \log\left(C_d(i,j)\right) \tag{3.14}$$

A texture's mean can be an indication of a texture's regularity.

$$Mean = \sum_i \sum_j i C_d(i,j) \tag{3.15}$$

Co-occurrence based features are usually normalized by the total number of pixel co-occurrences at the specified distance in the region, in order to yield values which can be compared on the scale [0,1], and can be compared irrespective of varying region sizes and mean intensities. Texture description through co-occurrence matrix based features is thoroughly detailed in [25].

3.3.3 Histogram Description

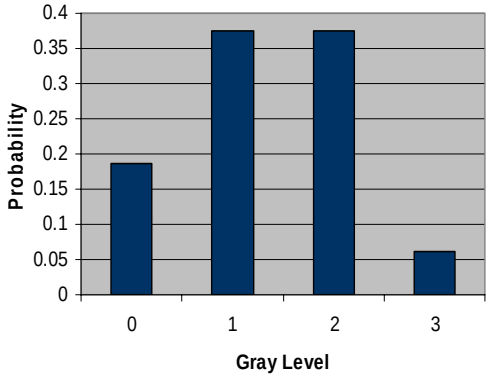
How dark a region is can often be an indicator of its type and cause. An easy way to describe a region's darkness is through the use of its histogram, which is simply the set of probabilities of the pixel intensities in the region being examined.

$$h(i) = \frac{n_i}{n}, i = 0,1,...,N-1 \tag{3.16}$$

Here h(i) is shown to represent the probability of pixel intensity n_i occurring in a region made up of n pixels falling into one of N intensities.

$$\begin{bmatrix} 2 & 2 & 1 & 1 \\ 2 & 1 & 0 & 0 \\ 3 & 2 & 2 & 1 \\ 0 & 2 & 1 & 1 \end{bmatrix}$$

a)



b)

Figure 3.8: a) A 4x4 2-bit image (n = 16, N = 4) and b) its corresponding histogram.

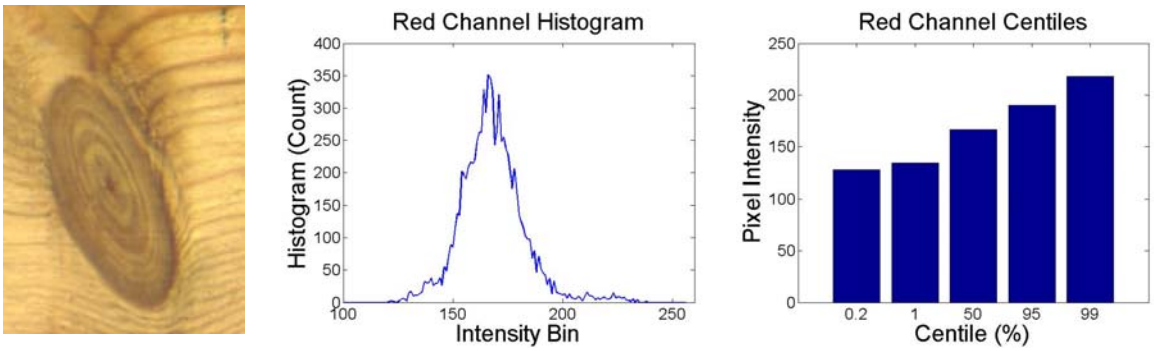
There are many common histogram measures used as features in pattern recognition applications, such as moments, entropy, mean, mean square value, skewness and kurtosis [20]. A simpler approach was used in [7], where input to one of the base classifiers was simply the histogram bin values.

3.3.4 Colour Description

Colour conveys a great deal of information for humans. Tasks such as produce and meat inspection have traditionally been very colour oriented, and new automatic systems designed for marble and gem inspection all use colour information.

As with all of the feature types mentioned already, there are a great many approaches to its implementation, but often it is the simplest ones which capture the most attention, as they are usually quite intuitive, and the cheapest computationally.

In [9] colour percentiles were used to good effect in the segmentation and classification of lumber knots. Percentiles can be thought of as inverse cumulative colour histograms, in that they are measures of the colour intensity at which a certain percentage of a region's pixels lie below.



a) b) c)

Figure 3.9: a) A lumber knot, b) it's red-channel histogram (not normalized), and c) select resultant percentile features.

For instance, in the example illustrated above, it can be seen that 0.2% of the knot's pixels have a red intensity intensity of 128 or less, and only 1% of pixels have intensities of 219 or more. This translates to the fact that almost 99% of the colour values are composed of the red band [128,219], which might be a telling piece of information concerning this particular lumber knot.

3.3.5 Product Specific Description

There are some features of a product's defects that may be specific to that particular product, and which cannot (and should not) be incorporated into systems not designed for said product. An example of this might be the relationship between the defects found in a product. An example of such a meta-defect can be as simple as counting the number of defects found on an item. If for instance, there are more than a certain threshold number, the item is defective, whereas if there are less, it

passes. Such a feature might be useful for determining grades of lumber for example, where a board with less than 3 dark knots and 1 split is still quality grade, but 2 splits or 4 knots render it inferior.

A more subtle example would be the verification of fiber count and density in filter material. In [28] a statistical model of the proper distribution of filter fibers was created, and as the material was examined, it was compared to determine its quality.

Defect location can be another telling feature. Often, a specific defect will reoccur much more frequently at a common location in relation to a product's boundaries. For example, in the examination of lumber defects, splits occur much more frequently at a board's horizontal boundaries than in its center or near vertical boundaries.

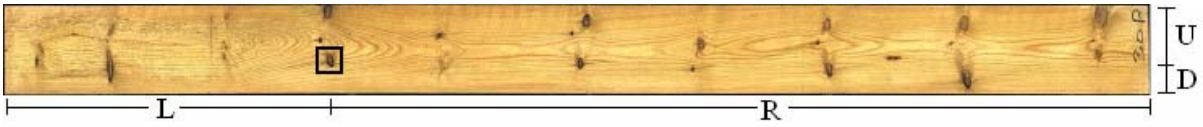


Figure 3.10: A dead knot defect on lumber with positional measurements indicated.

A simple way to encode positional features is through normalized vertical and horizontal positions (NHP and NVP).

$$NVP = \frac{D}{D+U} \qquad NHP = \frac{L}{L+H} \tag{3.17}$$

Variations of these can be used to cater to specific product and defect types, for example elongated defects might be better described by taking two measurements with respect to their vertical and horizontal extremities instead of just one from their centroids so as to get a more precise idea of whether or not they originate at the product's boundaries.

3.4 Feature Selection

Feature Selection is the process by which an effective subset of features is chosen for the task at hand. It may seem reasonable that when it comes to defect description, more features would be better, since information can only be gained with additional measurements. This is actually not the case; rather the reverse is often true. It is such a well known phenomenon it is sometimes referred to somewhat hyperbolically as the "curse of dimensionality" [16].

This is the process by which a classifier is so saturated with either redundant or contradictory measurements that it is confused, and its performance decreases. Curiously, it often doesn't require many features for this point to be reached. It usually occurs more rapidly when a classifier is presented with highly dependent features; for this reason it is usually desirable to use a feature set which is as close to independent as possible, in order to form a more orthogonal feature space [26].

In order to select an effective feature set, what is first required is a method of enumerating a candidate feature set's descriptive abilities, so that they might be relatively evaluated. In this way, one might gauge how adding or removing a specific feature affects the set's effectiveness. There are many methods of comparing feature sets, including mutual information, intra and inter cluster distances, as well as a classifier's performance on the dataset [16].

Because a complete search of an n-feature space requires evaluation of an $O(2^n)$ search space, it is easy to see that an exhaustive search is rendered unfeasible for even modestly sized feature sets. The only other known optimal solution to the problem is the branch and bound algorithm, which requires a monotonic evaluation criteria, and is also often too expensive to perform [27]. This is why the problem of feature selection is often approached using suboptimal methods which don't guarantee finding the best solution.

Two very simple examples of these are sequential forward and backward searches. The two are effectively each other's complements. Forward selection begins with an empty feature set, and increases it iteratively by always including the feature judged to boost the set's performance the most. Backward selection begins with the entire set, and keeps eliminating features which are most detrimental to performance [26]. These are rudimentary methods, and their chances of detecting a near optimal solution are quite low, because not nearly enough of the possible feature subsets are attempted, a situation attributed to the so called "nesting effect" of feature subsets.

A class of algorithms that attempts to avoid this are the (l-r) feature selection methods, which can add or remove multiple features at any iteration, but it is difficult to arrive at ideal values of l and r for any given feature set. Pudil's floating selection methods overcome this by allowing the values to alter at given points in the search [31]. It has been reported that these floating methods tend to achieve excellent results, approaching those of the branch and bound algorithm, while dominating it and other methods in terms of runtime [27].

Other feature selection methods of note are those based on stochastic genetic algorithms [6]. By encoding the feature sets as a population of binary bitstrings, genetic algorithms can randomly alter them through such evolutionary inspired operations as mutation, crossover, reproduction, and selection. By evaluation the population members' fitness' at each generation (iteration), it is possible to eliminate those not contributing to the search's goal of an effective feature set, and generating randomly those that do, thereby improving the population's mean fitness [9]. Genetic algorithms are also useful because they can be used to modify the parameters of the classifier used to operate on said feature set simultaneously.

3.5 Recognition

Deciding which class a feature pattern belongs to can be done in a number of ways. Popular methods in the literature include density based methods, such as knn classification; these are very thoroughly researched methods, and are usually quite effective and resilient to outliers. Probabilistic methods including the Bayesian family of classifiers are standard fare in any pattern recognition task, and can be shown to exhibit a minimum expected error [16]. The difficulty with the Bayesian classifiers is formulating an accurate representation of the probability distribution function, models are usually generated through Parzen windows or other estimation techniques[26]. Connectionist methods based on neural networks have also been widely explored in the literature [2][13]. Optimizing a neural network can be a difficult process in that many intangible parameters (number of hidden layers, number of nodes in each layer, activation functions, etc.) can affect its output, and cannot be guessed at in any consistently meaningful way ahead of time [6]. A long process of trial and error is usually required, and actually tuning the network through the back-propogation method is usually a time-consuming process in itself as well.

Any of the methods mentioned above can be combined into multi-classifier systems, which attempt to combine the strengths of their constituent classifiers, or minimize their weaknesses. Multi-classifier systems have been very well documented in the literature, typical decision schemes involve majority voting, the mean, max, and min combination schemes, and behaviour-knowledge space and Dempster-Shafer ensemble methods [29] [30].

An interesting multi-classifier system was put forward by Iivarinen and Visa in [7], that much of this work is based on. Here, three self-organizing maps (SOMs) [10] are used as base classifiers, and receive separate feature subsets, one each based on shape, texture and histogram features.

During training, a dataset is passed through the base classifiers for unsupervised clustering, and based on human visual inspection of the resultant groups, labels are assigned as suggested by the defects' major characteristics in each cluster. For example, if all of the defects in one shape cluster exhibited a rounded quality, that cluster might be labeled "Round". The same would apply for an elongated nature in shape, a rough quality in texture, a dark nature in histogram, etc.

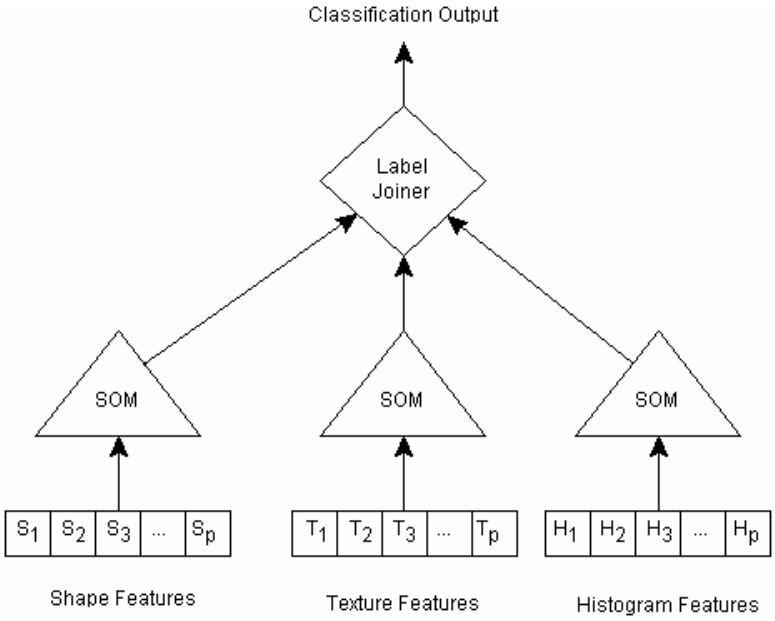


Figure 3.11: The Feature delineated multi-SOM approach explored in [7].

The final stage of the algorithm marked as a label joiner in Figure 3.11 simply appends the labels assigned by the SOM base classifiers, so that a typical final classification might be that of a "somewhat elongated, smooth, very dark" defect. This approach was pursued in [7] for many reasons, but mainly because it had proved difficult to establish a definite label for many individual samples of the defect training/testing sets by human operators, and this batch approach simplified the process. Also, it was reasoned that if new defect types were to be introduced to the system in the future, it would be easy to modify by running the training routine again, possibly with additional

nodes in the SOMs to learn the new classes presented to them. Finally, it can be seen as a solution to the problem of unreliable human ground truths; namely that they can be as unreliable as human inspection, and can be just as prone to error [12].

The disadvantage of this approach is that it could easily prove unattractive to production facilities hoping to automate an existing human-based inspection system (as is often the case). In this situation, it is not unreasonable to assume that a clear design requirement would be that the automated system must mimic the existing human classification procedure, and thus output the pre-established defect classes. The process of creating new labels for defects based on SOM clustering might therefore be unacceptable, with the unlikely exception of the labels (or some combination thereof) happening to correspond roughly to the desired output classes.

Also, it should be noted that it is not possible to quantitatively compare this system to others, since defect labels are assigned to suit the clustering tendencies of the base-classifiers employed, and are therefore not comparable with pre-assigned industry standards. These weaknesses are addressed by the classifier system proposed in chapter 4.

3.6 Framework Approaches

Most AVI work in the literature presents an idea as applied to one of the above categories; a new set of features, a unique classifier approach, or perhaps a new segmentation technique that have been created. These are generally tailored to a specific detection or recognition task, and do not port well to other problems in the AVI domain. There hasn't been much attention focused on an overall systems approach; this is probably due to the fact that most work of this kind is explored in-house by commercial companies [20].

Nevertheless, there do exist some interesting works that attempt to address the issue of designing an entire system, or organizing the components into manageable sub units for analysis and design stages. In [32] Sablatnig puts forward a very comprehensive framework for speeding up the development of AVI systems through the concept of software reuse and setup. By defining a general analysis graph for inspection that contains detail relations representing various detection algorithms, a separation of the process of detection of primitives from the model based analysis process is achieved. The result is an adaptable system for the design of AVI systems, which is demonstrated by application on the inspection of hygrometers and clocks [32].

In [28] Brzakovic et al. outline a simulated design environment for the design and testing of inspection systems for web materials. The proposed system consist of five sub-systems, roughly corresponding to the components of AVI outlined above, which can be implemented by a variety of different methods and algorithms depending on the task at hand.

1. Sensing Sub-system:

    - This component can simulate various modes of image acquisition and transforms, including resolution effects, space-variant sensing , and cortical projection in order to try and determine the most likely candidates for parameters related to optimizing imaging conditions.

2. Detection sub-system:

    - Here different image processing algorithms can be used to extract information for the inputs generated by the sensors. These include labeling and grouping techniques such as various local and global adaptive thresholding algorithms.

3. Characterization sub-system:

    - Contains the features that will be used by the decision making sub-system. It can include features similar to the ones already described, as well as many others.

4. Validation sub-system:

    - This sub-system is only active during the design stage, and is not used when the resultant system is in operation. Typically data analysis is performed here to determine the effectiveness of the features selected in the characterization sub-system, as well as the parameters required for the decision making sub-system.

5. Decision making sub-system:

    - Here recognition of the defects is ultimately performed. The classifier employed will typically be configured by the validation sub-system, and if sufficient performance cannot be achieved, then the previous sub-systems are cued for additional modifications, such as the addition of features to the characterization sub-system.

This approach to system design and evaluation was used to develop and validate an AVI system for the task of microtexture characterization in non-woven fibrous materials[28]. This design methodology is specifically of interest to us, as the generalized framework proposed in chapter 4 corresponds roughly to the Characterization, Validation, and Decision making sub-systems.

Chapter 4
Proposed Approach

It would be desirable to have available a generalized system for automated visual inspection, one that could work well with different product types, with minimal reengineering required between applications. Such a system would result in large savings in design and development times and costs. To this end, this chapter begins by examining the components of a typical automated visual inspection system. A framework is then proposed by which the description and recognition components might be generalized for operation with differing product types. Two novel methods (the so-called cluster-space and hybrid-space classifiers) are then introduced as instances of this framework. The operation of these methods is illustrated graphically, to demonstrate the feature reduction and transformation inherent to their operation. Lastly, system parameters are identified and appropriate search and optimization routines are established

4.1 AVI system analysis

To formulate a framework for a generalized AVI system – that is, one which might be used across many different product platforms – it is necessary to determine where the generalizations might be made. For this purpose, it is instructional to expand on Figure 2.1 in greater detail.

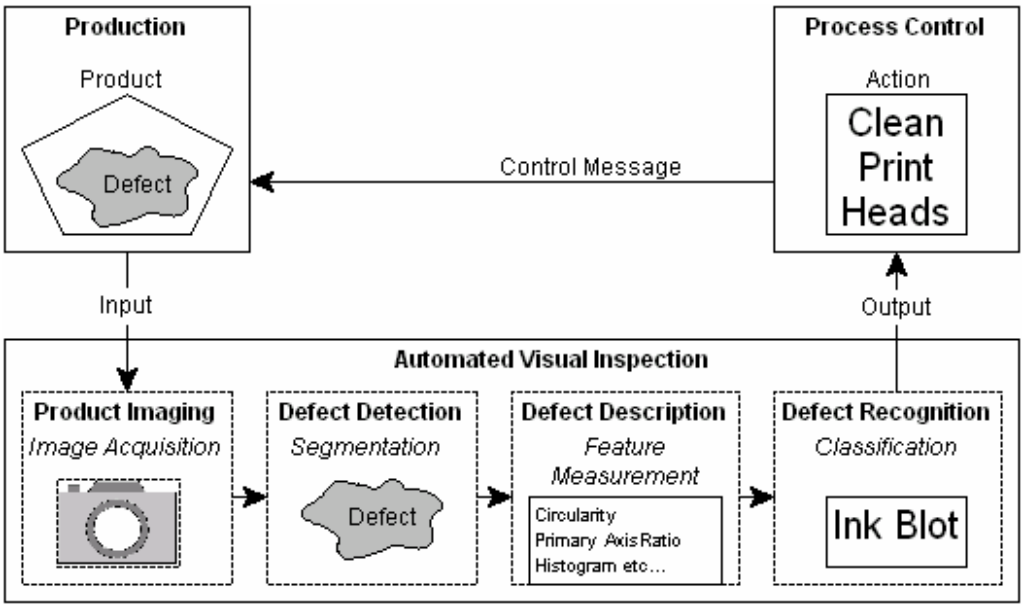


Figure 4.1: A typical AVI system and its constituent components.

It has been shown that the imaging sub-system (camera positioning and focusing, lighting, vibration compensation, etc.) is largely application-specific [12]. In general it is not feasible to devise a system which self-calibrates to meet the conditions of different manufacturing environments, as there exist too many parameters for which to compensate. A human engineer is always required to perform the fine tuning necessary to set-up an adequate platform for the acquisition of practical images at the correct resolution for the application at hand, and in fact the time and resources devoted to this task are usually well worth the effort, as many complex and time-consuming image processing operations can be avoided with some careful calibration of the lighting and image resolution settings [20].

Likewise, it is impossible at this time to devise a ubiquitous process-control system to interface with the infinite varieties of production systems. Simply speaking, if automated process-control capabilities are desired of an AVI system, the input requirements of the production control system must be assessed by an engineer, and a custom interface must be designed virtually from the ground-up if one does not already exist.

Defect detection – the isolation, or segmentation subsystem – is another aspect of AVI which is at present beyond our capability to generalize; or make adaptable to different product types. Computer routines are still a long way from being intelligent enough to be able to analyze images of several different non-homogenous images of varying products and being able to locate the defects, although there is some research being conducted in this area. Tasks as menial as locating the crack on a piece of metal as opposed to on a stone or a piece of lumber, is quite beyond the ability of present day computers, without explicit guidance from engineers in the form of a-priori domain knowledge, which must somehow describe the properties of the material being inspected.

Thus we are left with description and recognition, the two areas where the case for generalization can be made most convincingly. This is intuitively the case when one pauses to consider the fact that all two-dimensional surface defects (once segmented) effectively have the same types of visual characteristics with which they can be described: shape, colour/grayscale, texture, and such absolute information as can be deduced from the image from which it was segmented, such as size, position and orientation.

4.2 Proposed Framework for Generalization

Since all two-dimensional surface defects share the means by which they might be visually characterized, it is natural to assume that many of the features effective in separating defect classes on

one product might well be used effectively on others of an entirely different type. Thus one way to implement a generalized and automated defect recognition subsystem is to keep an ever expanding set of image features at all times. When a new application is encountered the existing features and an appropriate classifier are optimized using a search algorithm which monitors classification results (Figure 4.2). Then and only then, if classification error rates are not reduced to the required levels, an engineer must analyze the problem domain, and add novel features to the existing library as required.

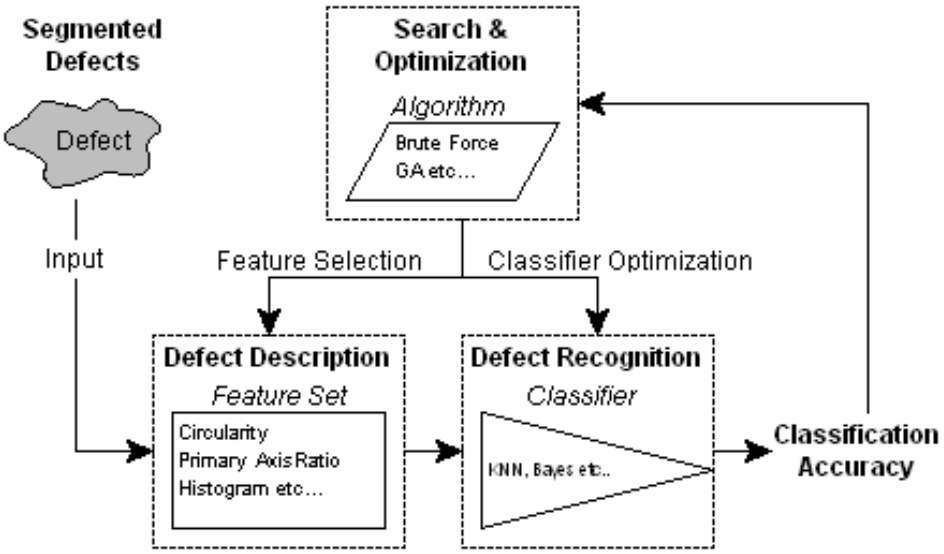


Figure 4.2: The generalized components of an AVI system which conform to the proposed framework.

An AVI system designed with this generalized system negates the need for an engineer during the initial setup and tuning phases for some applications, thereby cutting setup time and costs. In addition, as any system within this framework is exposed to more applications, its ever-expanding library of features will reduce the probability of requiring an engineer to add novel features, resulting in further cost savings with time.

4.3 Specific Method within Generalized Framework

The proposed framework might be realized in any number of specific ways. Regardless of the feature set, classifier, and search algorithms decided upon, however, one of its inherent disadvantages will persist, in that as previously unencountered applications force the addition of more and more features, the accumulated feature library may quickly grow unmanageable.

Because any algorithm's n-feature search-space increases exponentially at $O(2^n)$ [16], it can be easily seen that this system might quickly approach a limit at which searches yield either too great a runtime [9], or cannot produce a near optimal result because of the sheer magnitude of the task.

An easy way to limit the search is to use a feature-segmented multi-classifier approach, so that each classifier need only be optimized for a subset of the features, and likewise, feature selection routines need only focus on a portion of the entire feature-space. An interesting means to this end, and one which is followed in this work, is to have each classifier operate on all features of a certain logical grouping, so that all shape features are routed to one classifier, while all texture classifiers are only visible to another, etc.

This type of feature delineation was shown successful by Iivarinen et al. in the classification of paper defects [7]. Maenpaa et al. suggest in [3] that there are indeed merits to optimizing feature sets describing different characteristics independently. Separating feature sets by the characteristic they describe also has an intuitive similarity to human deduction, in that many times when encountered with a difficult classification task, one breaks the task into recognition subtasks. An overly simplified example might be: its shape is round, it has a stem, and its colour is red, therefore this fruit is an apple.

4.3.1 The Need to Conform to Existing Industrial Classifications

As mentioned in Chapter 3, classification of defects using the method described in [7] is limited by the fact that during classifier design the classes are redefined. That is, descriptive defect classes are created and assigned by examining the results of an initial unsupervised clustering of the dataset.

The disadvantage of this approach is that it would not be acceptable to any manufacturer whose design requirements include that the automated system mimic the existing human classification procedure, and thus outputs the pre-established defect classes.

4.3.2 Modifications Made

The classification method put forward in this paper in many ways is modeled on the one outlined in [7], but attempts to address this main limitation, namely, its inapplicability to quality inspection systems that require conformity to pre-assigned defect classes. The proposed solution is to replace the hard, or traditional, classifiers (SOMs) with ones based on fuzzy logic, and the label joiner with a hard classifier such as a KNN, Bayesian, or indeed even a SOM to perform the required "defuzzification" to the output classes expected. For an excellent primer on the motivation and mathematics behind Fuzzy logic, as well as numerous other aspects of soft computing, one would be well-advised to examine [6].

The key difference between hard and fuzzy classifiers is that those of the fuzzy variety do not attempt to assign absolute belongingness to one class at the exclusion of all others, but rather partial membership can be assigned to one, some, or all classes, depending on the nature of the pattern to be classified. The direct result is that each fuzzy base classifier will produce a real number (in the range [0,1], 0 indicating exclusion, and 1 complete membership) for each class, designating that object's level of belongingness to that particular class. An unsupervised fuzzy classifier method is utilized,

trained on an initial fuzzy c-means clustering of the data, because it can be seen as an analogue to the process of creating artificial sub-labels in [7]. Its advantage lies in the fact that instead of having a human supervisor inspect and label cluster groups according to their perceived information content during training (round, elongated, rough, dark, etc), the fuzzy clustering algorithm will find meaningful clusters based on the layout of the underlying feature space (if sufficiently delineated), and pass this data on to the final supervised defuzzifying classifier automatically, to let it decide into which of the preordained classes the object belongs.

4.3.3 Fuzzy C-Means Clustering

The Fuzzy c-means (FCM) algorithm [11] was utilized for the initial clustering of the data, because it is a basic and very well known form of fuzzy clustering, and there exists a large body of work documenting its specifics. In essence, it is a fuzzy superset of the classical (hard) k-means clustering algorithm [4]. Both of these methods' goals are the partitioning of a data space into a predefined number of groups, or clusters. But whereas k-means clustering assigns positive binary (total) membership to exclusively one cluster, with negative binary (no) membership to all others, its fuzzy c-means counterpart can assign membership values to one, some, or even all clusters. The process is described below.

If X is a finite set of n elements x_k, k = 1, 2…n, then the algorithm's goal is to return a fuzzy partition which describes its elements' levels of membership to each of c clusters. Each element x_k can be represented as a vector in the feature space $\Re^p$ by its p characteristic measurements (features), and each of the c clusters can be described by their center, or prototype, v_i in the same space.

$$X = \{x_1, x_2, .., x_n\} \in \Re^p \tag{4.1}$$

$$v = \{v_1, v_2, .., v_c\} \in \Re^p \tag{4.2}$$

The fuzzy partition desired is represented by means of a c x n matrix U, whose entries u_{ij} represent the degree of membership of element x_j to the partition space's *ith* cluster.

$$U = \begin{bmatrix} u_{11} & u_{12} & \dots & u_{1n} \\ u_{21} & u_{22} & \dots & u_{2n} \\ \dots & \dots & \dots & \dots \\ u_{c1} & u_{c1} & \dots & u_{cn} \end{bmatrix} \tag{4.3}$$

As mentioned earlier, a u_{ij} of 1 indicates that x_j belongs totally within cluster i while a value of 0 means just the opposite: it is not a member of that cluster at all. The particulars of the FCM algorithm call for an element's membership to all clusters always summing to 1, in a way mirroring a requirement of hard k-means clustering, namely that if an element belongs to a cluster with belongingness of 1 (total membership), it is at the exclusion of membership to all other clusters.

Therefore, after selecting the number of clusters desired c, the membership matrix is initialized with random values satisfying:

$$\sum_{i=1}^{c} u_{ij} = 1, \forall j = 1, ..., n \tag{4.4}$$

FCM clustering is based on intra-cluster distance, and thus at each iteration an expression of it is evaluated:

$$J(U, v) = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \sum_{j=1}^{n} {u_{ij}}^m {d_{ij}}^2 \tag{4.5}$$

where d_{ij} is a measure of dissimilarity between element x_j and cluster prototype v_i in feature-space $\Re^p$.

$$d_{ij}{}^2 = \left\| x_k - v_i \right\|^2 \tag{4.6}$$

and $\left\| \bullet \right\|$ is any inner product induced norm on $\Re^p$. Typically, the Euclidian distance is used for d_{ij}. Cluster centers are updated at each iteration by

$$c_i = \frac{\sum_{j=1}^{n} u_{ij}{}^m x_j}{\sum_{j=1}^{n} u_{ij}{}^m} \tag{4.7}$$

whereupon the fuzzy partition matrix is altered by

$$u_{ij} = \frac{1}{\sum_{k=1}^{c} \left(\frac{d_{ij}}{d_{kj}} \right)^{2/(m-1)}} \tag{4.8}$$

where m ϵ (1,∞) is a weighting exponent which determines the fuzziness of the partition. Typically the higher the exponent, the more membership values are shared amongst the c clusters, and the lower the exponent, the more likely it is for membership to be weighted heavily towards fewer clusters.

This algorithm continues until such iteration t wherein the dissimilarity function's (equation 4.5) progress has met a predetermined minimum threshold of improvement, ε_T, deemed sufficiently small for algorithm termination, or until a predefined maximum iteration T_{max} has been reached.

$$\left\| J^{(t)} - J^{(t-1)} \right\| \leq \varepsilon_T \quad \text{or} \quad T = T_{\max} \tag{4.9}$$

4.3.4 Unsupervised Fuzzy Classification

An intuitive way of building a classification routine based on the FCM clustering method was first outlined by Ruspini in [5]. It was proposed that a set of elements $X = \{x_1, x_2, \ldots, x_n\}$, cluster prototypes $v = \{v_1, v_2, \ldots, v_c\}$, and their corresponding c×n fuzzy partition U are made available a

priori to the classifier as base knowledge, either as created intuitively to follow a logical division by a human operator, or from a previous FCM operation on the data.

When the classifier routine is presented with an unknown element x_α, it is an easy matter to follow the fuzzy c-means algorithm as described in 4.3.3, whilst only allowing the update of $U_\alpha = u_{i\alpha} \ \forall \ i = 1, 2, \ldots, c$ while keeping X, v, and the rest of U constant. In this way, each new element x_α is classified by assigning it a membership to one, some or all clusters, through an iterative process of measuring its similarity to the cluster centers v, in a fashion similar to that of FCM clustering.

The output of the classifier ideally falls in line with that of the pre-classified training set.

4.4 Cluster-Space Interpretation

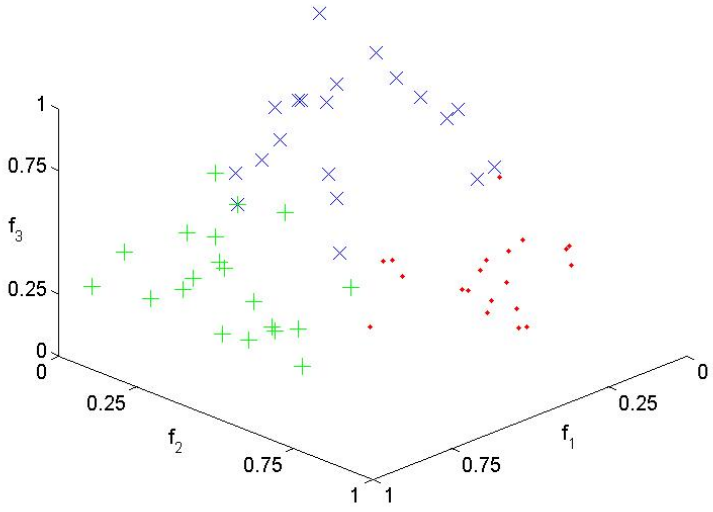
Because the c outputs from each fuzzy base-classifier (one for every cluster), will typically be much smaller than p (the number of input features used to describe each individual datum in the original feature space), an advantage of this system as opposed to a traditional classifier which deals with the data directly is the resultant reduction in feature-space dimensionality for the final classifier (a reduction of (p-c) features for each feature group). It is well documented that a reduced feature space will often yield a more effective classifier, not only because the classifier will undoubtedly run faster, but also because it can often yield more accurate results [16]. The increase in speed is due to the reduction in the complexity of the feature-space, and the corresponding decrease in operations required to ascertain and compare relations between data. The increase in classification accuracy is because often, as more features are introduced to a system, a classifier forced to make decision boundaries can become muddled by the amount of correlation and redundancies in the features, and because of the resultant complications in cluster distributions – the so-called “curse of

dimensionality"[16]. This reduction in the amount of features is therefore an advantage of our system – the final classifier will typically be exposed to a greatly reduced feature set.

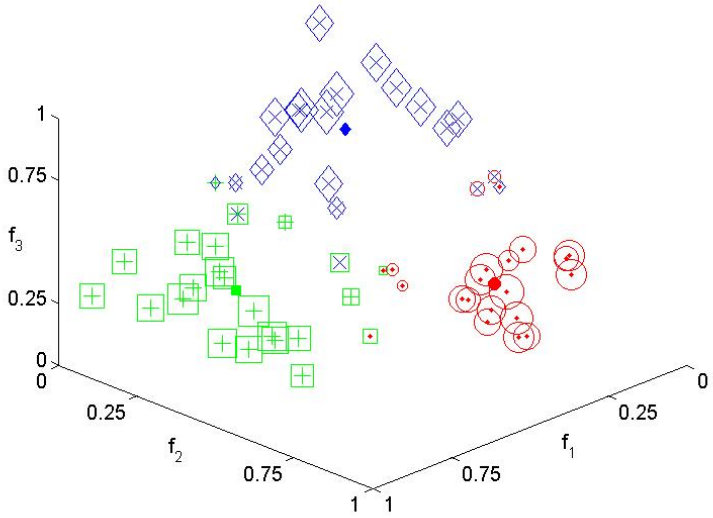
A further reduction in feature space is realized immediately however, when one reconsiders the constraints imposed upon each defects' cluster memberships by equation 4.4. Since cluster memberships of any defect α ($U_\alpha = \{u_{1\alpha}, u_{2\alpha},\ldots, u_{c\alpha}\}$) must always sum to 1, there is an inherent correlation of the data presented to the final classifier by each fuzzy base classifier. As recently stated, this is a disadvantage, since for best performance it is desired to have a classifier operating on data whose measurements are as uncorrelated as possible [16], but it is one which is easily rectified.

Data in p-dimensional space $X = \{x_1, x_2,.., x_n\} \in \Re^p$ that satisfies $\sum_{k=1}^{p} {}_j f_k = 1$, where ${}_jf_k$ is x_j's kth feature, all occupy a hyperplane in feature-space that is situated at an equal angle from all the planes formed by the major axes of the space. Because of this fact, to project the data onto one of the axis planes while preserving all of the information residing in the relative distances between data points, it is unnecessary to explicitly calculate the necessary projection transform; one must simply select one of the features, and eliminate it from all of the data-points.

To illustrate this corollary, one need only examine an example from the actual system being proposed. This will also be a useful exercise to understand the relationship between the input feature-space of the fuzzy base-classifiers, and their outputs as fed to the final classifier (from herein referred to as cluster-space). Consider a set of 60 points distributed evenly into three normal distributions in 3-dimensional space ($\Re^3$), with random means and covariance matrices. For our purposes we might imagine that the data represents three classes, each containing 20 items, all measured in the same three ways (that is, described by three features: f1, f2, & f3). This data is clustered using the fuzzy c-means algorithm (Figure 4.4).



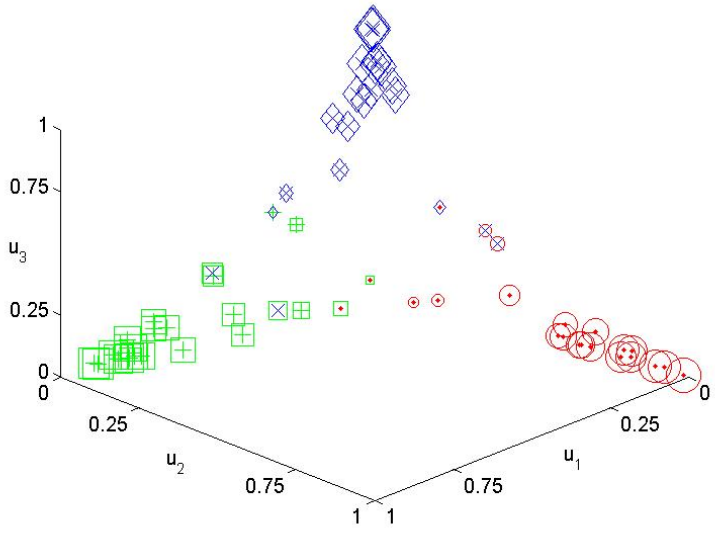
a)



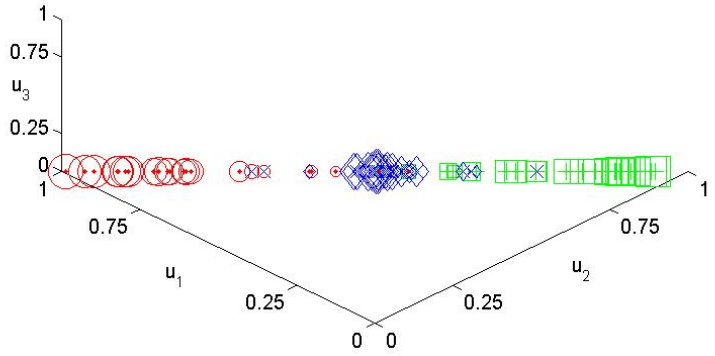
b)

Figure 4.3: a) Three normal distributions of twenty points each and b) The data as clustered by the FCM algorithm. Relative shell sizes indicate the degree of a points strongest membership (larger shells indicate greater belongingness), and solid shells indicate cluster centers as output by FCM.

In 4.3 a) each point (‘·’, ‘+’, or ‘x’) is an individual datum whose form and colour represent the class to which it belongs. In 4.3 b), this information is repeated, along with a hollow shell for each point, indicating the cluster to which that datum has been assigned by the FCM algorithm. The solid shells indicate the cluster centers as returned by FCM, while the relative sizes of the hollow shells indicate the degree of its strongest cluster membership. Upon inspection, it can be seen that the majority of the data points have been placed into clusters that agree very well with their classes. In our representation, this can be thought of as well selected features, which are quite effective at delineating the classes. Were this not the case, there would certainly be more “misclassifications” (a misnomer since the FCM algorithm doesn’t actually classify the data.)



a) Data in cluster-space.



b) Profile view of cluster-space

Figure 4.4: a) The distribution of data-points in 3D cluster-space, and (b) the same distribution as seen looking down the profile of the plane u1+u2+u3 = 1.

In Figure 4.4a), the data is shown in cluster-space (as output by the FCM routine). In cluster-space, the axis vertices near which the bulk of the data lie represent the cluster centers in feature space (since the ordered triples $\{1,0,0\},\{0,1,0\},\{0,0,1\}$ in cluster-space correspond with full membership to clusters 1,2, and 3, respectively). Not surprisingly, the misclassified data lie at a distance from all of the axes corners in cluster-space (as they do from all cluster-centers in feature-space), and thus the difficulty in correctly assigning them. Upon altering the viewpoint of our cluster-space representation and replotting (Figure 4.4b), the effects of the membership requirements described by eq.4.4 on the FCM algorithm are made immediately clear. All the data lie in the plane $u_1+u_2+u_3 = 1$, which is the hyperplane in $\Re^p$ of dimensionality p-1 discussed earlier.

By eliminating the 3rd cluster feature (u_3), corresponding to the z-axes in Figure 4.4, the data all settle onto the plane formed by the 1st and 2nd cluster-space axes (the u_1u_2 plane). Although the absolute distances are scaled by the projection, they are all scaled proportionately, which will not affect the final classifier's training-set information content. Also, it is useful to note that the ranges of u_1 and u_2 (specifically, $\{0 \le u_1, u_2 \le 1 \mid u_1, u_2 \in \Re\}$) are preserved in this way, so that any classifier requiring knowledge a-priori of its inputs' domains are unaffected.

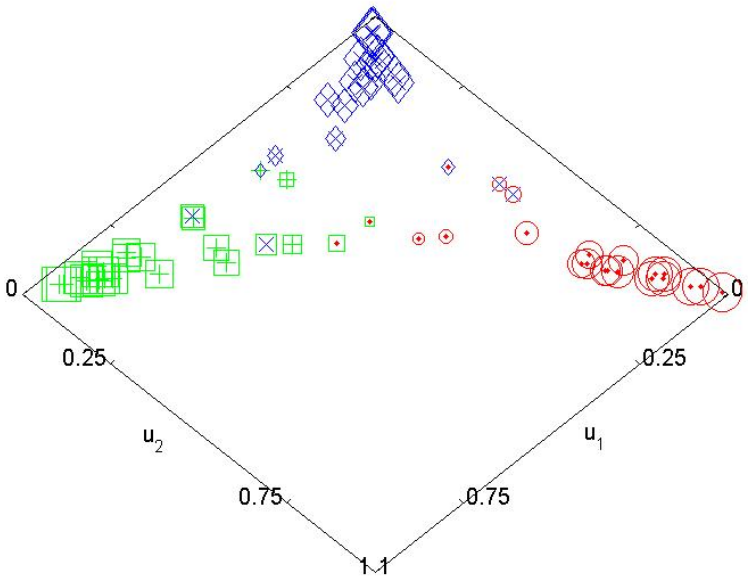


Figure 4.5: Data in cluster-space as projected onto u_1u_2 axis.

Viewing the results in Figure 4.5 and comparing them to the initial distribution in cluster-space shown in Figure 4.4 a) serves to reaffirm that the relative distributions and inter data-point distances are indeed preserved.

4.5 Final Cluster-Space Multi-Classifier

The fuzzy base-classifiers built using the parameters of the FCM algorithm therefore need only pass c-1 of their outputs to the final classifier, a further reduction in feature space.

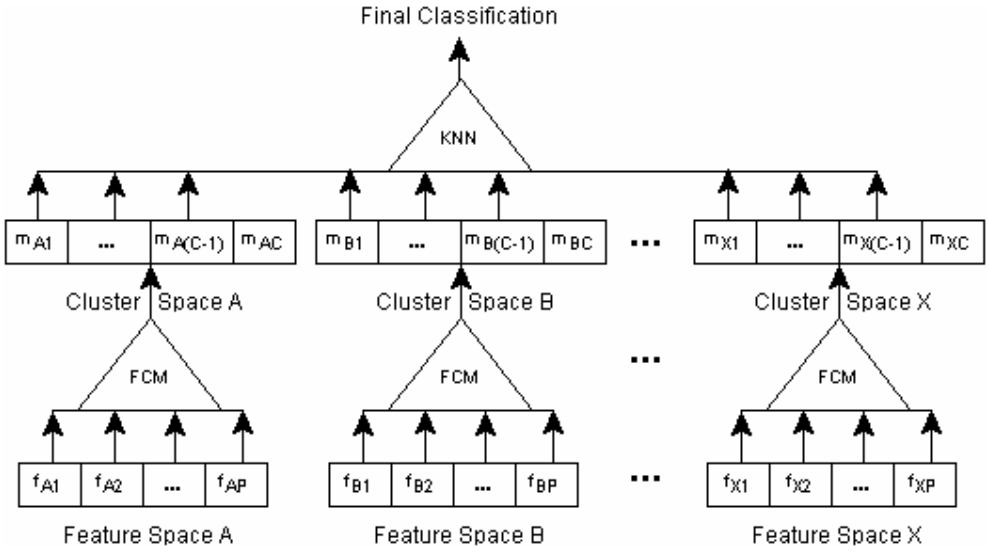


Figure 4.6: The proposed "Cluster-space" multi-classifier system.

The end result of the modifications to [7] are depicted in Figure 4.6, from herein referred to as the cluster-space multi-classifier. Note the flexibility of the system to integrate new feature pools as required by application. It should also be mentioned that each feature and cluster space can be of varying lengths, and that typically each feature space's respective cluster space will be much smaller ($c<<p$). In essence, the final classifier will have as input a feature space much different from the one afforded the fuzzy base classifiers, the result of some highly non-linear transforms to the original spaces. It will be presented with a cluster-space of sorts, one which holds information about each defect's level of belongingness to certain sub clusters that each base classifier has established in the structure of the data of its own feature space.

A typical shape base-classifier with two cluster outputs will thus likely output the degree of a defect's roundness, vs. its tendency towards elongation, while the histogram sub-classifier might relate the defect's darkness vs. its lightness. The actual sub-classifications are determined based on the structure of the underlying feature-spaces presented during the classifiers' training (FCM) stages,

and are not necessarily the system integrator's concern. If uncomfortable with such a black-box approach, and observation of the sub-classifiers' operating "rationales" is desired, a look into each sub-cluster's composition can easily be afforded as in [7], simply by grouping all of the defects in the training or testing sets by their most significant cluster membership, which should correspond roughly to the group to which they most belong.

4.6 A Fuzzy Hybrid Approach

It is conceivable that feeding only membership values to the final classifier might result in a significant information loss, namely that information stored by the relationships described in the conventional feature-space.

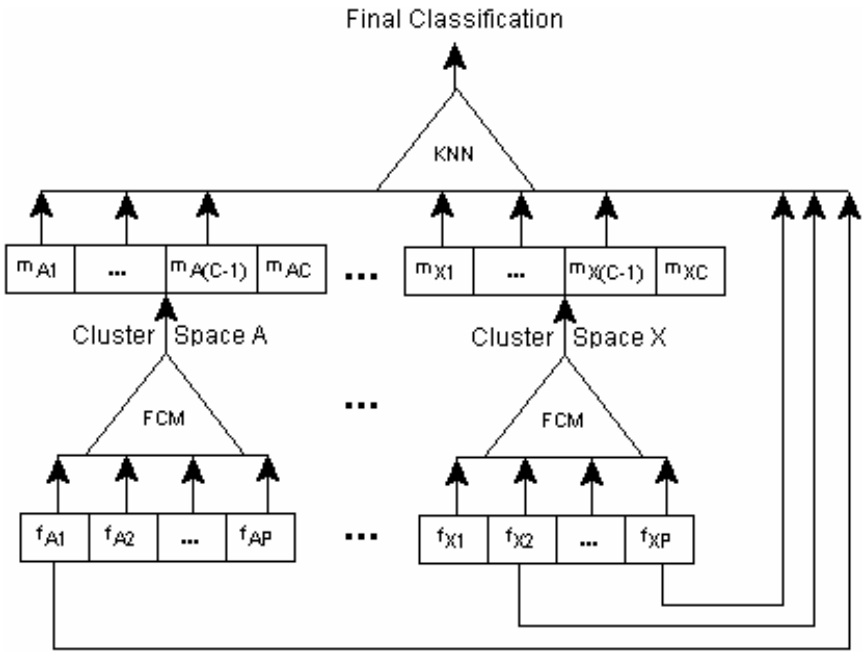


Figure 4.7: The hybrid technique proposed.

To combat this possibility, a logical extension of the cluster-space multi-classifier is put forward, one in which some of the traditional features bypass the fuzzy sub-classifiers and are fed directly to the

final defuzzifier. Here a new set of parameters is introduced: that describing which of the traditional features should be fed through to the end classifier along with the fuzzy group memberships as output by the fuzzy sub-classifiers.

4.7 System Parameters and their Calibration

The proposed systems are characterized by many parameters which require optimization, above and beyond those of the existing parameters required by the traditional (hard) final classifier used to defuzzify the base classifications.

They are:

1. $c_f \forall f \in 1,2..N$, the number of classes into which to classify defects in each feature sub-space f.
2. $m_f \forall f \in 1,2..N$, the exponent with which to raise each feature sub-group F's fuzzy partition U.
3. ε_{T1} and ε_{T2} the minimum improvements of the FCM's dissimilarity function.
4. $T_{\max 1}$ and $T_{\max 2}$ the maximum iteration allowed before forced FCM termination.

Because it is desired that the system be easily adaptable to different product types (and therefore different feature-space distributions), no presumptions are made of the feature-space structure, and a brute-force combinatorial search is executed to find the best values of c, within the bounds c Є 0 | [2,C_T], where C_T is the total number of pre-existing defect classes. Two is the minimum number of clusters required by the FCM algorithm, while zero indicates exclusion of the feature subgroup entirely. An inherent assumption is made here that the feature-subspaces will not be readily delineable into more clusters than the number suggested by the existing defect classifications, which

is not necessarily the case, but is adopted to keep optimization search times down. Also assumed is that the number of feature sub-groups N is known, but this is a valid assumption, since it will always equal the total number of subgroups existing in the system up to that time (it may grow with the addition of new feature groups).

There is no established way to select m_f for any and all occasions [4], and therefore it will also be determined using a brute-force approach, limited by $m_F \in [2,9]$ to keep search times reasonable. An m_F of 2 is used as the minimum because it results in the relatively well-known Dunn's functional [15], and because it avoids lower limits which require the computationally expensive evaluation of fractional exponents. An upper boundary of 9 was enforced because it was deemed to represent a sufficiently fuzzy cluster-space. Higher exponents tend to produce more heavily concurrent memberships amongst classes, effectively resulting in a reduced and less discriminatory cluster-space.

Parameters ε_T and $T_{\max}$ are included as pairs, because one is required for each of the clustering (training) and classification (testing) stages of the base classifiers. This is because the respective dissimilarity functions are not comparable over phases, since during training all cluster centers and memberships are changing, while during testing, only the cluster memberships of individual test-points are considered. In order to simplify matters, both T_{max} parameters are set to 1000, so that they are rendered negligible, and the number of iterations for which the FCM algorithm will run is solely decided by its respective ε_T .

Although ε_{T1} and ε_{T2} control the stopping point for very different evaluation functions, there should exist a meaningful relationship between them, since they both describe adequate stopping conditions under which the fuzzy partition is deemed adequate. Isolating and exploiting this relationship to aid in their selection is difficult, however, because the scale of the dissimilarity

functions' outputs vary greatly with m_F (the exponent of the fuzzy partition), and the type of feature-space. Also, since the dissimilarity functions behave as inverse exponentials, proper cutoff points cannot be easily selected through graphical analysis.

Ultimately, because varying ε_{T2} as a function of ε_{T1} proved to yield an average change of under 2% in the accuracy of the resulting classifier (and in no predictable fashion), setting them to be equal in each sub-classifier seemed appropriate. They were allowed to range in $[10^{-3}\ 10^{-15}]$, with a search step size of 10^{-3}.

Chapter 5
Experimental Setup and Results

This chapter presents and discusses the various experiments which have been performed to ascertain the proposed methods' effectiveness. Cluster-space and hybrid-space classifiers are first compared to standard Bayesian and KNN benchmark classifiers as well as majority and mean-rule multi-classifiers in the recognition of lumber defects. They are evaluated over the entire feature set, logical feature sub-groups, and subsets of both as a result of a standard feature selection routine.

These experiments are then repeated with a separate dataset of hand-written numerals, to quantify the method's effectiveness in another pattern recognition domain.

Finally a genetic algorithm is used to optimize the proposed systems as well as a benchmark in an attempt to evaluate both feature-set and classifier parameters simultaneously. All trials are performed in a windows based MATLAB environment, using the PRTools 3.0 pattern recognition toolbox [19].

5.1 Application – industrial lumber defects

The inspection of lumber is an important part of its manufacturing process. It has been referred to several times in this work, and is used as an example application in [33]. The task explored here is the differentiation of defect types (defect *recognition)*, once segmented, whereas [33] used SOMs in an attempt at both *detection* and *recognition*.

5.1.1 The DataSet

The dataset utilized consists of 281lumber defects which fall into one of six classes:

Defect Class	Instances
Sound Knot	50
Dead Knot	50
Black Knot	31
Pitch Pockets	50
Straight Shake	50
Core Strip (Heartwood)	50

Table 5.1: Segmented lumber defects.

It was constructed as a subset of the lumber dataset employed in [33], which is available online [34]. All defects used were manually segmented from the board images establish human ground truths for the shape of the defects being classified.

The features used were organized into 5 feature subgroups as shown in Table 5.2.

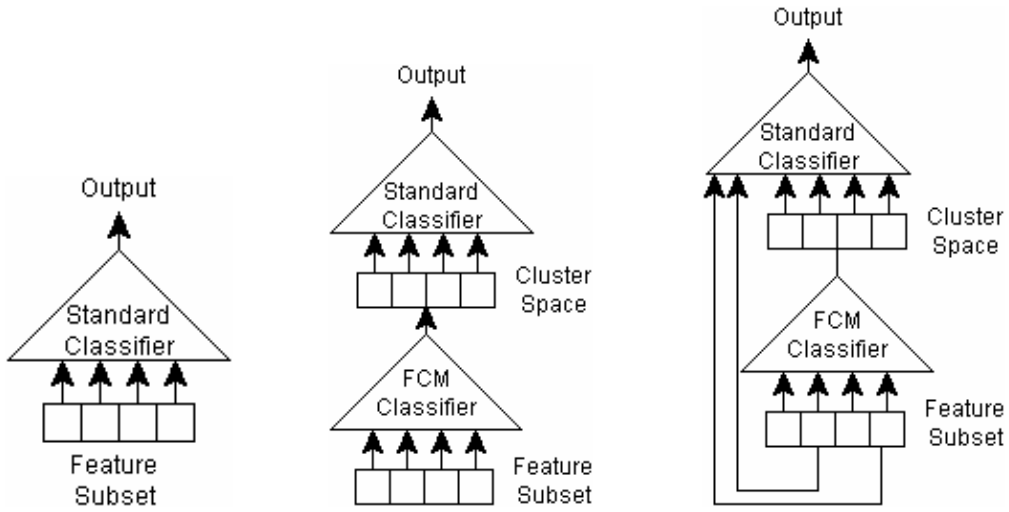
Subgroup	Shape	Texture	Histogram	Position	Colour
	Principle Axis Ratio	Contrast	Bins 0-7	Inclination	Red Percentiles 70, 80, 90, 95
	Convexity	Energy	Mean	Area	
	Circular & Elliptical Variance	Entropy		Center Coords.X & Y	Green Percentiles 70, 80, 90, 95
	Compactness	Mean		Top & Bottom	
	Circular & Elliptical, Inner, Outer, Relative and Absolute Differences	At distances of {(1,0), (2,0), (0,1), (0,2), (1,1), (2,2), (-1,1), (-2,2)}		Left & Right	Blue Percentiles 70, 80, 90, 95
	Circularity				
Total	14	32	9	8	12

Table 5.2: Organization of defect dataset into 5 logical feature subsets.

All features were normalized linearly to lie in the range [0,1], so that the Euclidean based FCM clustering would not be overly affected by some features at the expense of others.

5.1.2 Feature Subsets

The first experiment examines how the proposed system will compare against a regular benchmark classifier when performing on the logical feature subsets. No feature selection was employed; the entire feature subsets were used as is, with no attempt at optimization.



a) b) c)

Figure 5.1: Experimental setup for comparing classifier types on feature subsets. The classifier types used are a) standard classifier, b) cluster-space classifier, c) hybrid-space classifier.

Here the Standard classifiers are KNN, Bayesian, and Parzen density based Bayesian classifiers. The benchmark results are shown only for the optimal classifier, as determined by an appropriate combinatorial search for parameters, using 3 randomly selected train/test set combinations. The results listed are then the mean accuracy rates of the best classifiers on 20 random feature train/test sets, including the 3 on which the optimal parameters were determined.

KNN classifiers with K values ranging between 1 (a simple nearest-neighbour classifier) and 9 were tried. Bayesian classifiers were used whose two regularization parameters r, and s were varied in a combinatorial search between 0 and 1 with step sizes of 0.2.

$$\{ 0 \le s, r \le 1 \mid s, r = \frac{i}{5} \mid i = 0,1,...,5 \} \tag{5.1}$$

These parameters were used to form an estimation of the dataset's covariance matrix by

$$C' = (1\text{-}r\text{-}s)*C_t + r*\mathrm{diag}(C_t)*I_c + s*\mathrm{mean}(\mathrm{diag}(C_t))*I_c \tag{5.2}$$

Where C' is the resulting estimated covariance matrix, C_t is the training set's covariance matrix, I_c is the identity matrix with equal dimensions to C_t, diag() returns the diagonal of a matrix, and mean() returns a mean.

The Parzen based classifier employed no parameters; its smoothing parameter was automatically determined by analysis of the training set of each train/test set pair.

The proposed cluster and hybrid space methods are optimized in a similar combinatorial fashion, but with more intermediate steps. Firstly, as mentioned in Chapter 4, the base FCM classifier parameters must be established. This is done by a combinatorial search in the space $c_f \times m_f \times \varepsilon_{T1}$ for each logical feature subset f. Here c_f (the number of clusters into which to group the data in each feature subset) is allowed to vary between 2 and C_T (the total number of classes present in the data), m_f (the exponent to which to raise the fuzzy partition) can take on values between 2 and 9 (a value deemed "fuzzy enough"), and ε_{T1} is limited between 10^{-3} and 10^{-15} with a step interval of 10^{-3}.

The best combination of these three parameters is determined by performance over the same 3 train/test set combinations used to optimize the benchmark classifiers, using an arbitrary defuzzifier (Bayesian, r = 0, s = 0). The results are noted in tables 5.3 and 5.4.

	Cluster Number (c_f)				
Feature Group	2	3	4	5	6
Shape	6	8	9	7	6
Texture	8	3	7	9	6
Histogram	3	9	8	7	6
Position	3	5	4	2	2
Colour	3	9	8	7	3

Table 5.3: Best m_f values for each combination of c_f and feature subgroup.

	Cluster Number (c_f)				
Feature Group	2	3	4	5	6
Shape	1.00E-03	1.00E-06	1.00E-06	1.00E-09	1.00E-03
Texture	1.00E-15	1.00E-03	1.00E-03	1.00E-12	1.00E-03
Histogram	1.00E-03	1.00E-03	1.00E-03	1.00E-03	1.00E-03
Position	1.00E-06	1.00E-09	1.00E-09	1.00E-06	1.00E-03
Colour	1.00E-03	1.00E-12	1.00E-06	1.00E-06	1.00E-06

Table 5.4: Best ε_{T1} values for each combination of c_f and feature subgroup.

The defuzzifying classifier's parameters are then determined by running a search identical to the one described for the optimization of the benchmark classifiers, using the output of the best FCM base-classifiers found previously.

The hybrid-space classifiers are formed by running an exhaustive combinatorial feature-space search using the optimal cluster-space, with the addition of some of the most "promising" features, as determined by Pudil's forward floating selection [31] of 5 features out of the feature space. That is to say that the cluster space is enhanced by one of the 32 permutations possible of those 5 best features. Although it was desired to use an unbiased measure of feature suitability with the feature selection algorithm, it was found that the sums or minimums of Euclidean and Mahalanobis distances yielded poor features sets, and therefore a Bayesian classifier's performance over a disjoint test sets (r = 0, s = 0), was used as the feature suitability measure.

This may be seen as giving an advantage to the Bayesian classifier when the final tests are performed, but was necessary to gain the accuracy increase desired of the hybrid-space classifiers.

The results listed in Figures 5.1 and 5.2 show the mean accuracy results of running the best cluster and hybrid-space classifiers over the same 20 train/test sets as the benchmark classifiers.

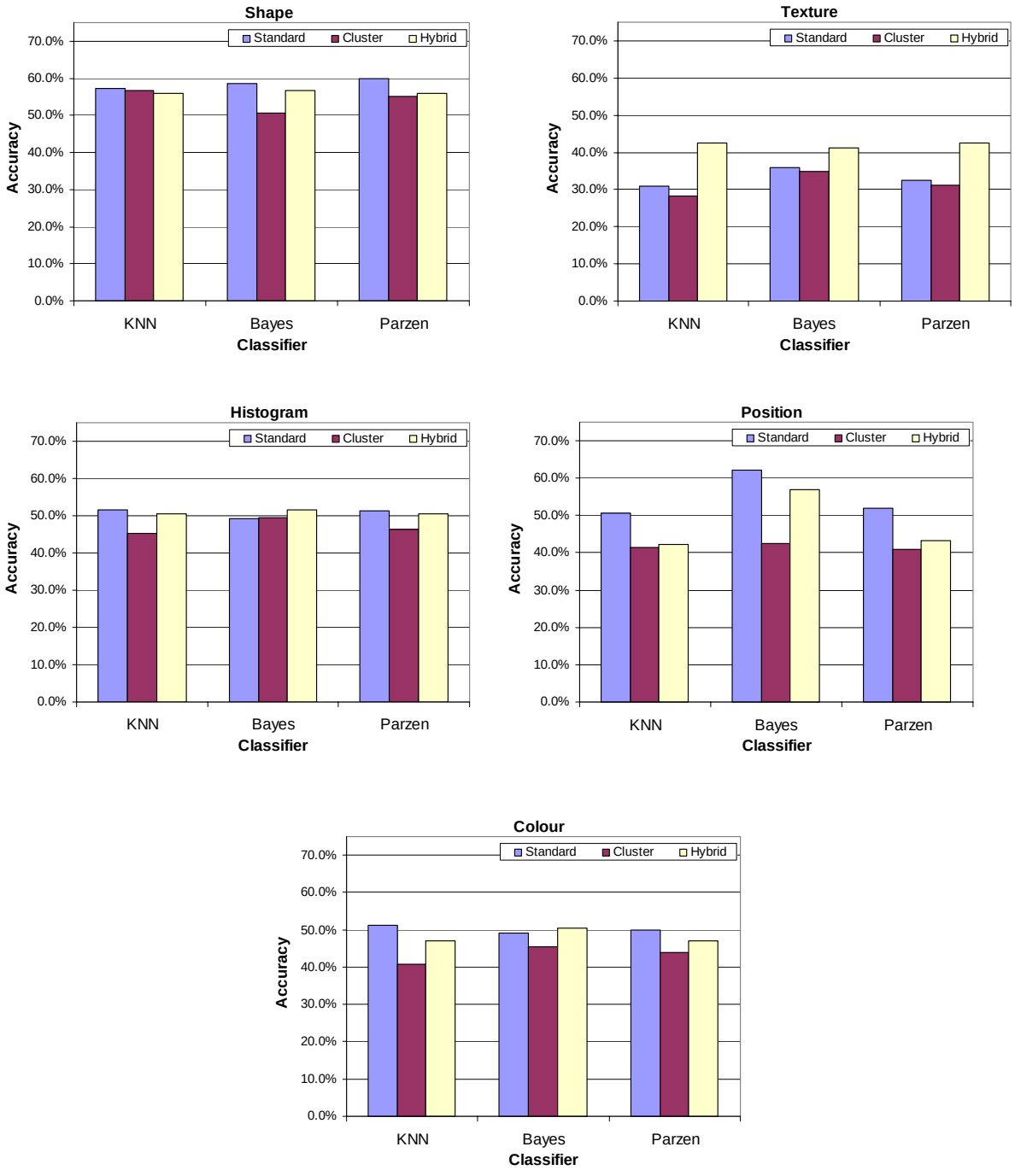


Figure 5.2: Classifier performances over defect feature subsets.

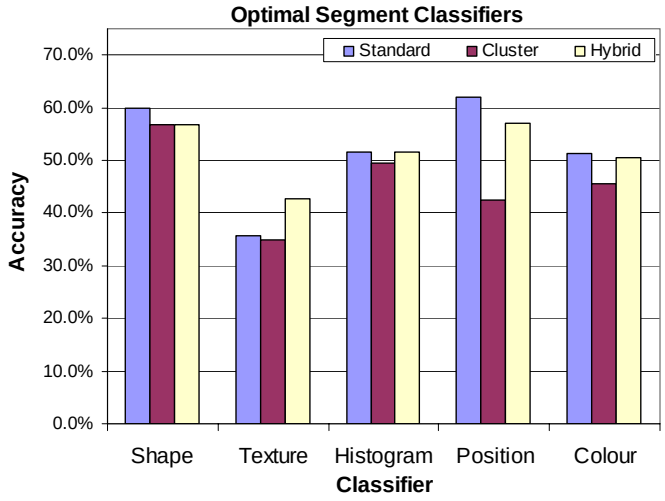


Figure 5.3: Summary of best performances over defect feature sub-groups.

There are three main observations to take from these results. Firstly, it appears that the shape and position subgroups contain the most relevant information, as their classification rates approach 60%, which the others hang near 50% accuracy. Second, most of the subgroups are dominated by the benchmark performances, which seems to suggest that the proposed methods (in their single-classifier forms) lose too much information content when transforming from the native feature-space into the new cluster-space.

Thirdly, the texture subgroup, which exhibits the poorest classification performances, is the one in which a proposed method (the hybrid classifier) considerably outperforms the benchmark. This is probably due to the large amount of redundancy contained in the texture subgroup, whose 32 features (the most of any subgroup) are simply the same co-occurrence measures taken at different pixel resolutions and orientations. This would suggest that the proposed methods (at least when using only one FCM sub-classifier) are most useful in a poor feature space, which may not be optimized in another manner.

5.1.3 Entire Feature Set

A performance comparison was performed by incorporating all of the feature subsets, effectively exposing the classifiers to all features simultaneously.

Here the classifiers compared were the standard KNN, Bayes, and Parzen density based classifiers already described, as well as 2 varieties of multi-classifiers based on those components, using the majority vote, and the mean ensemble rules. The multi cluster-space classifiers were optimized by running a combinatorial search on the 5 best cluster-space classifiers with unique cluster-numbers, that is, on the best cluster-space classifiers with the c_f 's of 2 through $C_T = 6$. The hybrid classifier was then assembled by enhancing the best multi cluster-space classifier with a permutation of the best available features (this time 10 of them). Once again, Pudil's floating forward selection was used to determine 10 of the best features, whose 1024 combinations were then used to augment the best multi cluster-space classifier.

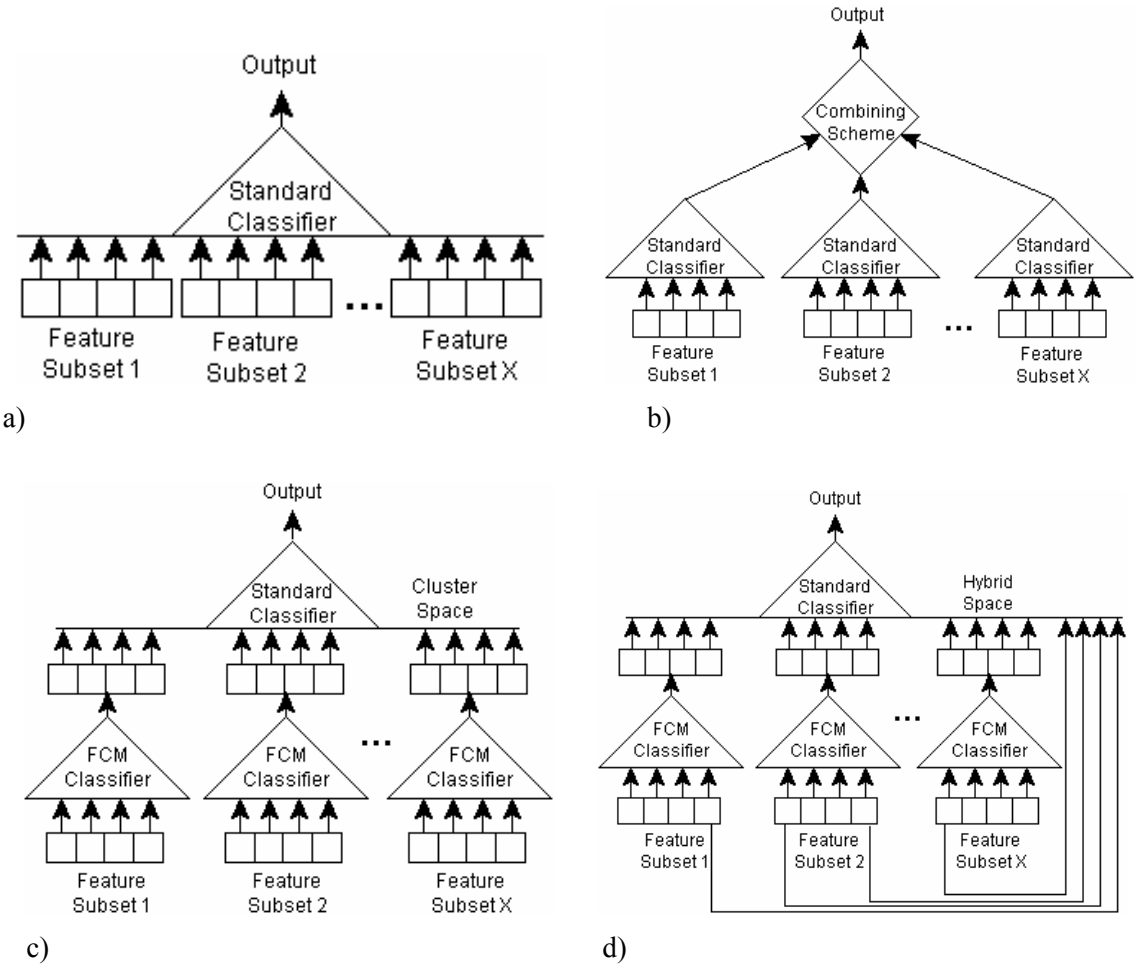


Figure 5.4: Classifier types compared over entire feature set: a) standard classifiers, b) standard multi-classifiers formed with majority-vote and mean aggregating techniques, c) proposed multi cluster-space classifiers, and d) proposed multi hybrid-space classifiers.

The steps of the search for optimal multi cluster and hybrid space classifiers are summarized below.

1. Utilize best values of m_f, and ε_{T1} for each allowed value of c_f ($2 \le c_f \le C_T$) in each of the N feature subgroups, as found previously during the subgroup searches (recorded in tables 5.3 and 5.4).

2. Perform combinatorial search in space defined by

$$\{c_1 \times c_2 \times \ldots \times c_{N_C} \mid 2 \le c_f \le C_T, 1 \le f \le N\} \tag{5.3}$$

   evaluating performance by classification accuracy over 3 random train/test sets using Bayesian classifier for defuzzification (r = 0,s = 0).

3. Determine best defuzzifier by running parameter searches on KNN, Bayesian, and Parzen density based Bayesian classifiers over resulting cluster-space.

4. The best cluster-space classifier found in step 3 is tested over 20 random train/test sets; the mean result noted.

Algorithm 5.1: Incremental combinatorial approach to determining parameters for multi cluster-space classifier.

The iterative combinatorial search for the multi hybrid-space classifier builds on the result of the multi cluster-space search.

1. Determine 10 most promising features by performing Pudil's forward floating selection using Bayesian performance over disjoint test set as criteria.

2. Perform exhaustive search on hybrid-spaces formed by the supplementation of best cluster-space found previously in cluster-space search, with all permutations of 10 direct features found in step 1.

3. Determine best defuzzifier by running parameter searches on KNN, Bayesian, and Parzen density based Bayesian classifiers over resulting hybrid-space.

4. Best hybrid-space classifier found in step 3 is tested over 20 random train/test sets; the mean result noted.

Algorithm 5.2: Incremental combinatorial approach to determining parameters for multi hybrid-space classifier.

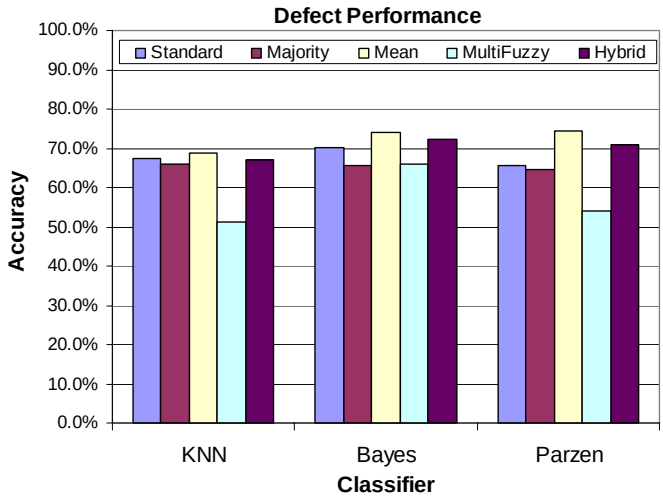


Figure 5.5: Classifier performance over entire defect feature set.

As expected, a marked performance increase was observed over those classifiers which operated on only the individual feature subsets. The hybrid-space multi-classifiers met or exceeded the performance of the standard classifiers, and were only bested by the mean aggregated multi-

classifiers. Still of note too, is the rather plain performance of all the classifiers overall (maximum 75% accuracy), which would indicate either a poorly selected feature space, or unreliable human ground truths (a hypothesis postulated in both [7] and [13]).

Feature Type	Cluster Number	Direct Features
Shape	3	Compactness
Texture	2	Energy01
Histogram	5	Histogram Mean
Postion	5	Inclination & Top
Colour	2	Percentile Blue70
Total Features	12	18

Table 5.5: Number of features fed to defuzzifier of multi cluster and hybrid-space classifiers.

Table 5.5 illustrates one of the key advantages of the proposed systems; the 75 features used to describe the lumber defects are transformed into a much lesser number. In the case of the multi hybrid-space classifier, this reduction did not lead to any information loss, as it performed just as well, if not better than its corresponding benchmark (disregarding the mean-rule multi-classifier). This could be a key issue in the case of a classifier whose computation time increases dramatically with an increase in feature size.

5.1.4 Feature Selection Incorporated

Because the optimization of the number of cluster groups to pass on to the defuzzifying final classifier in both the proposed methods might be seen as a form of feature selection, the tests were repeated using an optimized subset of the feature sets, in order to negate that advantage.

Pudil's floating forward selection was again employed, this time to obtain optimal feature sets, one using the entire feature set, and the other using the logical feature subsets as starting points. Standard classifiers were used on datasets defined by the feature selection on the entire feature set, as well as on the feature set formed by joining the optimal feature subsets as selected by the selection algorithm

operating on the feature sub-groups independently. The majority and mean aggregated benchmark classifiers, as well as the multi cluster-space and multi hybrid-space classifiers were only exposed to the latter feature sets, because they were more optimally suited, considering that each sub-classifier sees only its corresponding sub-group's features. Also, because the feature selection routine operating on the entire feature set could output a feature set with only one or two (or even none) features in a sub-group, it would be redundant to apply a cluster-space transform in this case. Appendix C lists the features selected for this test.

The benchmark classifiers, as well as the proposed methods were optimized and tested as described previously, the results shown in Figure 5.6. Here the "Standard Entire" classifier refers to those benchmarks exposed to the feature set formed by the floating search using all of the available features, while the "Standard Subgroups" classifier is the one which operated on the feature set resulting from the union of the individual feature sub-group selections.

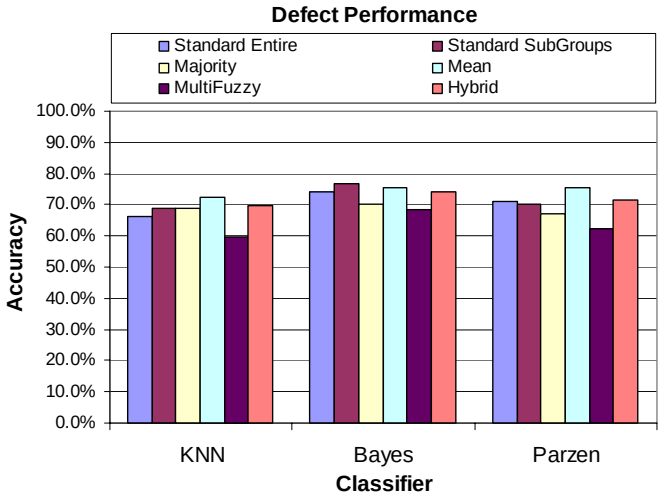


Figure 5.6: Performances of classifiers on feature sets selected by Pudil's floating forward selection.

Again, it can be seen that the multi-hybrid method for the most part meets or exceeds the performances of the benchmark classifiers, with the exception of the mean aggregated multi-classifiers, which again appeared to dominate performances. One notable development seen here is that the standard classifier using the feature sets optimized by subgroup outperformed its counterpart (which used the feature set optimized in its entirety) in two of the three cases (KNN and Bayesian). This seems to suggest again, (as in [3, 9]), that there is definitely merit in optimizing features independently by their type. This aspect of feature space optimization should probably be investigated further.

Feature Type	Cluster Number	Direct Features
Shape	3	Absolute Circular Difference & Outer Elliptical Difference
Texture	3	Entropy-11
Histogram	2	Bin3
Position	5	Inclination & Bottom
Colour	4	Percentile Blue95
Total Features	12	19

Table 5.6: Number of features fed to defuzzifier.

Once again, using the multi and hybrid-space transforms significantly reduces the number of features that the defuzzifier must operate on.

5.2 Application – character recognition

The defect dataset used in this work appears ill-posed for high-performance pattern recognition purposes, with a maximum recognition in the neighbourhood of 75% apparently possible. This is due either to the selection of features available (which may not have been up to the task), or more likely because of the unreliable human ground truths used as suggested in [7] and [13]. Upon subjective analysis, it can be seen that the visible differences between, for example, black knots and dead knots

is often very questionable, with the two classes often confused as a result. A better defined set of defect classes may be appropriate.

As a test of the proposed methods in a better defined pattern recognition environment, we turn to the task of optical character recognition.

5.2.1 The Dataset

The dataset employed for this purpose is a subset of the handwritten digits database offered by the CEDAR center [35]. It consists of 2100 pre-segmented samples of each of the decimal digits 0 through 9 in binary bitmap format. A sample selection can be found in Appendix B: Handwritten Digits.

The features used include a standard assortment of basic measurements, including horizontal, vertical and grid histograms, taken in 16 bins each; 16 relative projection features; 6 projection moments, as well as a group of 6 additional unrelated measurements including perimeter, density, form factor, rectangularity, as well as the relative position of the character's centroid.

All told, the feature set consists of 76 features distributed in 6 logical feature subgroups.

Feature Subgroup	Total Features
Horizontal Histogram	16
Vertical Histogram	16
Grid Histogram	16
Relative Projection	16
Projection Moments	6
Various	6

Table 5.7: Numeral features distributed into 6 logical subgroups.

Again, all features were normalized to lie in the range [0,1].

5.2.2 Feature Subsets

The benchmarks and proposed methods were optimized in the manner already specified, with one notable exception in that there were 6 feature subgroups instead of 5. Also, because the number of classes grew from 6 defect types to 10 digits, it was necessary to constrict the maximum FCM partition exponent m_f to 6 from the 10 suggested by C_T (the total number of classes) in order to limit search times, because a complete combinatorial search space for the later multi cluster-space and hybrid-space classifiers would be unwieldy at 1 million possible permutations.

Another departure from the method used in the case of the defect set was in the training/testing procedures. All optimization and parameter searches were performed on a 25% subset of the datasets because using the entire 2100 instances would require too much time for the combinatorial searches.

All testing was performed on the entire dataset however, and the figures shown all display the mean accuracy of classifiers operating on identical randomly selected train/test set pairs.

	Cluster Number				
Feature Group	2	3	4	5	6
Horizontal Histogram	8	4	6	5	7
Vertical Histogram	2	8	8	7	6
Relative Projection	6	8	4	6	5
Projection Moments	9	7	9	3	6
Assorted Features	7	4	9	8	8
Grid Histogram	7	5	5	7	9

Table 5.8: Best m_f values for each combination of c_f and feature subgroup.

	Cluster Number				
Feature Group	2.00E+00	3.00E+00	4.00E+00	5.00E+00	6.00E+00
Horizontal Histogram	1.00E-06	1.00E-03	1.00E-03	1.00E-03	1.00E-03
Vertical Histogram	1.00E-15	1.00E-03	1.00E-03	1.00E-03	1.00E-03
Relative Projection	1.00E-09	1.00E-09	1.00E-03	1.00E-09	1.00E-03
Projection Moments	1.00E-12	1.00E-03	1.00E-06	1.00E-03	1.00E-03
Assorted Features	1.00E-03	1.00E-03	1.00E-03	1.00E-03	1.00E-03
Grid Histogram	1.00E-03	1.00E-09	1.00E-03	1.00E-03	1.00E-06

Table 5.9: Best ε_{T1} values for each combination of c_f and feature subgroup.

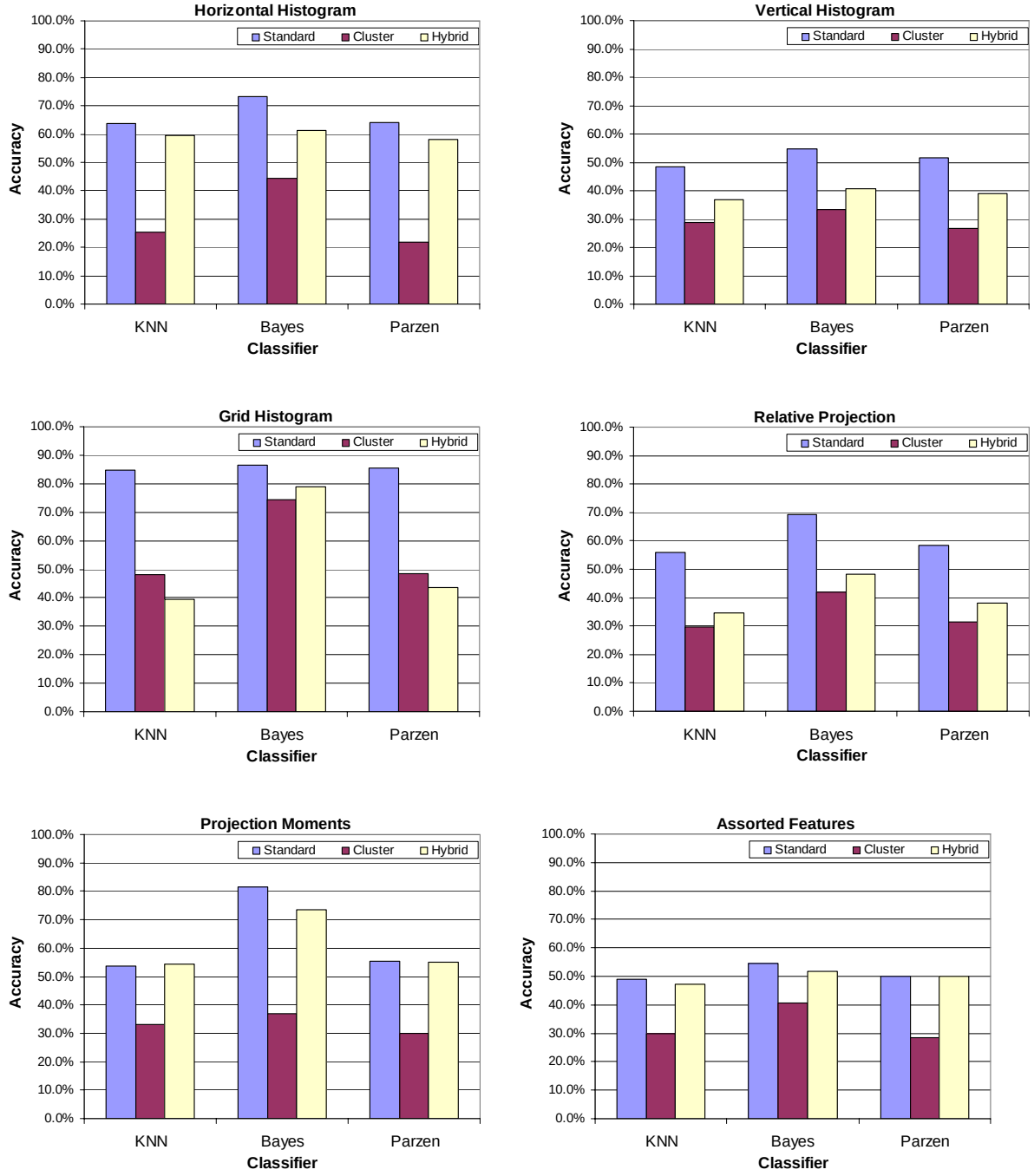


Figure 5.7: Classifier performances over numeral feature subsets.

Figure 5.7 illustrates once again that it is in difficult feature-spaces (those with poor classifier performances), that the proposed methods thrive. In this case, only in the projection moments and assorted feature-spaces does a single FCM hybrid method meet the standard classifier's performances.

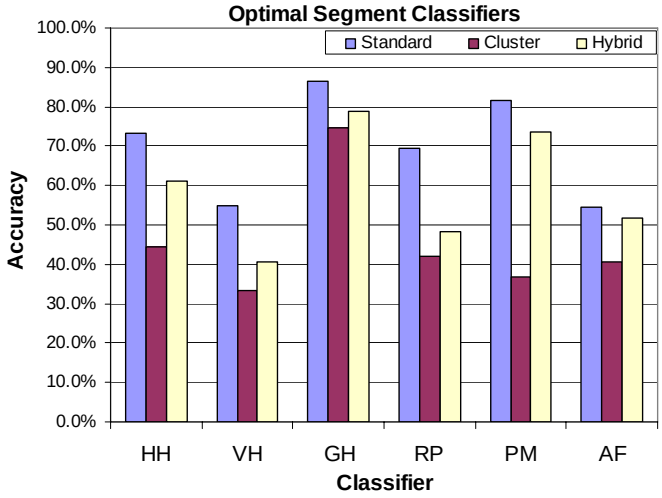


Figure 5.8: Summary of best performances over numeral feature sub-groups.

When comparing the best classifier performances over sub-groups in Figure 5.8, it becomes clear that the single cluster-space and hybrid-space classifiers simply lose too much information, especially in a better defined feature space, such as these numeral features afford (as opposed to the lumber defect data).

5.2.3 Entire Numeral Set

The quality of the numeral features used is really illustrated when the feature subgroups are joined. As illustrated in Figure 5.9, the Bayesian benchmarks approached 95% accuracy on the entire feature

set. The mean-rule aggregated multi-classifiers again dominated performance statistics, except in the case of the KNN classifiers, in which the standard benchmark was the best performer.

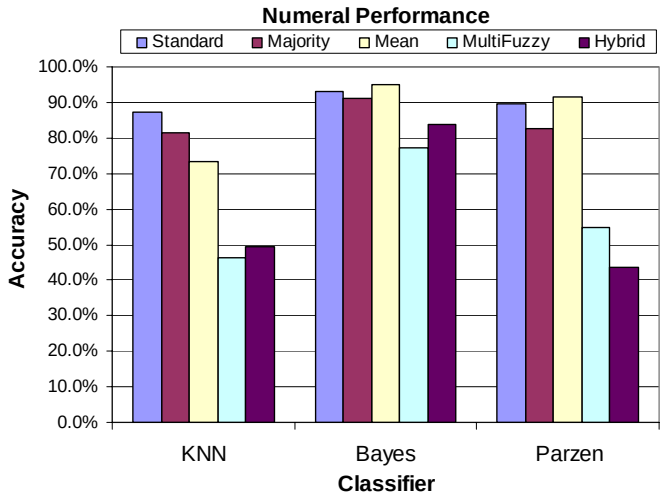


Figure 5.9: Classifier performance over entire numeral set.

Both of the proposed methods underperformed the benchmarks, indicating that in cases of well-defined feature spaces in which standard classifiers are adequate to the task, too much information loss can be expected in the conversion to cluster or hybrid-space.

Feature Type	Cluster Number	Direct Features
Horizontal Histogram	0	0
Vertical Histogram	0	0
Relative Projection	4	0
Projection Moments	0	Projection Moments 2
Assorted Features	0	0
Grid Histogram	8	Grid Histogram 3 & 14
Total Features	10	13

Table 5.10: Number of features fed to defuzzifier.

The fact that most feature subgroups pass no information to the defuzzifier (Table 5.10) is puzzling. There is no clear explanation for the best near-optimal solution ignoring so much potential information; this is a point at which further study could be directed in the future.

5.2.4 Feature Selection Incorporated

To test performances over optimized feature-spaces, the feature sets were altered in the way already described for the lumber dataset: through Pudil's forward floating search to yield feature subsets, by exposing the selection algorithm to both the entire feature set, and to the individual feature subsets. The features selected are listed in Appendix D.

The results of the various classifiers operating on these feature sets is shown in Figure 5.10.

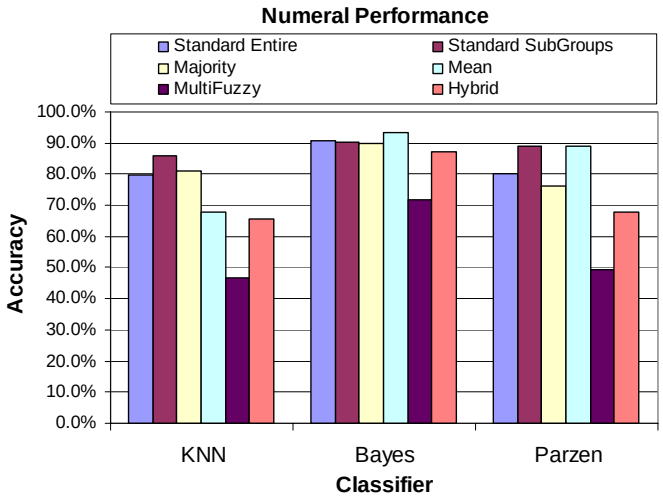


Figure 5.10: Performances of classifiers on numeral feature sets selected by Pudil's floating forward selection.

The proposed methods are again outperformed by the benchmarks, indicating that well-defined feature spaces do not appear to be suited to the cluster or hybrid-space transforms. It is interesting to note that the classifiers operating on the supposedly superior feature subsets selected by Pudil's algorithm in some cases actually underperformed those which operated on the entire feature-set, before selection. This is probably the result of the non-optimistic nature of the forward floating selection, and also the non-monotonous nature of the ranking criteria used (Bayesian classification error on an independent testing set).

Feature Type	Cluster Number	Direct Features
Horizontal Histogram	0	
Vertical Histogram	0	
Relative Projection	4	
Projection Moments	3	Projection Moments 2, 4, & 6
Assorted Features	0	
Grid Histogram	6	Grid Histogram14
Total Features	10	14

Table 5.11: Number of features fed to defuzzifier.

Once again, the best multi cluster-space classifier found appears to be ignoring the information from many feature subgroups, confirming the peculiar results demonstrated in table 5.10. A possible explanation is that the FCM base-classifier parameters found in the single cluster-space searches simply yield cluster-spaces which, although quite descriptive in the single cluster-space classifier case, are for the most part not well suited for combining with each other. This is not confirmable at this time; more experimentation is required.

Another important result to note is that, once again, a mean-rule multi-classifier aggregate was the top performer, and that the standard classifiers operating on feature subsets formed by joining the selected subsets matched or outperformed the standard classifiers which performed on the feature sets which were formed by the selection algorithm on the entire feature set.

This indicates again, that optimizing the feature subsets independently has resulted in a better feature set than the one formed by selection on the feature set in its entirety. This may indicate that it is easier to find an optimal or near-optimal solution by searching in smaller independent spaces, and then joining the results, rather than searching in an unmanageably large space.

5.3 Simultaneous Feature and Classifier Optimization

The most flexible version of the generalized framework proposed in Chapter 4 suggests the simultaneous optimization of the feature set and the classifier operating on it. Unfortunately, even with the added advantage of optimizing feature subgroups independently of each other, the search space is still unmanageably large for any type of exhaustive combinatorial search. As well, because the search space is so highly nonlinear and far from well-behaved, no reliable gradient or hill-climbing heuristic can be used to help guide the search.

In such instances it has often been shown beneficial to introduce a form of stochastic search such as evolutionary algorithms [6]. To this end, it was attempted to encode all the necessary parameters in the form of a bitstring, and allow the evolutionary operators of mutation, crossover, reproduction and selection to guide the evolution of a "population" of solutions.

Because the Bayesian classifiers typically have shown the best performance thus far, the genetic optimization has been limited to these.

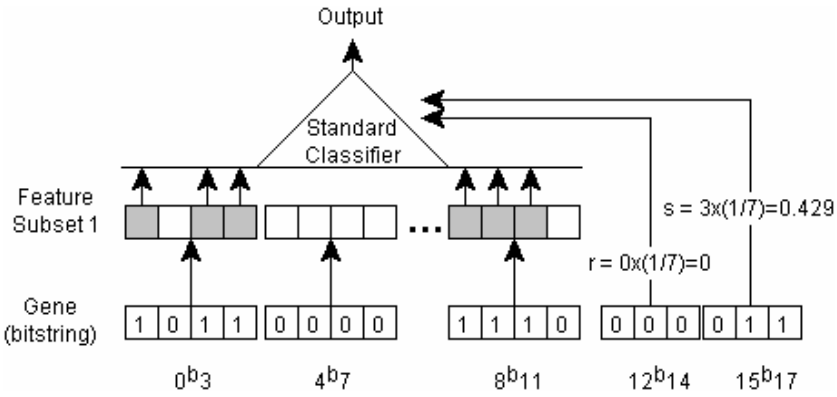


Figure 5.11: Gene encoding for benchmark.

Gene encoding of the benchmark Bayesian classifier was left relatively simple, and is illustrated in Figure 5.11. One bit represents each feature, a 1 indicating that feature is to be used, a zero indicating

that it is not to be passed on to the classifier. The r and s regularization parameters used to estimate covariance (as per equation 5.1) are encoded in 3 bits each, allowing for values between 0 and 1 with a step size of 1/7 (approximately 0.143).

$$\{0 \le s, r \le 1 \mid s, r = \frac{i}{7} \mid i = 0,1,...,7\} \tag{5.4}$$

The encoding of the hybrid-space classifier was necessarily more complex. Two bits are required for each feature, one to indicate whether to pass it on to its corresponding FCM base classifier, and one to indicate whether it will be one of the direct features bypassing the cluster-space transformation.

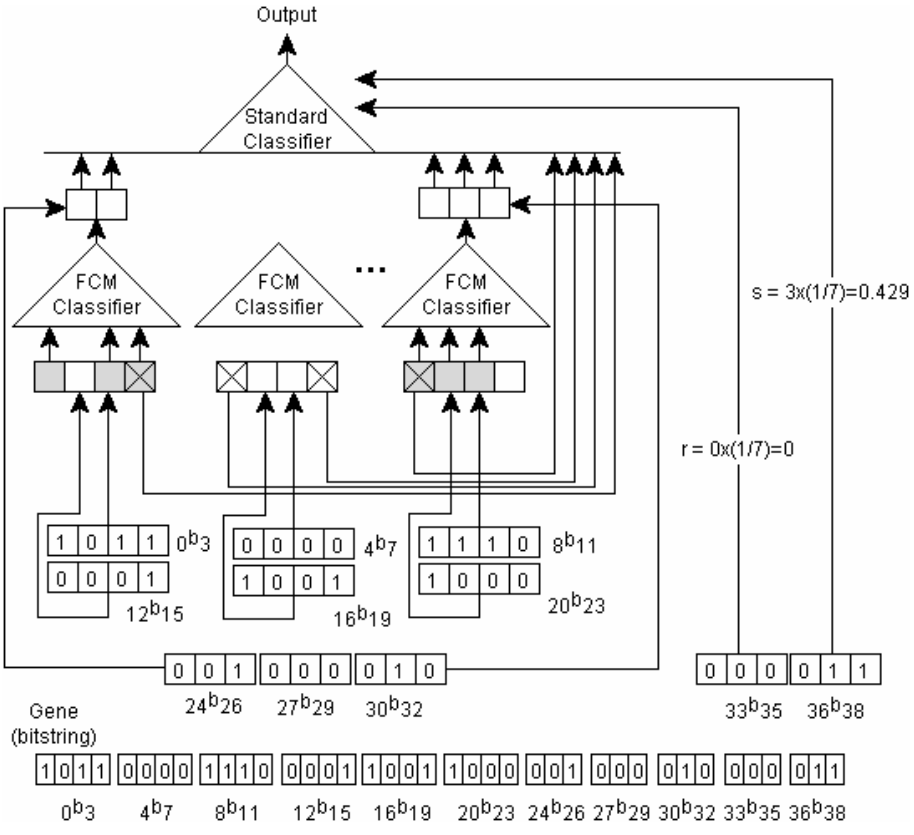


Figure 5.12: Gene encoding of proposed multi hybrid-space classifier.

One of the immediate consequences of this scheme is that a feature can be omitted from the cluster-space transformation, but still passed directly to the defuzzifier, a state of affairs not permitted in the previous hybrid-space classifiers, which limited the direct features to those which had already been selected as those to be transformed as well.

Three bits are required for encoding how many clusters to group each feature subspace into through the FCM classifiers. 8 unique combinations allow for the encoding of 0, 2, 3,…, 8 possible clusters for each sub-classifier. Finally, as was the case with the benchmark encoding scheme, 3 bits are used for each of the r and s parameter of the final defuzzifier.

The fitness of each individual gene in the population was evaluated as it was previously during the combinatorial search, by the accuracy of the resulting classifier over the same three random training/testing set pairs.

5.3.1 Lumber Defects

The genetic algorithm was allowed to run for the benchmark classifiers until such a time as there was no improvement for 50 generations.

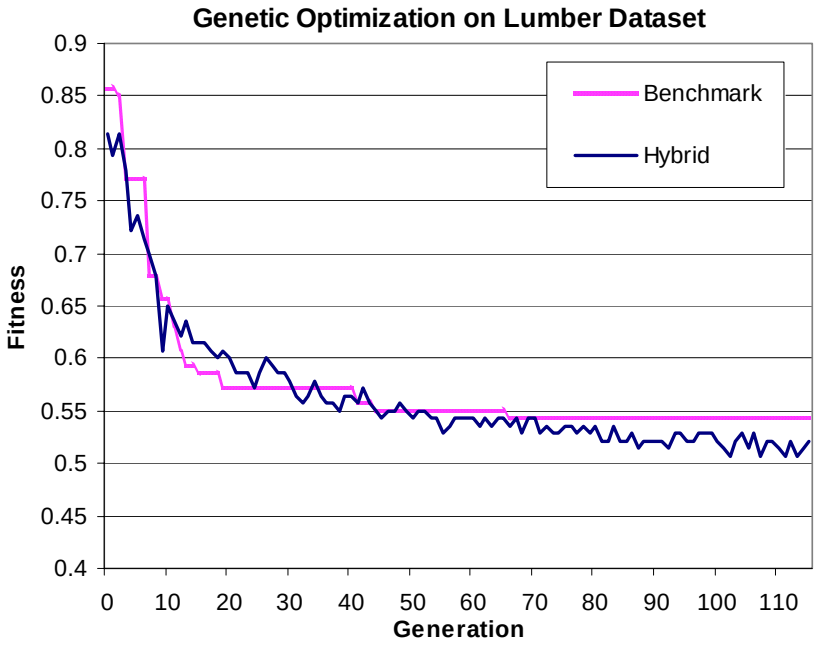


Figure 5.13: Best fitness values of each generation, evaluated as the sum of recognition error rates on three randomly chosen (but consistent across generations) lumber train/test set combinations.

The hybrid classifier population's fitness function was not as well behaved; its best fitness seemed to vary a great deal from generation to generation, as can be seen in Figure 5.13. This was most likely due to the underlying random effects of the cluster-space transform, in that an identical classifier may cause slightly differing cluster-spaces given identical feature-spaces, depending on such factors as the starting point of the fuzzy cluster centers.

	Benchmark	Hybrid-Space
Accuracy (%)	78.3	79.2
Standard Deviation (%)	3.6	3.2

Table 5.12: Genetic algorithm optimized lumber defect classifier results.

Table 5.12 indicates that the hybrid classifier outperforms the standard benchmark, but only by a slim margin. Also, it should be said that the classifiers found in this way (by optimizing feature sets

as well as classifier parameters simultaneously) outperformed any of the previous classifiers, definitely lending support to the intuitive notion that feature-spaces and classifier parameters should be found in tandem.

Although a simultaneous search dramatically increases the search space, and probably rules out any chance of an optimal solution, a near-optimal solution appears to be within reach of a stochastic algorithm.

5.3.2 Numeral Recognition

Numeral benchmark optimization was also allowed to run until fitness showed no improvements over 50 generations.

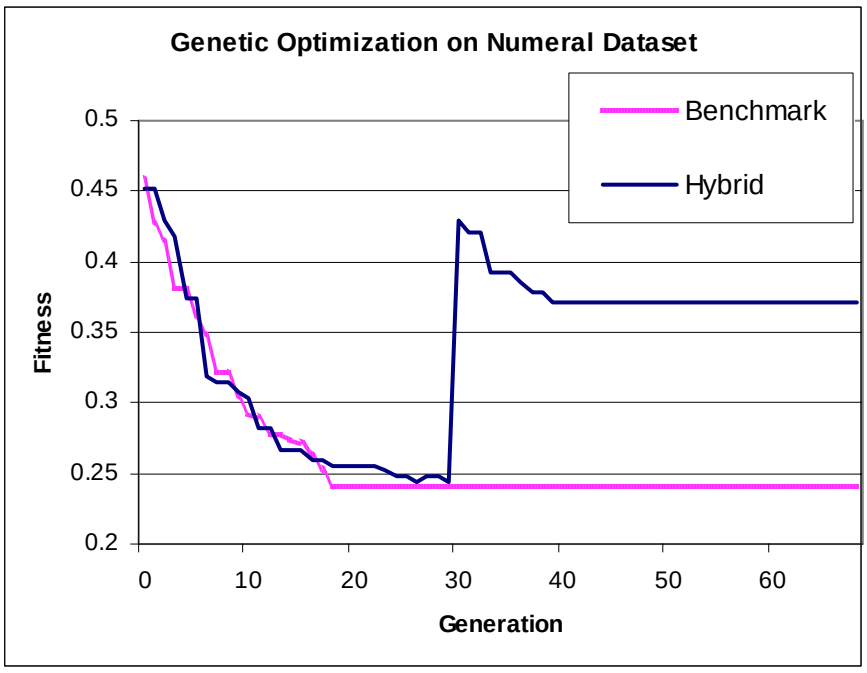


Figure 5.14: Best fitness values of each generation, evaluated as the sum of recognition error rates on three randomly chosen (but consistent across generations) numeral train/test set combinations.

The large spike in hybrid fitness shown at around generation 30 was due to a change in the way fitness was evaluated to decrease search times at that point. Here the fitness function was modified so that the classifiers formed operated over a lesser subset of the numeral dataset, thus accounting for the large decrease in fitness that resulted as a loss of information in each of the population's individuals (the hybrid classifiers).

	Benchmark	Hybrid-Space
Accuracy (%)	93.2%	90.7%
Standard Deviation (%)	0.7%	1.0%

Table 5.13: Genetic algorithm optimized numeral classifier results.

The results listed in table 5.4 indicate that the benchmark classifier outperformed its multi hybrid-space counterpart, but this is definitely the result of a non-optimal search result. Because the cluster numbers of the base FCM classifiers are allowed to take on values of 0 (indicating that classifiers exclusion from the defuzzifiers hybrid-space), it is easy to see that all possible benchmark classifier configurations are simply subsets of the total possible hybrid-space classifiers (those with cluster-numbers of 0, and therefore hybrid-spaces composed entirely of features which bypass the cluster-space transform).

It is certain therefore, that given enough run-time, and varying starting conditions, a hybrid-space classifier should be found with at least equal (if not superior) performance to that of the best benchmark.

Chapter 6
Conclusions and Future Work

This thesis proposed a new framework for the design of defect recognition systems for automated visual inspection. The main goals were a system that could easily be adapted to different applications, with minimal design decisions by an engineer or operator. In addition, two novel methods were proposed which fit well within this framework. There are several conclusions that can be drawn from this work.

6.1 Conclusions

- The cluster-space and hybrid-space classifiers work best in difficult feature spaces, which haven't been modified by any transforms or selection techniques. Examples of this result are all of the single hybrid-space classifiers in the texture subgroup of the lumber dataset, whose performance is far above and beyond those of the standard classifiers. Also, the single hybrid-space classifiers with Bayesian defuzzifiers slightly outperformed their standard Bayesian counterparts in the Histogram and Colour subgroups.

- In cases of better defined feature-spaces (such as that provided by the numeral dataset), which enable excellent classification rates by simple benchmark classifiers, the proposed methods do not fare as well. This is likely the result of too much information loss during the conversion to cluster-space.

- There appears to be validity in the notion that logical feature subsets can be optimized separately, as suggested in [3] and [33]. This is evidenced by the fact that the benchmark multi-classifiers tended to outperform the standard benchmarks which were exposed to the entire feature-set, whether optimized or not. Especially significant is the fact that the regular benchmarks (single classifiers) which operated on the union of separately optimized feature subsets in many cases outperformed those operating on the feature set optimized in its entirety.

- Finally, the concept of searching both the feature-space and classifier parameter-space simultaneously looks very promising. Despite the fact that the resulting search-space is much greater than that of an independent search of feature and parameter spaces, it appears that a stochastic search such as that made possible through genetic algorithms is more likely to produce a near optimal solution in this larger space, than are multiple non-optimal searches in separate search spaces.

6.2 Future Work

- It would be useful to explore the possibility of a better incremental combinatorial search when optimizing the proposed classifiers. Because each search step is performed using the results of an arbitrary Bayesian classifier (one with both r and s parameters set to 0), the final results in which different classifiers are used to defuzzify the cluster or hybrid spaces are

likely skewed towards this particular Bayesian classifier. A better approach might be to use KNN and Parzen based classifiers for the intermediate steps of their own search; better performances for those classifiers might then occur.

- It has been noted that as the exponent used to raise the fuzzy partition increases, the resulting cluster-space grows fuzzier; that is, its membership values tend to be shared more evenly among clusters. This results in mean membership values which approach values of 1/n, where n is the number of clusters. A direct result of this is that some parts of the cluster-space (the extremes) remain unutilized. It might be interesting to attempt a scaling of the cluster-space to limit it to known extrema (as was performed in the actual feature-spaces to limit values to the range [0,1]), so that some cluster features don't dominate others in magnitude.

- Some of the main difficulties of this project were the long search times associated with traversing the search spaces. In particular, the optimization of the multi and hybrid classifiers, which required the construction and evaluation of 3 classifiers proved quite time consuming. Optimizing the code for better execution times could easily be accomplished, and using a language closer to native resources than MATLAB might also be beneficial. Because most of the optimization searches were of the "embarrassingly parallel" variety (combinatorial and evolutionary search), a parallel implementation would be very easy to realize on specialized hardware, and would go a long way towards cutting search times.

- An implicit assumption made in the search for the optimal multi cluster and hybrid-space classifiers was that the most effective single cluster-space classifiers for each cluster number when joined in tandem would produce the best multi-classifier. Therefore, no search was

performed for optimal m_f and ε_{T1} parameters; they were assumed as given by the previous single classifier searches. This was done to significantly narrow the search space, but it could prove a false assumption, in that ε_{T1} and particularly m_f can have a great impact on the distribution of the resulting cluster-space, and although such a cluster-space might be ideal for resulting classifiers operating on only a single subset, they might be far from ideal to work in unison with the other feature subgroup cluster-spaces. Expanding the combinatorial search to include ε_{T1} and m_f in the multi-classifier case is simply not feasible due to the resulting exponential increases in search-space, but it may be worth attempting in the case of the evolutionary search.

- This work didn't explore in any detail the characteristics of a particular feature set that might indicate whether or not it might benefit from a partial or full transformation into cluster-space. The only definitive information appeared to suggest that a complicated feature-space with much redundancy probably stands to gain more than a well ordered or highly orthogonal or independent one. In the case of the latter, valuable information appears to be lost in the transition. An interesting extension of this work therefore, would be in exploring this particular detail: what characteristics make feature spaces right for transformation into cluster-space, which are better supplemented with cluster information into a hybrid-space, and which are better left fed purely to a classifier unaltered?

Appendix A

Dataset 1: Segmented Lumber Defects

Figure A.1: Sample board images from lumber dataset [34].

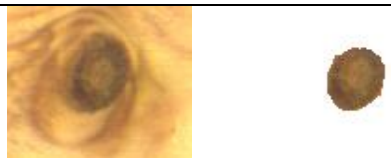
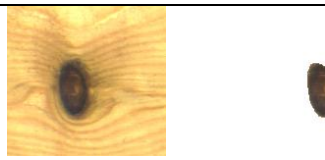
Sound knots		
Dead knots		
Black knots		
Pitch Pockets		
Straight Shake		
Core Strip (Heart-Wood)		

Figure A.2: Sample of defects segmented to establish shape data.

Appendix B

Dataset 2: Handwritten Digits

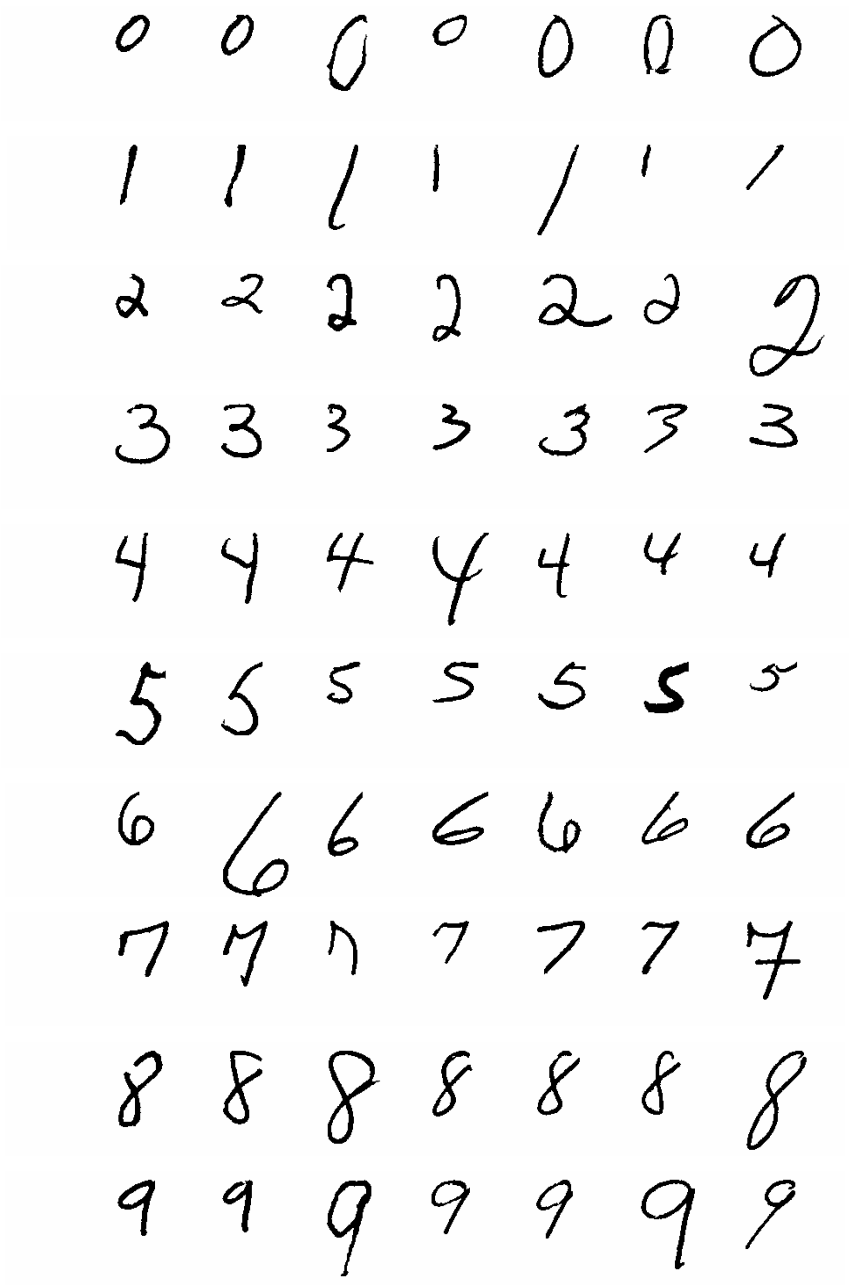


Figure B.1: Sample of segmented digits from CEDAR database [35].

Appendix C

Feature Selection: Lumber Results

This appendix contains detailed information concerning which features were selected in the various experiments dealing with the lumber dataset. The entire column contains those features selected by forward floating selection when exposed to the entire feature set, while the Subgroup contains those features selected by floating feature selection dealing with just the features contained within that subgroup. GA FCM and GA Direct lists those features passed on to the FCM base-classifiers for conversion to cluster-space and those which bypassed the conversion to be dealt with directly by the defuzzifier, respectively. Totals are tallied to give an indication of the descriptiveness of each feature.

Shape Feature	Entire	Subgroup	GA FCM	GA Direct	Total
Convexity				x	1
Primary Axis Ratio	x	x			2
Compactness	x	x		x	3
Circular Variance			x	x	2
Elliptical Variance		x	x		2
Circularity	x	x	x		3
Absolute Circular Difference	x	x			2
Absolute Elliptical Difference			x		1
Relative Circular Difference		x	x		2
Relative Elliptical Difference		x			1
Outer Circular Difference		x	x	x	3
Outer Elliptical Difference		x		x	2
Inner Circular Difference			x	x	2
Inner Elliptical Difference	x	x	x		3
Total	5	10	8	6	

Table C.1: Shape features selected.

Texture Feature	Entire	Subgroup	GA FCM	GA Direct	Total
Entropy10			x		1
Energy10			x	x	2
Contrast10				x	1
Mean10			x	x	2
Entropy20			x		1
Energy20					0
Contrast20					0
Mean20		x			1
Entropy01		x			1
Energy01	x				1
Contrast01		x	x	x	3
Mean01	x			x	2
Entropy02			x		1
Energy02	x				1
Contrast02					0
Mean02			x		1
Entropy-11		x		x	2
Energy-11			x		1
Contrast-11			x		1
Mean-11					0
Entropy-22			x	x	2
Energy-22			x		1
Contrast-22			x		1
Mean-22			x	x	2
Entropy11		x		x	2
Energy11					0
Contrast11					0
Mean11					0
Entropy22					0
Energy22					0
Contrast22				x	1
Mean22				x	1
Total	3	5	13	11	

Table C.2: Texture features selected.

Histogram Feature	Entire	Subgroup	GA FCM	GA Direct	Total
Bin0			x		1
Bin1	x		x	x	3
Bin2		x		x	2
Bin3		x	x	x	3
Bin4			x		1
Bin5		x			1
Bin6				x	1
Bin7				x	1
Histogram Mean		x			1
Total	1	4	4	5	

Table C.3: Histogram feature selected.

Position Feature	Entire	Subgroup	GA FCM	GA Direct	Total
Top		x			1
Bottom	x	x		x	3
Left		x		x	2
Right		x		x	2
CenterX		x		x	2
CenterY		x		x	2
Area		x	x		2
Inclination		x	x	x	3
Total	1	8	2	6	

Table C.4: Position features selected.

Colour Feature	Entire	Subgroup	GA FCM	GA Direct	Total
Percentile Red95		x		x	2
Percentile Green95				x	1
Percentile Blue95		x	x	x	3
Percentile Red90	x				1
Percentile Green90	x		x		2
Percentile Blue90			x	x	2
Percentile Red80	x		x		2
Percentile Green80			x		1
Percentile Blue80	x		x		2
Percentile Red70	x	x	x		3
Percentile Green70			x		1
Percentile Blue70	x	x		x	3
Total	6	4	8	5	

Table C.5: Colour features selected.

Appendix D

Feature Selection: Numeral Results

Feature selection data is listed here in a fashion identical to that used in Appendix C.

Horizontal Histogram	Entire	Subgroup	GA FCM	GA Direct	Total
Horizontal Histogram1					0
Horizontal Histogram2			x		1
Horizontal Histogram3					0
Horizontal Histogram4		x			1
Horizontal Histogram5					0
Horizontal Histogram6				x	1
Horizontal Histogram7		x			1
Horizontal Histogram8		x	x		2
Horizontal Histogram9		x			1
Horizontal Histogram10		x			1
Horizontal Histogram11		x			1
Horizontal Histogram12		x		x	2
Horizontal Histogram13		x	x	x	3
Horizontal Histogram14		x	x	x	3
Horizontal Histogram15		x			1
Horizontal Histogram16		x		x	2
Total	0	11	4	5	

Table D.1: Horizontal histogram features selected.

Vertical Histogram	Entire	Subgroup	GA FCM	GA Direct	Total
Vertical Histogram1		x		x	2
Vertical Histogram2		x		x	2
Vertical Histogram3		x	x	x	3
Vertical Histogram4		x			1
Vertical Histogram5		x			1
Vertical Histogram6		x	x		2
Vertical Histogram7			x	x	2
Vertical Histogram8			x		1
Vertical Histogram9			x	x	2
Vertical Histogram10		x			1
Vertical Histogram11		x			1
Vertical Histogram12		x	x		2
Vertical Histogram13	x	x	x		3
Vertical Histogram14		x		x	2
Vertical Histogram15		x	x	x	3
Vertical Histogram16		x		x	2
Total	1	13	8	8	

Table D.2: Vertical histogram features selected.

Grid Histogram	Entire	Subgroup	GA FCM	GA Direct	Total
Grid Histogram1		x	x	x	3
Grid Histogram2		x		x	2
Grid Histogram3			x	x	2
Grid Histogram4			x	x	2
Grid Histogram5	x	x	x	x	4
Grid Histogram6		x		x	2
Grid Histogram7	x	x		x	3
Grid Histogram8	x		x	x	3
Grid Histogram9	x	x		x	3
Grid Histogram10	x	x		x	3
Grid Histogram11		x	x		2
Grid Histogram12				x	1
Grid Histogram13	x			x	2
Grid Histogram14	x	x		x	3
Grid Histogram15	x	x		x	3
Grid Histogram16			x		1
Total	8	10	7	14	

Table D.3: Grid histogram features selected.

Relative Projection	Entire	Subgroup	GA FCM	GA Direct	Total
Relative Projection1				x	1
Relative Projection2		x		x	2
Relative Projection3		x			1
Relative Projection4		x		x	2
Relative Projection5		x		x	2
Relative Projection6			x	x	2
Relative Projection7		x	x		2
Relative Projection8			x		1
Relative Projection9		x		x	2
Relative Projection10		x		x	2
Relative Projection11		x		x	2
Relative Projection12		x	x	x	3
Relative Projection13		x			1
Relative Projection14			x		1
Relative Projection15		x	x	x	3
Relative Projection16			x		1
Total	0	11	7	10	

Table D.4: Relative projection features selected.

Projection Moments	Entire	Subgroup	GA FCM	GA Direct	Total
Projection Moments1		x		x	2
Projection Moments2	x	x	x	x	4
Projection Moments3		x	x		2
Projection Moments4	x	x	x		3
Projection Moments5		x			1
Projection Moments6	x	x		x	3
Total	3	6	3	3	

Table D.5: Projection moments features selected.

Assorted Features	Entire	Subgroup	GA FCM	GA Direct	Total
Density		x	x		2
Perimeter				x	1
CentroidX		x		x	2
CentroidY		x	x		2
Form Factor	x	x	x	x	4
Rectangularity		x	x	x	3
Total	1	5	4	4	

Table D.6: Assorted features selected.

References

[1] T. S. Newman and A. K. Jain. A survey of automated visual inspection. *Computer Vision and Image Understanding*. 61(2):231-262, 1995.

[2] E. Bayro-Corrochano. Review of automated visual inspection 1983 to 1993 Part I&II. *Intelligent Robots and Computer Vision XII.* SPIE 2055:128-172, 1993.

[3] T. Maenpaa, M. Pietikainen, and J. Viertola. Separating color and pattern information for color texture discrimination. *Pattern Recognition*, *Proceedings of 16th International Conference on*. 1(11-15): 668 – 671, 2002.

[4] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

[5] E. Ruspini. A new approach to clustering. *Inf. Control*, 15:22-32, 1969.

[6] M. Negnevitsky. *Artificial Intelligence, A Guide to Intelligent Systems*. Addison Wesley, Harlow, England, 2002.

[7] J. Iivarinen and A. Visa. An adaptive texture and shape based defect classification. *Pattern Recognition*, *Fourteenth International Conference on*. 1(16-20):117 -122, 1998.

[9] T. Mäenpää, J. Viertola, and M. Pietikäinen. Optimising colour and texture features for real-time visual inspection. *Pattern Analysis & Applications.* 6(3): 169 - 175, 2003.

[10] T. Kohonen. *Self-Organization and Associative Memory 3rd Ed*. Springer Series in Information Sciences, Springer-Verlag, 1989.

[11] J.C. Bezdek. *Fuzzy Mathematics in Pattern Classification.* Ph.D. Thesis, Applied Math. Center, Cornell University, Ithaca, 1973.

[12] B. G. Batchelor and P. F. Whelan. *Intelligent Vision Systems for Industry.* Springer Verlag, London, 1997.

[13] A. D. H. Thomas, M. G. Rodd, J. D. Holt, and C. J. Neill. Real-time industrial visual inspection: a review. *Real-time Imaging* 1:139-158, 1995.

[14] A. Kumar and G. K. H. Pang. Defect detection in textured materials using Gabor filters. *Industry Applications, IEEE Transactions on*. 38(2):425-440, 2002.

[15] J.C. Dunn. A fuzzy relative of the ISODATA process and its use in detecting compact, well separated clusters, *J. Cybernetics,* 3:95-104, 1973.

[16] S. Theodoridis and K. Koutroumbas. *Pattern Recognition 2nd Ed*. Elsevier Academic Press, San Diego, 2003.

[17] M. Perua and J. Iivarinen. Efficiency of simple shape descriptors. *Applications of Artificial Intelligence X: Machine Vision and Robotics*. Proc. SPIE 1708: 99-106, 1992.

[18] M. Koprnicky, M. Ahmed, and M. Kamel. Contour description through set operations on dynamic reference shapes. *International Conference on Image Analysis and Recognition*. LNCS 321:400-407, Springer-Verlag, Berlin 2004.

[19] B. Duin. *PRTools Version 3.0*. Pattern Recognition Group, Delft University of Technolgy, The Netherlands, 2000. http://www.prtools.org/

[20] B. G. Batchelor and P. F. Whelan. *Intelligent Vision Systems for Industry*. Springer Verlag, London, 1997.

[21] B. G. Batchelor and P. F. Waltz. *Intelligent Machine Vision*. Springer Verlag, London, 2001.

[22] US Ink. *Press Doctor 2002, A Web Offset Newspaper Guide*. US Ink, New Jersey, 2002.

[23]A. Kumar and G.K.H. Pang. Defect detection in textured materials using optimized filters. *Systems, Man, and Cybernetics, IEEE Transactions on, Part B: Cybernetics*, 32(5):553-570, 2002.

[24] K. R. Castleman. *Digital Image Processing*. Prentice Hall, New Jersey, 1996.

[25] R. M. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE* 67(5):786-804, 1979.

[26] J. P. Marques de Sá. *Pattern Recognition. Concepts, Methods and Applications*. Springer-Verlag, Berlin, 2001.

[27] A. Jain and D. Zongker. Feature selection: evaluation, application, and small sample performance. *Pattern Analysis and Machine Intelligence, IEEE Transacations on*. 19(2):153-158, 1997.

[28] D. P. Brzakovic, P. R. Bakic, N. S. Vujovic, H. Sari-Sarraf. A generalized development environment for inspection of web materials. *Robotics and Automation, IEEE International Conference on*, New Mexico, 1997.

[29] D. Bahler and L. Navarro. Methods for combining heterogeneous sets of classifiers. *Artificial Intelligence* (AAAI), *17th Natl. Conference on*, 2000.

[30] K. Chen, L. Wang, H. Chi. Methods of combining multiple classifiers with different features and their applications to text-independent speaker identification. *Pattern Recognition and Artificial Intelligence, International Journal of.* 11(3): 417-445, 1997.

[31] P. Pudil, J. Novovicova, J. Kittler. Floating search methods in feature selection. *Pattern Recognition Letters*, 15(11):1119-1125, 1994.

[32] R. Sablatnig. A flexible concept for automatic visual inspection. *Czech Pattern Recognition Workshop'97, Proc. of (CPRW'97)*, 87-96, 1997.

[33] O. Silven, M. Niskanen, and H. Kauppinen. Wood inspection with non-supervised clustering. *Machine Vision and Applications.* 13:275-285, 2003.

[34] VTT Building Technology and University of Oulu. *Lumber defect dataset and ground truths*. University of Oulu, Finland. http://www.ee.oulu.fi/~nisu/wood

http://www.ee.oulu.fi/~olli/Projects/Boards/Defect.File.Format.html

[35] Center of Excellence for document analysis and recognition. *CEDAR handwritten digit database*. CEDAR, State University of New York, Buffalo. http://www.cedar.buffalo.edu/Databases/CDROM1/