Learning to Learn:
An Adaptive Visual Object Recognition Approach for Handwritten Text Recognition

K.V.Haak (s1216260)
haak@ai.rug.nl

Supervisors:
Drs. T. van der Zant
Prof.Dr. L.R.B. Schomaker

Department of Artificial Intelligence
***University* of Groningen**

Summary

After some years of experience, humans read handwritten texts in a remarkably effortless and swift manner. While during development visual processing streams in the human brain adjust to perform the task of handwritten text recognition fluently, visual processing and recognition strategies become highly specialized. This thesis therefore approaches the problem of automated handwritten text recognition from a developmental perspective, wherein adaptive visual processing mechanisms specialize themselves to perform the task of recognizing writings.

Three well justified assumptions form the base of the approach. First, it is assumed that humans are provided with a basic mechanism of visual processing. Accounting for this assumption, the standard model of visual processing of Poggio and Edelman [70] is the cornerstone of the proposed approach and is used to represent images of handwritten texts. Motivated by the first 100-200 milliseconds of primate visual processing, the resulting representations are well suited for particular discrimination tasks. Secondly, it is assumed that the cortical processes underlying handwritten text recognition are merely a specialized form of general visual processing. This assumption is accounted for by specializing the previously mentioned representations of handwritten texts even more, while additionally specializing artificial neural network classifiers for particular discrimination tasks. Thirdly, based on the concept of *Neural Darwinism* of Edelman [23], it is assumed that specialization of this kind can be modelled with evolutionary algorithms.

Incrementally, a handwritten text recognizer is developed within this paradigm, whereas each step if experimentally assessed. First, it is demonstrated that the representations resulting from the use of the standard model of visual processing on handwritten texts are excellent for classification tasks. Secondly, using these representations, artificial neural networks with integrated feature selection abilities evolve into task-specific binary classifiers. After specialization, the task-specific representations are very sparse, while the artificial neural networks are very simple perfrom nearly flawless classification. Lastly, the task specific classifiers are combined to address multiple class recognition problems. With room for improvement, the resulting handwritten text recognizer needs very little characteristics to recognize handwritten texts accurate.

Contents

Chapter 1

Introduction

Visual object recognition is key for survival in many species and understanding this fundamental cognitive task is a primary goal in Neuroscience, Artificial Intelligence, and the field of Pattern Recognition. In addition to the capability of recognizing mere real objects, humans have the extraordinary ability to vividly imagine a world filled with objects, only by perceiving symbols in the form of for instance handwritten texts. For example, when a friend writes you a letter describing his room, there is no need for the visual input your presence in the room would invoke. Instead, it is sufficient to interpret the symbols written on the paper in order to perfectly imagine what your friend's room looks like. The interpretation of writings has become a key factor in human communication. Obviously, because symbolic representations make life a lot easier, considering you did not need to travel to your friends room to know what it looks like. From the moment that humans started to use writings for communication, tremendous advances in our common knowledge have been made for communication could now be stored on a piece of paper and read whenever needed.

Researchers have made many attempts to mimic aspects of human reading behavior such as the recognition of handwritten words. Unfortunately, mimicking this recognition ability is far from trivial, and until now automating handwritten text recognition remains an unsolved problem. This thesis gives an account on the difficulties underlying handwritten text recognition and proposes a new approach for future research. In short, it is assumed that handwritten text recognition is merely a particular highly specialized form of visual object recognition. The *general* principles of visual processing are the same whether it comes to either the recognition of a bird, car, face or a handwritten word. Moreover, when attempts are made to mimic handwritten text recognition with a computer program, one should take the development of such specific functionality into account. The development of a natural system often gives great insight into its functionality as an adult system. In addition, modeling the development of a system avoids difficulties that arise when bluntly setting up the parametrization of an adult system, since it is very hard to choose appropriate task-dependent parametrizations.

To elicit the difficulties of handwritten text recognition, one of Bongard's problems (BP 100) is used as an example throughout this thesis. Bongard problems are puzzles designed by the Russian scientist M. Bongard. In his book Pattern Recognition (1970) [10], he presents the reader hundred examples with

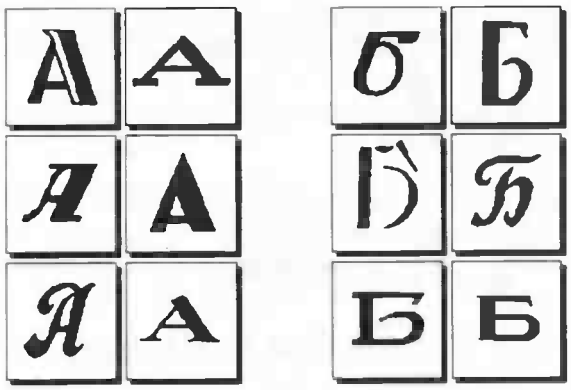


Figure 1.1: Bongard's Problem no. 100

twelve figures of which the six leftmost form class I, and the six rightmost form class II (Fig. 1.1). The reader is given the task to define rules that unambiguously predict which of the figures belongs to which class. Any recognizer, human or machine, uses what can be interpreted as rules to predict class-membership, and when recognizing handwritten words similar rules have to be defined by the word-recognizer in order to tell what word has been written down. In case of the example presented in figure 1.1, one could easily say that the figures on the left side belong to class I because they are all different appearances of the character 'A'. Similarly, since all figures on the right side are different appearances of the character 'b', they constitute class II. Unfortunately, these rules are not very useful since paradoxically, the rules need the figures to be recognized as either an 'A' or a 'b', which in turn needs rules to recognize the figures. Different, more abstract rules are needed to assign class-membership to any of the figures in figure 1.1. Finding the nature of the rules that constitute to recognition has been a key issue for the field of pattern recognition and is therefore a key topic in handwritten text recognition research.

The recognition of handwritten texts may well be one of the most challenging problems in the field of Pattern Recognition, as it requires many of the most difficult issues of the field have to be addressed. First of all, handwritten texts can appear in a myriad of shapes. This issue will henceforth be referred to as high within-class variability, which is explained below. Secondly, when recognizing handwritten words, either a holistic or reductionist approach is possible, and perhaps it is even necessary to employ both. Using a holistic approach, the recognizer attempts to recognize for instance a word as a whole. Contrary to the holistic approach, using a reductionist approach, the recognizer attempts to classify the word by focusing on its parts, say by its characters. Both approaches suffer severe difficulties. In case of the holistic approach, trying to recognize one out of tens of thousands of other words provides major difficulty for currently used statistical recognition techniques, a problem that will be addressed later. In case of the reductionist approach, recognizing characters prior to the word itself currently needs the characters to be segmented, a task that is often impossible without having recognized the word in the first place. Lastly, different handwritten words may have very similar appearances, which means different

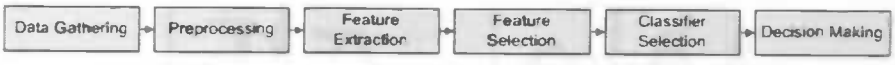


Figure 1.2: The design-cycle of Pattern Recognition according to [22].

classes are not well separable. This problem lies at the heart of pattern recognition problems, and solving it implies that the word needs to be represented in another way in order to define rules that assign class-membership.

1.1 Current Approaches in Pattern Recognition

Before continuing the elucidation of a new approach towards handwritten text recognition it is useful to provide some insight into currently used approaches in the field of Pattern Recognition. Roughly, the design of a pattern recognition system involves three aspects: data acquisition and preprocessing, data representation, and decision making [40]. Duda, Hart and Stork (1973) [22] define six steps in the design cycle. The first two steps consists of data gathering and preprocessing. Clean, usually labeled examples are needed for current classification strategies in order to be able to derive a function that maps an unknown pattern onto a class. In the next step the designer needs to extract features from the data that are presumed to be useful to constitute an input for the mapping function. That is, one needs to extract (a combination of) features, samples, or other primitives that represent one class, but not another. The fourth step selects from the necessary features those that are sufficient to represent the different classes. The main rationale behind this step is a reduction of the dimensionality of the input space, a topic that will be addressed later. In the fourth step the designer has to make choices concerning the method to be used for the derivation of the mapping function. In addition, in most of the cases parameters need to be set because the model needs to be tuned for the classification of a selected set of possible input patterns. In step five one derives the actual classification function. The last step consists of post-processing the output of the classification function (i.e., decision making).

1.1.1 Data Gathering

Data gathering is difficult and most of the time costly. When addressing a classification problem, one needs to decide what amount of data is needed to derive an appropriate mapping function. This is a decision that is concerned with the trade-off between the robustness of the resulting model and the amount of effort one is willing to put into the task. Current strategies often benefit from examples that have as little noise as possible, that are normalized, and usually labeled. Labeling is a task that can be even more time consuming and therefore expensive than the actual accumulation of the examples themselves. Classification of patterns that are highly variable (noisy) need more examples as opposed to patterns belonging to one class that all exhibit consistent similarities. Normalization is needed to prevent an increase of the dimensionality of the input space. When for instance an object belonging to one class is presented at arbitrary points along the horizontal axis of an image, a one-dimensional subspace is needed to represent the variable shift in location. Normalization is

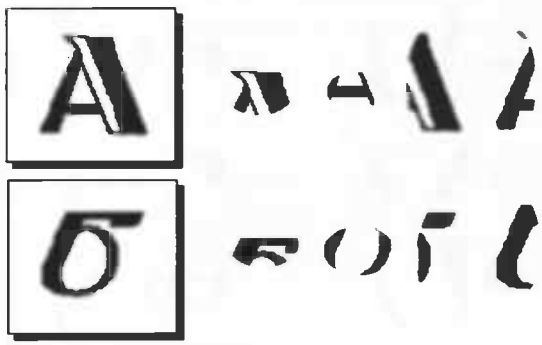


Figure 1.3: A possible method of feature extraction. On the left, two of the original figures from Bongard's Problem no.100 are shown of which a top, center, right, and left part of the picture are segmented. The result, shown left, is another representation of the two figures, facilitating the definition of classification rules.

thus needed to make the input patterns as invariant as possible, and thereby constraining the dimensionality of the input space. The same rationale applies to noise removal. Preprocessing the data appropriately therefore makes sure the obvious in-variances are removed, resulting in a simplified input space.

1.1.2 Feature Extraction

Feature extraction is the process of extracting characteristics from the gathered examples that are useful for the classification of an example. That is, an image of a particular word can only be classified as one, after extracting the characteristics of that word, for instance, the characters that constitute the word. Extracting features alters the representation of the example; a holistic representation is substituted by a description of the presence of local characteristics. Finding the characteristics of a class is far from trivial, they have to be class-specific and invariant, and characteristics can sometimes be a conjunction of features. For example, a face is only a face if it can be decomposed into a oval shape enfolding two eyes, and a nose, and a mouth. In addition, the class-specific facial features are expected to appear in fixed relative positions.

The process of feature extraction differs for every classification problem. That is, every class is described by different features and it is key to find them. Depending on the particular classification task, a wide range of possible feature extraction methods is available. Among the most commonly used are Principal Component Analysis (PCA) [44], Discriminant Analysis [59], and Self-Organizing Maps [51]. PCA extracts the m most expressive features by computing the k largest eigenvectors of the $d \times d$ covariance matrix of the n d-dimensional examples. PCA depends on the assumption the data is normally distributed, but comparable methods that do not need this assumption are also available (e.g., Projection Pursuit [38], and Independent Component Analysis [39])[40]. In contrast of PCA, Discriminant Analysis makes use of available class information by taking into account how separable the classes are when search-

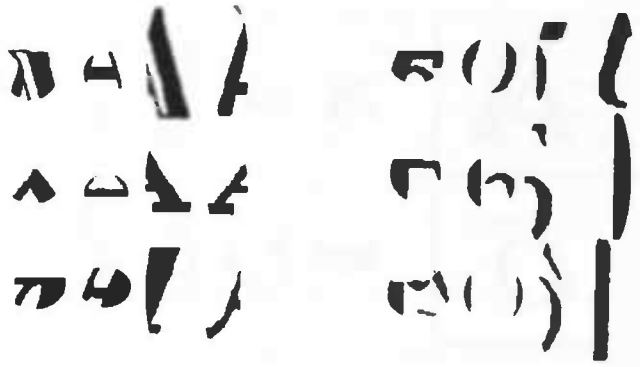


Figure 1.4: Extracted features from three of the figures of both classes in Bongard's Problem no. 100: The left figure shows extracted features from class I, the right from class II. Which of these features should be selected in order to predict class-membership as reliable as possible? Some of these features seem too similar and might be confusing, others are simply necessary.

ing for the most expressive features [59]. The Kohonen Self-Organizing Map (SOM) is a usually two-dimensional grid of artificial neurons. During training of the SOM, iteratively the neuron with a weight vector most similar to a sampled input vector is appointed and the weight vectors in the neighborhood of this particular neuron are updated towards the input vector. After training, a topology preserving map is formed that can be used to map an input pattern onto the activation of a single neuron in the SOM [51].

1.1.3 Feature Selection

Feature selection is the process of selecting a sufficient subset of features that, given a classification problem, elicit a facilitated discrimination between classes. The main rationale behind this step is the reduction of dimensionality. A pattern that is represented by a set of d features can be seen as a d-dimensional vector. The size of this vector (i.e., the dimensionality of the input space) determines to great extend the number of examples needed to later on derive an appropriate mapping function. More specifically, the quantity of data needed to derive a mapping function grows exponentially with the dimensionality of the input space. This phenomenon has been termed the curse of dimensionality [5]. When forced to work with a limited amount of data, which is usually the case in practice, increasing the dimensionality of the input space rapidly leads to the point where data is very sparse. This will then result in a very poor representation of the appropriate mapping function [8].

One way to avoid this problem is to choose a subset from the set of extracted features, thereby reducing the dimensionality of the input space. Once the necessary features are extracted in the feature extraction step, it is key to select a subset that contains only the necessary and sufficient features needed to discern different classes from each other. According to Occams razor [9], the minimum message length [74], and the minimum description length [88] the set of necessary and sufficient features can be defined by the minimal subset containing only

strongly relevant features, leaving out irrelevant and weakly relevant features that do not, or not always contribute to an appropriate mapping function [42].

Several methods of feature selection have been proposed, e.g., FOCUS [2] and Relief [48, 47, 52], and many others [41]. These methods either exhaustively examine all possible subsets of features to find the minimal subset of features that is sufficient to determine the class of a pattern, or sample examples randomly when assigning relevance values to features. Either of these methods has drawbacks, ranging from a high time-complexity to neglecting weakly relevant features that might have been useful regarding classification of some of the patterns.

1.1.4 Choice of Approach

When an appropriate representation of the data is found the next step in the design cycle is to choose an approach to derive a mapping function that predicts the class of an example. The authors of [40] state that the possible approaches are versatile but can be structured as follows. According to Bayesian decision theory a pattern belongs to a class according to a class-probability density function. When the class-probability densities are known it is possible to use the Bayesian Decision Rule [6] to predict the class of an unknown pattern. When the class-conditional probabilities are unknown they have to be learned from the gathered data and prediction of classes is done either by supervised or unsupervised learning methods, depending on the availability of class labels in the set of gathered data.

In case of supervised learning it is possible that, although the class-probability densities themselves are unknown (e.g., because of missing mean vectors or covariance matrices), the form (e.g., a multivariate Gaussian) of these densities is known. When this is the case a parametric approach is usually chosen. By estimating the missing parameters of the class-conditional probability density function (i.e., by learning what these values could be) it is again possible to predict the class labels by using the Bayesian Decision Rule. When also the form of the class-conditional probability densities is unknown, a non-parametric approach is used. Non-parametric approaches either estimate the density function itself, or construct a decision boundary based on the gathered labeled data. Examples of the latter approach are the k-nearest neighbor rule [16], artificial neural networks [75], and support vector machines [86]. In case of unsupervised learning the distinction between parametric and non-parametric approaches can again be made. Having information about the form of the class-conditional probability densities the approach of Mixture Resolving can be used (e.g., [18]). When this is not the case the designer is forced to enter the realm of cluster analysis, using an exploratory data analysis method. In addition to the choices that have to made concerning the use of the actual approach, many of the possible methods need parameters to be set.

1.1.5 Conclusion

Currently, designing an appropriate strategy for pattern recognition is laborious and difficult. When addressing a new problem domain, new data has to be gathered, new methods of feature extraction have to be designed, one has to find out which of those features are most relevant for the problem at hand,

and careful decisions have to be made concerning the classification method and its parametrization. This theses advocates that, inspired by nature, most of the steps in the design-cycle of pattern recognition can be automated, introducing the concept of self-adaptation towards domain-specific recognition problems such as handwritten text recognition. Below, the rationale behind this concept is elucidated, introducing three working assumptions regarding human handwritten text recognition.

1.2 Cognitive basis of Handwriting Recognition

Just over fifty centuries ago the cultural invention of writing took place [15], which revolutionized human communication. Even though, evolutionary accounts of functional specialization towards human reading abilities are predicated. On an evolutionary scale, fifty centuries years is simply insufficient time for selective pressures to 'engineer' a specialized module for visual word recognition. Bearing this in mind, either nature 'predicted' this cultural invention and provided humans with word recognition abilities in advance, or other mechanisms were at play, providing humans with the ability to learn to recognize words swiftly and robustly. Although coincidental 'prediction' of the invention of writing cannot be ruled out, it seems more likely that indeed the human brain is equipped with mechanisms that allow them to adapt to new tasks such as visual word recognition. Findings from research concerning neurogenesis and neuroplasticity support this presupposition, and allow for inspiration from nature developing a self-adapting word recognition system.

1.2.1 Specialized recognition: Development and Plasticity

Nature has provided most of the more complex living animals with a visual system, enabling them to visually recognize objects they encounter and act accordingly. Visual object recognition can be considered a fundamental cognitive task, and it is fair to presume that nature provided these creatures with basic visual processing capabilities by Darwinian evolution. Each of these animals, humans alike, are intrinsically provided with basic visual processing capabilities. There are two different visual processing streams of which the first is the so called dorsal stream that gives rise to automatic motoric responses without necessarily being aware of the environment. The ventral pathway relays visual information in order to perform the task of object recognition, following the necessary steps to discriminate one object from another [29]. What happens in the ventral stream is functionally equivalent to what researchers assessing the problem of object recognition are trying to mimic, using the design-cycle of pattern recognition.

Concerning the cognitive basis of object recognition, developmentalists have either argued that for instance face recognition is merely part of domain-general pattern recognition mechanisms [81] or that the details of the face are part of a genetically specified social module [12]. Nativists have used the existence of adults suffering prosopagnosia resulting from specific brain damage as a strong index of an innately specified face recognition module. However, it can also be argued that face recognition can be modularized with time as the child gradually learns about faces and as circuits in the brain progressively become specialized

for face recognition [43]. Evidence for specialization regarding word recognition is also available. McCandliss et al. (2003) [58] showed that the visual word form area, the fusiform gyrus, exhibits increasing expertise for reading.

Evidence from findings concerning cortical plasticity suggest that the brain indeed has the capability to alter the sensory processing functions of different brain areas depending on task demands. Apart from rigorous changes in function of brain areas (e.g., visual stimuli processed by multi-modal parts of the auditory cortex due to changing thalamus afferents [65]), it has also been shown that expertise for cars and birds recruits brain areas involved with face recognition [28] and intensive and long-lasting experience of altered sensory input induces permanent changes in the functional organization of the somatosensory cortex [11].

It is conceivable that the main processing functions involved in recognition are not genetically determined and hardwired in the neural circuits, but are the result of interactions between epigenetic influences (intrinsic factors that result in for instance architectural constraints) and the basic neural plasticity mechanisms. An only roughly predetermined organization, the plastic self-organizing behavior of the brain, and experience allow for the development of domain-specific object recognition mechanisms depending on task demands. Genetic instructions for assembling our neural structures are only approximate. Since life is unpredictable, humans have evolved the ability to redesign their brains within bounds, in response to their experience [80]. Behavioral, neurophysiological and neuroimaging data indicate that a single system is sufficient for the recognition of all objects at all levels [82]. In addition, evidence indicates that the task demands and learning that arise from different forms of feedback determine which computational routines are recruited automatically for recognition [82]. Taken together, object recognition is highly dependent on structural development and experience.

The brain knows many ways to restructure its processing units. Neuroscience elicited both progressive and regressive factors subserving the development of cortical areas. Known regressive developmental factors such as cell death, axonal retraction and synaptic pruning, or progressive factors like birth and proliferation of neurons, migration, and local dendritic branching, all contribute to a developmental mechanism that allows generic processing functionality to specialize with consideration of specific task demands.

1.2.2 Assumptions

The abovementioned findings make a case for the postulation of three assumptions regarding handwritten text recognition. First of all, it can be assumed that there is a general principle of object recognition. Basically, visual processing does not differ between tasks. Intrinsically defined processing mechanisms lay at the root of every recognition problem, whether face recognition or handwritten word recognition. These processing mechanisms enjoy the advantage of adaptivity to domain-specific object recognition. This assumption implies that within the design-cycle of pattern recognition, both preprocessing and feature extraction can be defined by a single mechanism: visual processing according to the ventral visual processing stream.

Secondly, handwritten text recognition is a particular (specialized) type of visual object recognition. The invention of writing introduced a new recognition

task, demanding visual processing of written words. Considering that adaptive mechanisms account for specialization of visual information processing, it is possible that also recognition of visually presented written words is a specialized function of the human recognition system, like the highly specialized ability to recognize faces or cars. It is assumed here that specialization towards swift and robust word recognition is the result of neuronal development bounded by epigenetic factors that roughly predetermined the organization of the processing mechanism. This assumption states that preprocessing and feature extraction in the domain of written word recognition can be modeled with similar techniques as in any other object recognition task, when chosen the appropriate (biologically plausible) method. This appropriate method should then be specialized for the task at hand.

Lastly, neuronal development, constituted by progressive and regressive mechanisms, is assumed to be a local evolutionary process, acting on the micro-level of neuronal competition. Neuronal structures are competing for activation, those that are suited for a task at hand will be given the opportunity to specialize even further, those that are not will be disregarded concerning the task. It is known that after the first months of life, large-scale elimination of axons and dendrites takes place, and it is also known that synaptic pruning is heavily influenced by experience, represented by activation all neuronal structures compete for [23]. Indeed, it has been argued that pruning is the direct result of activity-dependent competition, a process by which experience molds the brain into its final form, with specialized abilities like written word recognition (e.g., [13, 14, 23]). Gerald Edelman [23] referred to competition among axons as a general principle. In neural Darwinism, as he called it, neuronal development starts with more neurons and synapses than are kept. Synapses form haphazardly, and then a selection process keeps some and rejects others. Although the analogy with Darwinian evolution should be used cautiously, it remains a fact that the most successful neuronal structures thrive at the expense of the less successful, as in Darwinian evolution [46]. The implication of this assumption is that it is possible to mimic the process of neuronal development with evolutionary methods, molding generic visual object recognition into domain-specific object recognition by experience.

1.2.3 Conclusion

Inspired by the developmental mechanisms, three assumptions allow for a new approach towards solving the problem of handwritten text recognition. By modeling general visual processing of the ventral pathway in the brain, evolutionary methods make it possible to mimic the development towards domain-specific recognition tasks. This would result in a self-adapting recognition system for handwritten text recognition. However, recall that there are several major difficulties that arise when attempting to automate handwritten text recognition. Issues such as high within-class variability and low between-class variability still have to be addressed. Moreover, choosing a holistic approach, it is still necessary to deal with the problem of recognizing one word out of a myriad of others. Choosing a reductionist approach, it is still key to find a solution for the problem of recognizing parts that are difficult to segregate from the whole.

This thesis hypothesizes that self-adaptation solves these problems partially. Using a model of the first processing steps in the ventral visual pathway, adaptive

feature extraction ensures that the appropriate representation will be 'learned' which allows for relatively easy classification, thereby addressing the problems of high within-class variability and low between-class variability. Restricting this study to holistic recognition, the difficulties arising when dealing with a myriad of categories (i.e., words) will be addressed within the divide and conquer paradigm. That is, a difficult problem of recognizing one out of a vast amount of words is divided into many simple problems. Aware of the dangers in using this method - i.e., divide and conquer allows for a phenomenon in pattern recognition called 'early commitment', a topic that will be addressed more extensively later - multiple recognizers are employed in the decision-making step of each simple subproblem, avoiding early commitment as much as possible. Using different representations at every level of decision making, recognizing a written word exhibits a behavior that zooms in on a recognition problem, a possible solution for recognizing a myriad of categories that seems fairly intuitive.

1.3 Relevance to the field of Artificial Intelligence

In the field of Artificial Intelligence understanding autonomous perception is a primary research objective. The fundamental cognitive task of perceiving the world has therefore been a key research topic, resulting in disciplines such as Pattern Recognition, Object Recognition, and Handwritten Text Recognition, but also Sound Recognition. Perception is a primary necessity for artificial systems to act and react, possibly intelligently, according to their own representational view of the environment. In the closed world of writings, presenting an artificial system with only a limited set of static input, visual perception in its most impressive form can be observed. The task of handwritten text recognition shows the need for more elaborate models of visual recognition that possibly incorporate top-down information, more ingenious mechanisms than plain methods of statistical pattern recognition. Nevertheless, humans are able to perform the task fast and reliably, indicating that human perception of writings follows different, but perhaps overlapping, procedures than currently employed in artificial systems.

Understanding these mechanisms or procedures is relevant for a better understanding of human visual perception itself, as well as for Artificial Intelligence with its objective of mimicking (possibly improving) human behavior. Addressing the cognitive basis of perception in handwritten text recognition, the neuronal development of visual written word processing can thus show a glimpse of the mechanisms at work, revealing valuable information about human approaches to handwritten text recognition.

1.4 Outline

Step by step a generic adaptive recognition system is developed that is automatically solving a handwritten text recognition problem. Chapter 2 introduces an adaptive feature extraction method that 'learns' a feature representation of a specific recognition problem: recognizing handwritten abbreviations of months extracted from a hundred year old logbook describing daily business of the

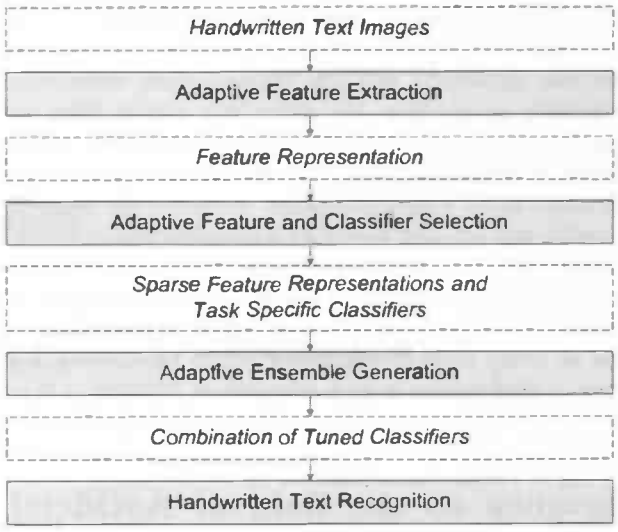


Figure 1.5: Adaptive handwritten text recognition outline. Incrementally, a adaptive recognizer is developed. Chapter 2 firstly deals with adaptive feature selection from which a feature representation results that is used in the following chapters. Chapter 3 describes feature selection and classifier tuning using the previously extracted features. Lastly, chapter 4 demonstrates a possible adaptive approach for decision making by combining the resulting specialized representations and classifiers into a fully grown recognizer system.

Dutch royal house, the 'Kabinet der Koningin' (1903). This method of feature extraction is a model that accounts for the first 100-200 milliseconds of visual processing in the ventral visual pathway [78, 79]. Chapter 2 deals with the first three steps of the design-cycle in pattern recognition (see figure 1.2). Experiments that ascertain the quality of the extracted features are presented.

Chapter 3 addresses neuronal development and adaptive feature and recognizer selection, dealing with step four and five of the design-cycle in pattern recognition. Mimicking neuronal development, a genetic algorithm [36] is used as an evolutionary method that sculptures a general recognition system into a domain-specific recognizer. Experiments are presented that determine the performance of this approach on both feature selection and recognizer selection.

Chapter 4 combines the previous methods and presents the generation of a large ensemble of specialized recognizers that together solve the particular problem. Using advantages of the feature extraction method and within the divide and conquer paradigm, a classification-scheme is formed. The performance of this overall system is presented. Finally, chapter 5 discusses the results, the benefits and drawbacks of the approach, and provides pointers for further research.

Chapter 2

Adaptive Feature Extraction

Visual object perception is a very complicated function, whose basis has interested researchers of many disciplines. This chapter will begin to address the problem of how mammals process visual information by identifying the structures, cells, and pathways of the visual system, and describing the specific functions that are performed by these elements. The focus will be on the lower-level processes that occur early in the ventral visual pathway where visual information is preprocessed before transmitted to higher levels.

With the three assumptions made in chapter 1 in mind, this chapter will lay the basis for a general procedure of visual information processing, providing a generic method of visual processing applicable to ample recognition problems, including handwritten text recognition. It will be argued that this generic method can then be employed in step two of the design-cycle of pattern recognition: feature extraction. Moreover, the resulting method of feature extraction needs less careful preprocessing, and most importantly, 'learns' to form a representation that enlightens differences between categories and similarities within categories in a semi-unsupervised way.

These characteristics of biologically motivated feature extraction are demonstrated in an empirical setting, showing the quality of the resulting features in the domain of handwritten text recognition. The method is applied to a variety of handwritten texts, ranging from handwritten digits and characters to handwritten words, making a case for the method to be a general method of object recognition that can later be specialized towards specific recognition tasks.

2.1 General Visual Object Recognition

The previously mentioned ventral visual pathway is dedicated to the processing of visual information with the goal of object recognition (as opposed to the dorsal visual pathway, which is thought to process information with the goal of performing actions [29]). After visual signals have left the eye via axons of the retinal ganglion cells, the signals cross to the opposite side of the brain in the optic chiasm. Past the optic chiasm, visual information is propagated through optic tracts, of which around 90% of the axons terminate in the lateral

geniculate nucleus (LGN) of the thalamus. Signals from the LGN travel to the primary visual cortex (V1), the first visual area in the occipital lobe. V1 in each hemisphere contains, like the LGN, a retinotopically organized map of the opposite visual field. Within the map, the central area of the visual field is magnified so that it receives a disproportionately large representation. The ventral stream then projects from V1 in a sequence to V2, V3, V4, to the posterior, central and anterior inferotemporal cortex [85].

First processing steps are thus performed in the primary visual cortex, a cortical area where Hubel and Wiesel [37] identified alternating simple and complex cells processing visual information. Recently, a theory has been proposed that accounts for the first 100-200 milliseconds of ventral visual processing in V1 [73]. This theory states that object recognition follows a feedforward path in the visual cortex. Starting in V1, in this feedforward path alternating simple and complex cells process visual information.

2.1.1 Standard Model of the Visual Cortex

Developed by Poggio and Edelman [70] in 1990, the standard model of visual processing accounts for well known facts about the ventral stream in the visual cortex [73] and [79]. First of all, visual processing is known to be hierarchical. It firstly realizes some tolerance to the different positions in which an object can appear in the visual field, and the size of the objects projection, and later tolerance to different viewpoints of the object and other transformations that might occur. Secondly, going downstream, the receptive fields of the simple and complex cells increase in size (i.e., the part of the visual field that enables a neuron to fire increases), allowing response to more complex visual stimuli. Thirdly, the authors postulate that the initial processing of visual information follows a feedforward path, which accounts for immediate recognition. Lastly, it is well known that "plasticity and learning occurs in all stages and certainly at the level of the inferotemporal (IT) and prefrontal cortex (PFC), the topmost layers of the hierarchy" [79].

Among the different methods of feature extraction methods currently available, this method positions itself as a method that combines appearance based feature descriptors (corresponding to small patches drawn from an image) and histogram-based descriptors. Patch-descriptors, such as described by [33, 89, 62, 84] are good shape descriptors, but do not account for in-variances regarding transformations, such as rotation and shift. Histogram-based descriptors on the other hand do account for possible in-variances. For example, SIFT-features [21] recognize previously seen, but transformed objects very robustly [61]. However, concerning generic object recognition, the amount of acceptable invariance prohibits exceptionally good performance [79].

Simplified, the standard model of the visual cortex constitutes four layers of alternating simple and complex cells. Simple cells in the first layer respond to stimuli in their receptive fields that are well described by Gabor functions [27]. Complex cells in the second and fourth layer gather their inputs from simple cell afferents from the previous layer, responding to the strongest activation among these afferents. The layers containing complex cells account for robustness with respect to positional and scale transformations. The third layer is key for introducing patch-descriptors subserving shape recognition. Patches of different sizes and rotations from previously seen shapes are matched to all possible crops

from the new image by means of radial basis functions [71].

2.1.2 Layer 1: Gabor Functions

The first layer consists of simple cells that take their input from the signals transmitted by the LGN. Neurons in the visual cortex propagate an increased response when specific stimuli are presented in particular parts of their 'visual field'. That is, specific stimuli presented in the receptive field of a neuron invoke increased response. The response of the neurons in the first layer, simple cells, are well modeled by Gabor functions, linear filters whose impulse response is defined by a harmonic function multiplied by a Gaussian function [45]. They are described by the following equation [27]:

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp(-\frac{(x'^2 + \gamma^2 y'^2)}{2\sigma^2}) \cos(\frac{2\pi}{\lambda} x' + \psi) \tag{2.1}$$

$$x' = x \cos\theta + y \sin\theta \tag{2.2}$$

$$y' = -x \sin\theta + y \cos\theta \tag{2.3}$$

The filter parameters $\lambda, \theta, \psi, \sigma$ and γ represent the wavelength, orientation, phase, width, and aspect ratio, respectively [27]. The first layer contains simple cells that implement equation (2.3), using different parametrization regarding width and orientation. Each pixel of the input image is the center of a couple of Gabor filters with different orientations and different sizes (measured in pixels), representing simple cells of different orientations and sizes. The filters are normalized so that they yield an activity between minus one and one.

Algorithm

1. For each orientation θ and size σ of a Gabor-Filter $g(\theta, \sigma)$,
   (a) Center $g(\theta, \sigma)$ at each pixel of the input image
   (b) Normalize $g(\theta, \sigma)$ so that it yields a response between -1 and 1
2. Calculate the response of all Gabor-filters $g(\theta, \sigma)$ at each pixel, this yields filtered $s1$-images for all values for θ and σ

2.1.3 Layer 2: Local Pooling

Complex cells in the second layer take their input from the simple cell afferents of the first layer. The complex cells respond according to the maximum activation among the input. Also complex cells focus their attention in a receptive field, usually twice as large as the receptive fields of simple cells in layer 1. According to Hubel and Wiesel [37], complex cells respond to oriented bars or edges anywhere in their receptive fields, showing some tolerance to position and size of the presented stimuli.

The receptive fields of the complex cells is restricted to neighboring simple cell afferent responses to similar orientation and similar size (recall that V1 is organized retinotopically). This way, a (limited) position- and scale-invariant representation is realized. In terms of the previously pointed out importance

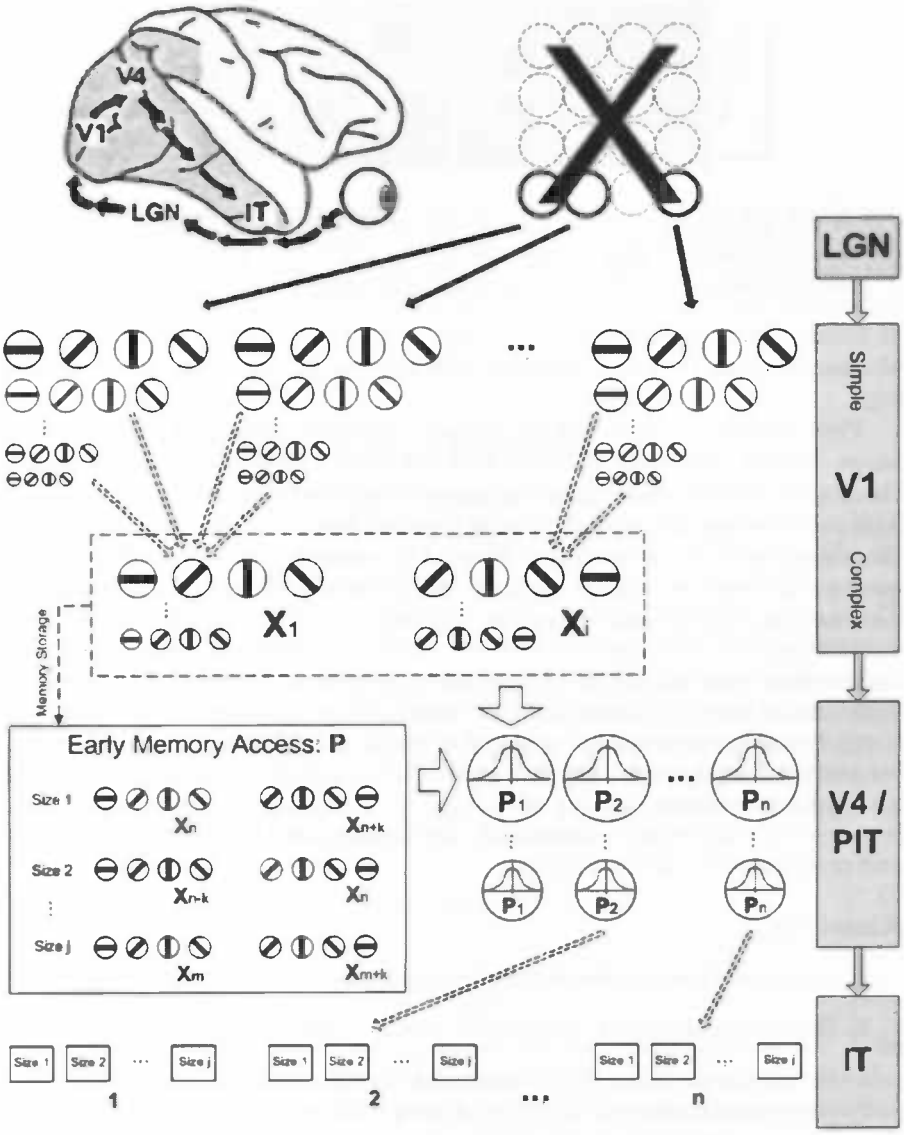


Figure 2.1: Overview of visual processing according to the standard model of visual processing. The scheme follows the ventral visual pathway running from the lateral geniculate nuclei to the inferotemporal cortex, via the primary visual cortex (V1), V4, and the posterior inferotemporal (PIT) cortex. V1-simple corresponds to the first layer of the standard model, while V1-complex, V4/PIT, and IT correspond to layers two, three, and four, respectively. Dashed arrows indicate max-operations.

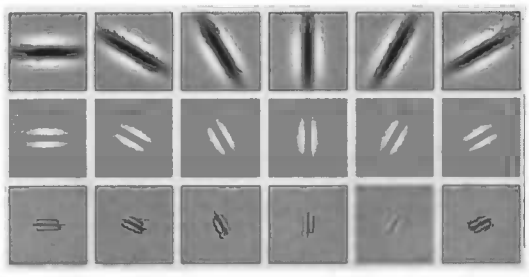


Figure 2.2: Examples of Gabor filter responses of different sizes and orientations, displayed as intensity map images. In the images, black and white correspond to positive and negative function values, respectively.

of dimensionality reduction, complex cells perform an important preprocessing step.

First, groups of simple cells are defined, containing filters of a particular size range. Within these scale bands of filters, a range r is defined that determines the size of a matrix of neighboring simple cells of all sizes in that scale band. This matrix of simple cell responses is then fed into one of the complex cells of the second layer. Only simple cell filters of the same orientation are fed into the same complex cell in order to preserve feature specificity regarding orientation. The complex cell then performs a max-operation on its input, responding to the strongest simple cell response within the matrix. In addition, complex cells take the absolute value of the simple responses, propagating an activation ranging from zero to one corresponding to the most extreme simple cell response, and thereby showing invariance to contrast reversal. Furthermore, the input matrices overlap in such a way that part of the simple cells feeding their response to a complex cell are also feeding their input to a neighboring complex cell. The responses of neighboring complex cells are then grouped together (subsampled) and propagated to the next layer.

Algorithm

1. Group all $s1$-images resulting from certain filter sizes together: scale bands
2. Define a pooling grid, a square of $(r \times 2)^2$ pixels, for each scale band
3. Divide the $s1$-images within each scale band into overlapping patches with size corresponding to the appropriate pooling grid
4. For each scale band,
   (a) For each orientation θ,
      i. For each $s1$-image i of orientation θ, within the scale band,
         A. Find the highest pixel-value $MAX(patch_i)$ of each patch within i
      ii. Find the complex cell responses $MAX(patch)$
      iii. Combine the complex cell responses into a $c1$-image, while grouping neighboring cell responses together

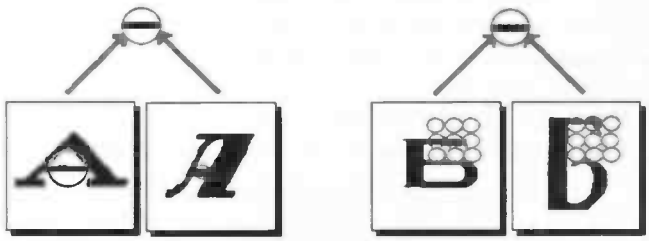


Figure 2.3: The result of processing after the second layer applied to four figures from Bongards Problem no. 100. Circles represent the receptive fields of simple cells in layer 1. The complex cells in layer 2 propagate those that respond most strongly. The two figures on the left show how limited scale invariance is achieved. The figures on the right demonstrate how tolerance to position is gained.

2.1.4 Layer 3: Radial Basis Functions

Simple cells in the third layer sample the afferent complex cells of the third layer in their receptive fields. Again, retinotopically organized complex cells allow these simple cells to sample the complex cell's output in a spatial neighborhood. In contrast to the complex cells in the previous layer, the receptive fields of simple cells in the third layer stretch over all orientations, combining bars and edges into more complex shapes.

In addition to their sampling behavior, simple cells in the third layer implement a function similar to a radial basis function (RBF [71]) [69]. Radial basis functions are a major class of neural network model, comparing the distance between input and a prototype [8]. For each sampled afferent complex cell activation pattern $\mathbf{X}$, a basis function is introduced and the output is then said to be equal to the weighted sum of the basis functions:

$$h(\mathbf{X}) = \sum_n w_n \exp(-\beta|\mathbf{X} - \mathbf{P}|^2) \tag{2.4}$$

$$\beta = \frac{1}{2\sigma^2} \tag{2.5}$$

The simple cells in the third layer implement the above equation together, each representing one of the basis functions $\phi = \exp(-\beta|\mathbf{X}-\mathbf{P}|^2)$, where σ defines the sharpness of tuning and $\mathbf{P}$ is the center of the RBF, representing the prototype. The weighted sum is replaced by global pooling of the next layer.

During training, random patches of different sizes are drawn from an image at the level of the second layer. Each patch contains all orientations. The third layer compares these patches by calculating the summed euclidean distance between the patch and every possible crop (combining all orientation) from the image of similar size. This comparison is done separately with each scale-band representation in the second layer.

Algorithm

1. During training only:
   (a) Select n random $c1$-images
   (b) For each of these images,
      i. For predetermined different patch-sizes
         A. Define a random position (x, y)
         B. For each scale-band, store the patch $\mathbf{P}$ containing all orientations
2. During normal processing:
   (a) For each stored patch $\mathbf{P}$,
      i. For each possible patch $\mathbf{X}$ (combining all orientations) from a $c1$-image within a certain scale-band,
         A. Calculate the response of a $s2$-cell $exp(\frac{-1}{2\sigma^2}|\mathbf{X}-\mathbf{P}|^2)$ for each scale band

2.1.5 Layer 4: Global Pooling

Lastly, the fourth layer contains again complex cells, similar to those in layer two. They replace the weighted sum in equation 2.4 with a output response according to the strongest activation (minimum distance) of the simple cell afferents of the third layer in their receptive fields. The receptive fields of these cells stretch over all simple cell scales and positions. A complex cell in this layer will respond according to the most active simple cell of the previous that is selective for the same combination of oriented bars, but regardless of its scale or position.

Algorithm

1. For each $\mathbf{P}$
   (a) Find the minimum distance within the scale bands, sizes, and positions
   (b) Add this distance to a feature vector

2.1.6 Stimulus-Driven Feature Extraction

Traditionally it was assumed that access to memories of objects occurs only after grouping and segregation processes have produced the figures or objects in the visual array [7]. Contrary to this assumption, neurophysiological investigations showed that processes aiding quick access to object memories should be considered early visual processes (e.g., [68]). The abovementioned prototypes fulfill the role of object memories retrieved early in visual processing.

Early memory access subserves learning in visual processing. The process of feature extraction performed by this mechanism of visual processing thus has the ability to adapt to specific stimuli previously encountered. The resulting activations are depending on the kind of visual input there has been before,

expressing the representation of new visual input in terms of previous experiences. This characteristic of the standard model of visual processing makes a case for the existence of a general mechanism of visual object recognition. If indeed previous experience determines the resulting representation, using this mechanism of visual processing as a method of feature extraction in combination with domain-specific prototypes results in a domain-specific representation of visual stimuli, generated by a general mechanism. This domain-specific representation then ensures easier classification and hence facilitates recognition. To test this hypothesis, this chapter presents experiments testing the quality of the representation at the level of layer 4 in the model.

2.2 Quality of the Features

2.2.1 Introduction

To assess the quality of the feature extraction method described above with respect to the domain-specific recognition problem of handwritten text recognition, three experiments are set up, classifying handwritten digits, capital characters, and abbreviated months. The nearest neighbor classifier is used to determine how separable the resulting feature space is when attempting to classify one of the classes out of the rest. That is, the performance is determined when predicting class-membership by assigning the class-label of the nearest neighbor in feature space to a unknown image (see figure 2.7).

2.2.2 Methods

The performance of the feature extraction method is determined in three experiments. For each of the experiments, data is gathered and preprocessed, feature extraction is performed, and the features are used in a straightforward classification task. Note that no feature selection takes place and that only minimal preprocessing is done.

The data is gathered from an over 100 year old logbook called 'Het Kabinet der Koningin' (1903), describing daily business around the Dutch royal house. The writings in this book are annotated with dates, presenting hundreds of handwritten numbers and abbreviated months throughout the book. Capital characters can be found anywhere in the written text. The handwritten numbers and capital characters are extracted from the book and labeled with an on-line tool used by volunteering users to label segmented writings. Segmentation is performed by determining 'cut-points' wherever there is white space exceeding a threshold. The segments are then presented to a user of the on-line tool and are labeled. The abbreviated months are extracted from a column on the left-side of each page in the book. The location of these columns is determined with a layout-analysis. In this layout analysis large vertical and horizontal lines are searched for, and by making use of the standard layout of pages in the book, the pages are segmented into areas containing particular information (see figure 2.4). Before extracting the writings from the column-images, the images are binarized using Otsu's method [64]. In addition, lost parts of the text are reconstructed applying thickening to the binarized images. Using projection-histograms, projecting black pixels on the vertical axis of the image, peaks in

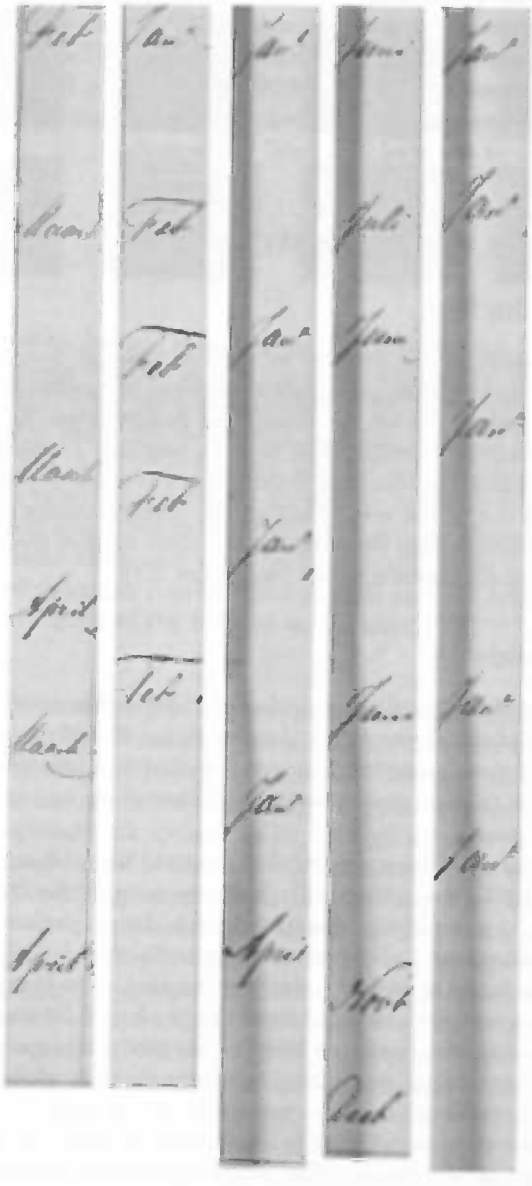


Figure 2.4: Columns in 'Het Kabinet der Koningin', containing month information. Handwritten months are automatically extracted from these columns.

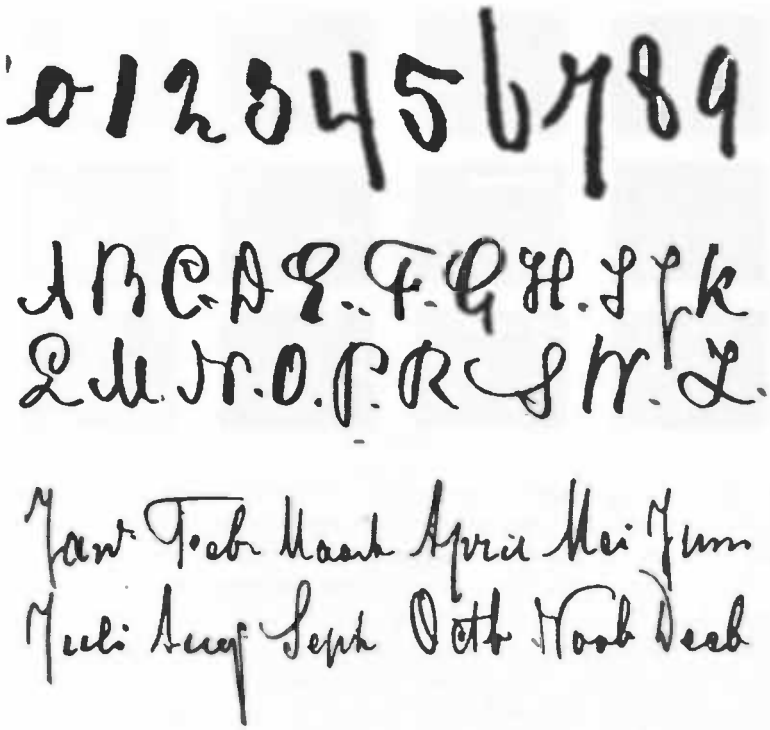


Figure 2.5: Examples of the images used to assess the performance of the features extraction method. From top to bottom: digits (ten classes), capital characters (20 classes), and abbreviated months (12 classes). All images are extracted from the 'Kabinet de Koningin'-logbook (1903) and those shown here are desheared and normalized. Images used to determine the performance of the features in simple classification tasks are not normalized and may contain more noise.

the smoothed histogram determine the location of written text. The text is then cut out at those places where the gradient of the smoothed histogram around the peaks starts approaching zero. Finally, the segmented text is desheared using horizontal projection histograms. Iteratively, the segmented text-image is sheared counter-clockwise from 45^o to -45^o with an affine-transformation. Projecting the black pixels in the horizontal axis for each transformation, the desired transformation produces a projection histogram with the highest peaks. The resulting images of the three datasets are shown in figure 2.5.

To represent the gathered and preprocessed images by features in accordance to the standard model of the visual cortex, an implementation of this model by Serre et al. is used (for a detailed description of the implementation, see [78, 79]). Moreover, the parameters used in the experiments are similar to those used in the experiments described in [79] because they represent biologically plausible values (see table 2.1). Simple cells in layer 1 are parametrized using empirically determined values, so that the tuning properties of the Gabor functions match

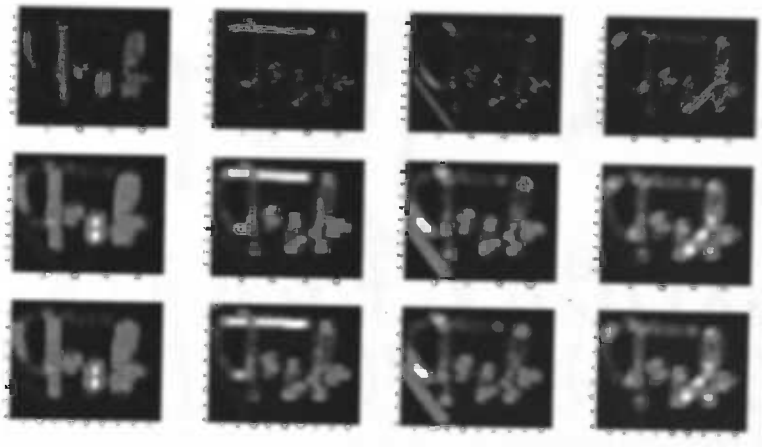


Figure 2.6: Intensity images representing the distribution of activation over one of the test-images. Higher activation is represented by lighter gray values. Columns from left to right have orientations of 0°, 90°, −45°, and 45°, respectively. The top row shows the activation over the image in the first layer, the center row shows the result after pooling, and the bottom row after re-sampling. The images in the third column clearly show noise in the form of a diagonal bar, a desheared vertical line. This artifact will probably be filtered out in following representations (layer three and four) because it will cause a mismatch with a prototype. One of the other representations with another orientation probably elicits a higher activation of the RBF in layer three.

those of most parafoveal simple cells in the primary visual cortex [19, 20, 77, 76]. The authors of [78, 79] sampled the parameter space by applying gratings, bars, and edges to the differently parametrized filters, selecting those filters that resemble findings of studies regarding the visual cortex of monkeys best. The simple cells appear with sixteen scales, and are arranged accordingly. Four orientations are used: 0°,90°,−45°, and 45°, resulting in 64 different layer one simple cells. The resulting images from the first layer are divided into scale bands, each scale band containing two neighboring filter sizes, resulting in eight scale bands for sixteen different scales. Complex cells in layer two sample the simple cell afferents by taking input from one of the scale bands and subsampling the filtered images with a grid-size that correlates with the filter sizes used in the scale bands. Each complex cell propagates the maximal activation within the results of the two filter sizes, resulting in an image that combines those elements of the image that maximally activated the previous layer. This is done for each orientation separately. Figure 2.7 shows an example of the representations of an image as it is propagated through the feedforward path of the first two layers.

The third layer constitutes the learning stage of the feature extraction method. Presenting a prototype to simple cells implementing a radial basis function for comparison, the cells respond corresponding to the match between the prototype and the image that is processed. Prototypes are represented similar to the partly processed image. That is, prototypes are patches (size 4×4, 8×8, 12×12, and 16×16 pixels) from images processed until the level of layer two, the output of the complex cells in the second layer. For each class a set of ten

Table 2.1: Standard model Parametrization [79]

Layer 1			Layer 2		
Filter-size	Gabor σ	Gabor λ	Scale-band	Pooling-grid	Overlap
7×7 9×9	2.8 3.6	3.5 4.6	Band 1	8×8	4
11×11 13×13	4.5 5.4	5.6 6.8	Band 2	10×10	5
15×15 17×17	6.3 7.3	7.9 9.1	Band 3	12×12	6
19×19 21×21	8.2 9.2	10.3 11.5	Band 4	14×14	7
23×23 25×25	10.2 11.3	12.7 14.1	Band 5	16×16	8
27×27 29×29	12.3 13.4	15.4 16.8	Band 6	18×18	9
31×31 33×33	14.6 15.8	18.2 19.7	Band 7	20×20	10
35×35 37×37	17.0 18.2	21.2 22.8	Band 8	22×22	11

prototypes is selected in all experiments. From these prototype sets (containing 100, 200, and 120, respectively, partly processed images) 100 random samples are drawn, resulting in an average of ten, five, and eight prototypes from each class, respectively. Whether these prototypes are appropriate is unknown and it is important to point out that the performance of the resulting feature representation depends strongly on the quality of the randomly selected prototypes. Choosing the prototypes randomly ensures the performance is not depending on manual optimization, for instance aiding the feature extraction method by presenting it with appropriate prototypes. The set of prototypes can be considered unlabeled, which means that when extracting features for any of the classes uses prototypes of all classes. Hence, the feature extraction method can be considered unsupervised.

At the level of the second layer images are represented in four different orientations, thus also the prototypes are represented in four orientations. Cells in the third layer, representing basis functions, then calculate the (Euclidean) distance between the image and a prototype for each patch-size and scale-band separately. The fourth layer then calculates a global maximum, replacing the weighted sum of equation 2.4 by a max-operator, resulting in four scalar values for each prototype of particular size. After processing the image, it is represented by a feature-vector of size 400.

The features are tested using an increasing size of an labeled set of examples, and a 100-fold cross-validation task is performed. For each fold, the data is split into a labeled and unlabeled set by randomly sampling the examples (as a variant of normal cross-validation, where the data is split up into k mutually exclusive folds). The labeled set contains up to 50, 40, and 50 examples per class, and the unlabeled set contains a minimal total of 1421, 3529, and 2941 examples in case of the digits, capital characters, and months, respectively. It is assured that the prototypes are not found in the unlabeled set. The labeled set are then used to apply a Voronoi tessellation (see figure 2.7 and 2.8) on the 400-dimensional feature-space. Examples of the unlabeled set are then assigned

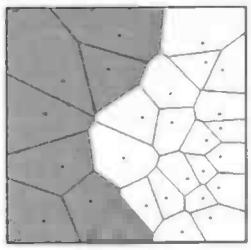


Figure 2.7: Voronoi diagram demonstrating two-dimensional nearest neighbor classification. Gray areas belong to class I, white areas to class II. The nearest neighbor algorithm assigns the label of class I to a unknown image when the (two-dimensional) feature representation is a coordinate falling in one of the gray areas, and class II otherwise. In a Voronoi diagram, cells consist of the points closer to one particular object than to any others [87].

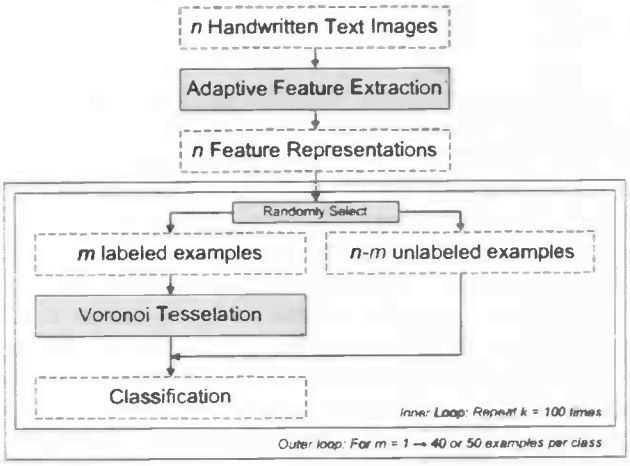


Figure 2.8: Experimental Procedure to assess the performance of the feature extraction method with a 1-nearest-neighbor classifier.

the label that corresponds to the subspace in the tessellation surrounding the closest example of the labeled set. Furthermore, the Voronoi tessellation is calculated using Euclidean distances, since this distance measure is used in the basis functions of the third layer of the feature extraction method.

2.2.3 Results & Discussion

Tables 2.2 summarizes the results of the handwritten digit classification task, table 2.3 for handwritten capital character classification, and lastly, table 2.4 summarizes the performances concerning the handwritten abbreviated month classification task.

Table 2.2: Handwritten Digits, Train=50

Class	$P(Correct)$	# Correct	# Total
0	0.9265	63	68
1	0.9459	105	111
2	0.9532	224	235
3	0.9412	272	289
4	0.9535	205	215
5	0.9767	126	129
6	0.9570	89	93
7	0.9362	44	47
8	0.8793	102	116
9	0.9068	107	118
Total:	0.9409	1337	1421

Averages of 100-fold cross-validation

Table 2.3: Handwritten Capitals, Train=40

Class	$P(Correct)$	# Correct	# Total
A	0.9652	277	287
B	0.9152	453	495
C	0.9239	267	289
D	0.8993	134	149
E	0.8866	86	97
F	0.9583	23	24
G	0.9539	269	282
H	0.9667	436	451
I	0.8462	11	13
J	0.9754	357	366
K	0.9125	146	160
L	0.9175	89	97
M	0.9818	323	329
N	1.0000	5	5
O	0.9730	36	37
P	0.8842	84	95
R	0.8153	128	157
S	0.8889	32	36
W	0.9860	141	143
Z	0.9412	16	17
Total:	0.9388	3313	3529

Averages of 100-fold cross-validation

Table 2.4: Handwritten Months, Train=50

Class	$P(Correct)$	# Correct	# Total
Jan	0.9902	403	407
Feb	0.9918	364	367
Mar	0.9715	307	316
Apr	0.9954	216	217
May	0.9907	214	216
Jun	0.9878	404	409
Jul	0.9918	363	366
Aug	1.0000	34	34
Sep	1.0000	177	177
Oct	0.9942	171	172
Nov	0.9934	151	152
Dec	0.9907	107	108
Total:	0.9898	2911	2941

Averages of 100-fold cross-validation

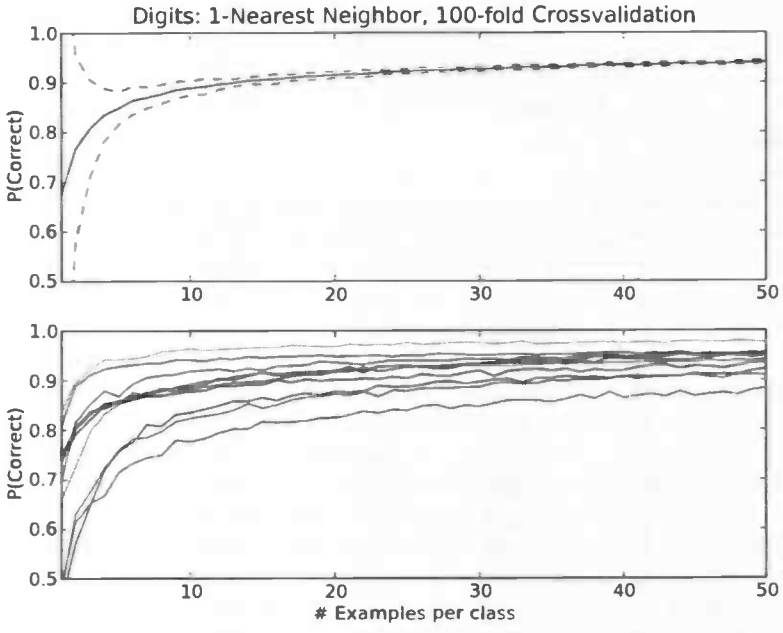
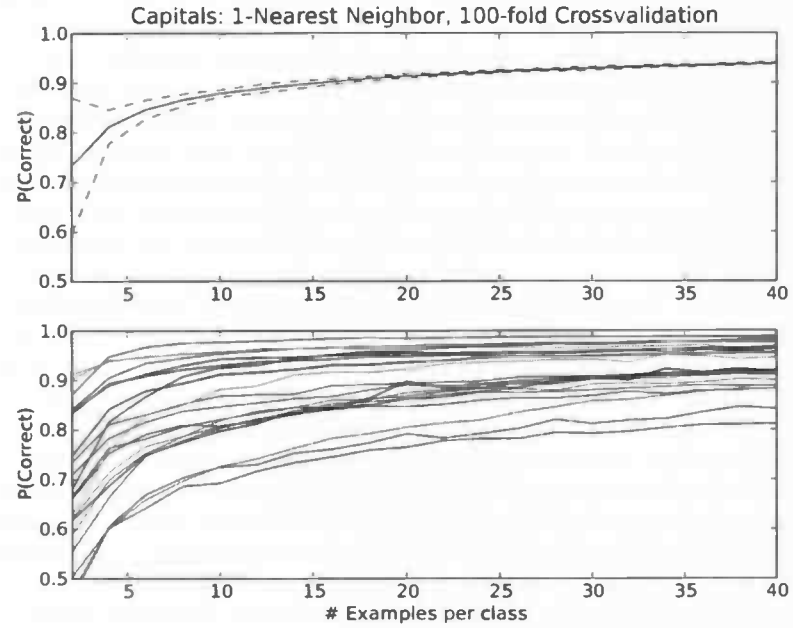
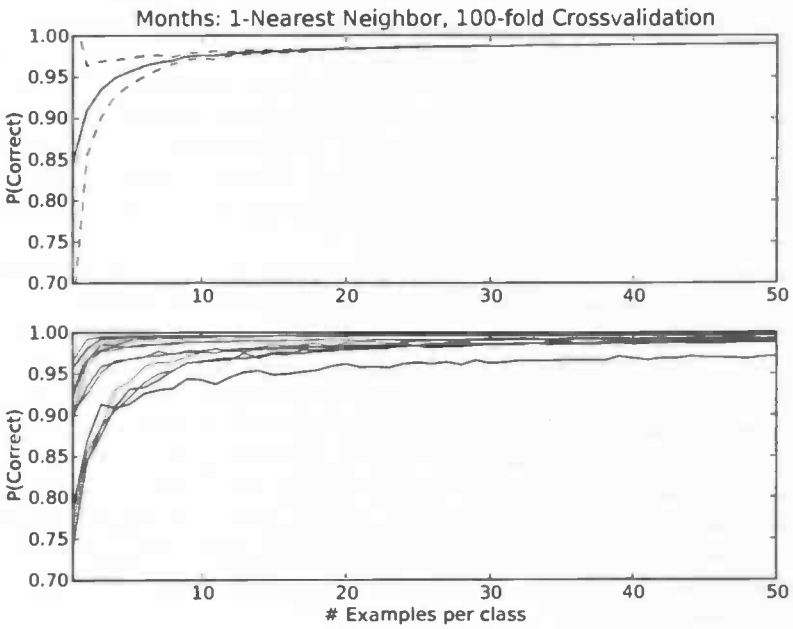


Figure 2.9: Performance of the feature extraction method on handwritten digits, capital characters (top, right page), and abbreviated months (bottom, right page). The quality of the features is measured by the average ratio of correct responses in a 100 random-folds cross-validation task. The top figure shows the overall average ratio and the standard deviations (dashed) regarding the 100 folds in the task. The averages and standard deviations are calculated over each fold, over all classes, indicating the confidence in the reported means. The bottom figure shows the average ratios per class.


Capitals: 1-Nearest Neighbor, 100-fold Crossvalidation
P(Correct)
1.0
0.9
0.8
0.7
0.6
0.5
5
10
15
20
25
30
35
40
Examples per class


Months: 1-Nearest Neighbor, 100-fold Crossvalidation
P(Correct)
1.00
0.95
0.90
0.85
0.80
0.75
0.70
10
20
30
40
50
Examples per class

As the tables indicate, the performance of the handwritten months dataset rises above the performance of both the handwritten capital characters and digits datasets. This difference is due to two issues. Firstly, the within-class variability of the abbreviated months is low regarding the other two datasets. The abbreviations of the months written in the columns on the right side of the pages annotate the text, and are written down more carefully. Recall that these abbreviations are extracted from the right columns only and that the other two datasets are gathered from all writings in the logbook. Secondly, the between-class variability of the abbreviated months is high regarding the other two datasets. This is due to the nature of the data, words carry more information than characters or digits and the images are therefore inherently more complex. The more specific the informational content of a writing, the more complex representations are needed to discriminate between the representations.

Nevertheless, the performances are certainly far from bad. Especially the performance on more complex images of handwritten text makes a case for the use of this feature extraction method representing immediate holistic recognition. Moreover, the method, originally designed for object recognition (e.g., animals or cars) [78] did not need any additional consideration regarding the parametrization in order to recognize handwritings very well. Concerning the hypothesis that feature extraction according to the generic mechanism of visual processing in the ventral stream of the brain results in easier classification caused by adaptation towards a specific domain, indeed the resulting representations allow for high performance using a very simple classification method.

Figure 2.8 shows that the number of examples needed for robust classification is fairly low. Ranging from one to fifty examples in case of the handwritten digits and one to forty in case of the capital characters, the performance meets its ceiling quickly, certainly in case of the handwritten abbreviated months. This suggests that the features that were learned by the extraction method allow for good generalization. In addition, the high performance with a one nearest-neighbor classifier tells that the representation separated the classes well. Employing more 'intelligent' classifiers is hypothesized to maintain or increase the performance of a nearest-neighbor classifier (whose performance is hard to beat), while using more sparse representations and generalizing the solution to the classification problem.

Serre et al. [79] demonstrate that larger feature vectors perform better in recognition tasks concerning 'real' objects, and feature vectors used here are reasonably large. The reason for this is that the feature extraction method relies on template-matching of randomly drawn patches. Increasing the sample-size (size of the feature-vector), increases the likelihood good features are found. Theories concerning the amount of features to be used state that more features not necessarily imply better performance [2]. Indeed, considering the random choice of the prototypes, not every prototype may be equally well suited to represent the categories, and make the differences more salient. The next chapter hypothesizes that the brain is naturally equipped with mechanisms that perform a task equivalent to feature selection, adapting the recognition process even more to a specific domain.

Chapter 3

Adaptive Feature and Classifier Selection

The previous chapter made a case for the possibility that a generic mechanism of object recognition is able to adapt itself to domain-specific stimuli. The chapter showed that step two in the design-cycle of pattern recognition can be automated, when a feature extraction method is chosen that is inspired by visual processing in accordance with the ventral stream of the brain. It not only realizes feature extraction for domain-general object recognition, but also for domain-specific stimuli such as written texts. Indeed, it seems that treating holistic handwritten text recognition as just a particular kind of object recognition as it is performed in nature already results in representations that are easy to classify.

Now it is time to assess the third assumption made in the introduction, s that there are other mechanisms in the brain that specialize recognition towards domain-specificity even more. Addressing the domain of handwritten text recognition, the next steps in the design-cycle of pattern recognition are feature selection and selecting an appropriate classifier. Since feature selection and selecting the right parametrization for a mapping function depend on the representation of the problem, these steps lend themselves well for specialization towards a particular domain. Recall that experience is a key issue in recognition, both in forming the appropriate representation and the method of classification. The objective is to develop a system that adapts itself to a specific recognition problem by learning how to represent the problem and how to perform the classification task. Experience is the driving force behind self-adaptation and this chapter provides a method to use experience in the steps of feature and classifier selection, inspired by natural mechanisms.

Feature selection is a key step in the design cycle of pattern recognition since it eludes the curse of dimensionality. Moreover, feature selection can improve recognition performance. As mentioned previously, the full representation of an image might be confusing, while disregarding some elements eliminates the confusion. This introduces the concepts of relevant and irrelevant features [42], where relevant features are wanted because they subserve recognition, and irrelevant features are not, since they increase dimensionality and may introduce confusion. When specializing onto a specific recognition task, also nature has

figured that it is better to alter the representation in order to reduce confusion.

The choice of the classifier can increase the performance. For instance, when applying a linear classifier to a problem that is not linearly separable, the performance will be low. When selecting a classifier, the aim is always to find the classifier that generalizes the solution to the recognition problem best, resulting in the best performance on newly presented examples. When there are outliers in the training set, it is best to ignore them (and if they are fulfilled not to be outliers, too little data was gathered).

Changing the representation of an object or writing by selecting relevant features, the recognition problem might suddenly become linearly separable or at least less complex. This implies that also a previously selected classifier might not be appropriate anymore, and hence the recognition mechanism is in need of another. Therefore, it is convenient to perform both selections together. In the brain, neurons perform classification by implementing some kind of classification function. In addition, neurons select their input by having their dendrites 'connected' with afferents of other cells. Conveniently, at the level of the connections between neurons and among the neurons themselves changes take place, such as the elimination of connections, cells, or growing connections or cells.

Subjecting artificial neural networks to a method that mimics the mechanisms that give rise to neuronal development and plasticity, feature and classifier selection can be performed. Moreover, since neuronal development and plasticity are thought to be influenced by experience, feature and classifier selection can find itself adapting to a particular domain, in this case handwritten text recognition.

3.1 Neural Darwinism

The third assumption postulated in the introduction states that neuronal development and plasticity can be seen as a local evolutionary process, acting on the level of neurons. This assumption is also advocated by other researchers [23, 60] arguing that developmental selection and experiential selection account for the production of adaptive behavior [23]. Gerald Edelman termed the combination of these mechanisms Neural Darwinism. In the introduction, the importance of taking developmental processes into account when modeling recognition behavior was pointed out. Evidence from studies in neuronal development and neuronal plasticity provided good arguments that indeed these mechanisms subserve learning of recognition. It was also pointed out that these mechanisms can be modeled by evolutionary algorithms, thereby allowing for the possibility to further specialize the visual processing mechanism towards a handwritten text recognizer. This section presents the rationale of employing evolutionary algorithms to specialize a recognizer.

3.1.1 Neuronal Development

Developmental research concerning the development of the visual system has found that infants are not able to perform complex visual tasks immediately after birth. The sensitivity of infants for contrasts [3] and orientation [91] gradually increases, there is even rivalry between the two eyes before fusion of both visual fields takes place [34]. Furthermore, the self-organizing behavior of the

cortex that also applies to the visual cortex implies that further organization is necessary after birth. Progressive phenomena such as birth and proliferation of neurons, migration, and local dendritic branching, but also regressive phenomena like cell death, axonal retraction and synaptic pruning are all part of the developing cortex, and after the first months of life, neural connections are rendered obsolete on a large scale.

These mechanisms are not only present when the brain is still in its youth. The existence of neural plasticity suggests that these mechanisms are still at work when necessary. The question is not whether the brain has the ability to change, but how it 'knows' when and how to change. How does the brain develop domain-specificity, 'modules' that deal exclusively with a single type of information, like face recognition and recognition of writings for humans, or fly detection for frogs? Of course, mechanisms can be at play that, based on genetics, following innate blue prints, develop these domain-specific functionalities.

In mosaic development, cells develop independently, determined by genetic factors. This type of development has major drawbacks for larger, more complex animals and humans. Firstly, mosaic development lacks flexibility facing abnormal conditions, and the cells will not develop properly. Moreover, in environments that are likely to change, hardwiring development is simply not viable. Secondly, it is not feasible to specify everything in the development of more complex creatures.

In fact, humans and most other complex creatures rely more on regulatory development [24]. This type of development depends on interactions between cells, activations induced by experience. Intrinsic factors do determine roughly what possibilities there are for these interactions, they do not define the final outcome. Regulatory development allows for greater flexibility, plasticity and specialization [25].

3.1.2 Competition

Regulatory development in the opinion of Edelman [23, 24], constitutes a mechanism he calls neuronal group selection. Neuronal groups are large collections of neurons that are basic units of selection and have the possibility to enhance the adaptive responses of the group as a whole by changes in the synaptic strengths. In addition, the groups themselves are dynamic and their characteristics are affected by the nature of the signals they receive. The feature extraction method presented in the previous chapter can be considered such a neuronal group, responding according to the signals it receives and received before.

Here, it is presupposed (and also by Edelman [24]) that regressive specialization factors select responses from such neuronal groups. Now consider neuronal classifiers a neuronal group. Also this group is affected by the signals it receives, being either (bottom-up) signals from a visual processor or (top-down) feedback on its performance. The group can undergo dynamic changes in its characteristics or can be selected to be part of the recognition mechanism. In terms of pattern recognition, neuronal group selection as Edelman describes it, enables neuronal feature and classifier selection.

Nature acts according to the laws of evolution, which is the primary mechanism that produced life as it is. Darwin referred to this mechanism as 'Survival of the fittest', describing it on a macro-level, the level of species. Disregarding the metaphysical question why this mechanism exists, there is no reason to

assume that this natural mechanism does not act on other levels as well, for instance on a neuronal level. Regressive developmental phenomena and neuronal plasticity suggest that indeed the structuring of the brain acts according to the mechanism of survival of the fittest, or simply competition. Neurons or groups of neurons have the ability to adapt to the signals they receive and compete for activation. If they are not provided with sufficient activation for a longer time, they die, meanwhile trying to adapt by relaying their connections. Although this form of survival of the fittest may not match Darwinian evolution, the evolutionary mechanism in principle still applies. This rationale allows to simulate the regressive factors by mimicking the process of evolution.

3.2 Genetic Feature and Classifier Selection

Feature and classifier selection can be considered a optimization problem, and nature solves this type of problem with the mechanism of evolution. Acting on a neuronal level, a process similar to evolution solves optimization problems like domain-specificity in handwritten text recognition. Models of generational evolution have long been used to solve various optimization problems, of which genetic algorithms [36] were one of the first. Genetic algorithms simulate evolution by representing individuals by a genetic code that may undergo several changes. Mimicking generational evolution, individuals propagate and their genes mutate, generating new individuals which may be optimal solutions. The driving force behind genetic algorithms is its fitness function, determining which of the individuals are allowed to propagate. When a new individual seems to have a high fitness, the evolutionary mechanism allows itself to explore the possibilities of the individual until other individuals appear to be more fit. This way, the solutions become better and better, and while solving a optimization problem, the search process converges towards an optimal solution.

Here, an individual is represented by a neuronal group and its dendrites, taking input from afferents of other groups. As mentioned before, both the method of feature extraction described in the previous chapter and a classifier can be seen as a neuronal group. Taking an artificial neural network (ANN) (consisting of multiple perceptrons [75]) to fulfill the role of classifier, the similarities become even more striking. The number of input nodes of the neural network determines how many input signals it receives, by its connections with feature extractor afferents. This determines which input signals are selected to participate in a so-called population code [66], implying sparser population coding when less active connections are involved. These individuals can thus fulfill the role of both feature selector and classifier.

The use of genetic algorithms to optimize feature selection and artificial neural networks has previously been shown to provide good results [17, 31, 63, 90]. What is new in this approach is the integration of both steps in the design-cycle of pattern recognition. In the following subsections experiments are described where the approach of adaptive feature and classifier selection is applied together with the features extracted with the method described in chapter 2.

3.2.1 Methods

When evolving artificial neural networks that integrate feature selection and classification, the objective of this experiment is to minimize both the classification-error and the dimensionality of the representation used for classification. The feature extraction method of chapter 2 is used to prepare the images of the previously introduced abbreviated months dataset for classification. The feature vector resulting from this extraction method contains 400 values, each representing the match of one of the 100 randomly sampled prototype patches with an image-crop of specific size. The resulting feature representations are then split into a train, test, and validation set, making sure the prototypes are drawn from the train set. The train and validation sets have a fixed number of examples per class: 50 and 40 examples, respectively. The remaining feature representations belong to the test set.

Integrating feature selection and classification, the artificial neural network consists of a gate equal to the size of the feature vectors resulting from extraction, and a fully connected multi-layer feedforward backpropagation neural network [75], consisting of an input, hidden, and output layer. The gate consist of product units [75] implemented simply with a vector γ allowing only the binary values, 0 and 1 (see figure 3.1). Rummelhart and McClelland refer to this combination of product- and summation-networks as sigma-pi units, and each input node of the neural network is described as a sigma-pi unit. When i indicates an gated afferent connection to an input node of the neural network, the response of the sigma-pi input units is described by:

$$y = \sum_i \gamma_i \times a_i \tag{3.1}$$

The network dynamics of the multi-layer backpropagation network are described by the following equations. The errors on the output nodes are calculated with equation 3.2: δ_i = error, t_i = output node's target value, o_i = the node's response, and $f'(net_i)$ being the derivative that modulates the discrepancy between target and output.

$$\delta_i = (t_i - o_i) f'(net_i) \tag{3.2}$$

The weights w_{ij} between receiving (i) and sending (j) hidden nodes are adjusted with a rate of change η by:

$$\Delta w_{ij} = \eta \delta_{ip} o_{jp} \tag{3.3}$$

The δ_{ip} of the hidden nodes are calculated by collecting the error of the k output nodes:

$$\delta_i = f'(net_i) \sum_k \delta_k w_{ki} \tag{3.4}$$

In the right-hand equation, δ_{ip} reflects the error on node i for input-pattern p and o_{jp} makes sure the weight-change on the connection from sender j to receiver i is also proportional to node j's activation.

The learning-rate shows a gradual decrease according to equation 3.5 (with ϵ representing the current epoch, ϵ_T the total number of epochs, and η representing the initial learning-rate), while the ϵ_T is kept small to avoid overfitting

(50 epochs for the *Jan* vs. *Jun* dichotomy, 100 epochs for the *Maart* vs. *April* dichotomy).

$$\eta \rightarrow \eta \frac{-\epsilon^3}{\epsilon_T} \tag{3.5}$$

The structure of the network is described by one hidden layer, with a variable number of hidden units h. When γ_i in equation 3.2 equals zero, the input unit i does not receive any input, rendering the unit obsolete. The number of input units is therefore set equal to $\sum_i \gamma$. Weights are initialized with random values between -5×10^{-2} and 5×10^{-2}, sampled from a uniform distribution. The number of output units is set to two, allowing responses between -1 and 1. Therefore, the targets used here are represented by tuples, $(1, -1)$ or $(-1, 1)$ corresponding to the dichotomy of two classes.

For the optimization of both feature selection and performance a straightforward implementation of a genetic algorithm [36] is used. The genetic code susceptible for mutation and uniform cross-over operations consists of two chromosomes determining the phenotype of the gate and structure of the network. One chromosome consists of a binary string which corresponds to γ, the other is a Gray-coded [30] bit string describing the number of hidden units of the network, thereby avoiding sudden large increase or decrease of h. The fitness of the network is determined using a fitness-function described by the sum of the accuracy, normalized to the interval $[0..\frac{1}{2}]$, and the number of feature-values used, also normalized to the interval $[0..\frac{1}{2}]$ with respect the total number of feature-values. The structure of the network regarding the number of hidden nodes is not considered by the fitness-function for the reason that in chapter 4 ensembles (which have to be diverse) are created from the population of the last generation. Even though, the number of hidden nodes is constrained by the appearance of the gate, while undirected structural cross-over and mutation assure the population is diverse concerning the hidden layers.

The genetic algorithm uses two selection phases, the first selecting two consecutive parents from a shuffled parent-pool obtained by tournament selection ($k = 2$, without replacement) of the population of parents and children. In each generation, parents produce offspring according to the following procedure: With the likelihood determined by the crossover-rate, a random cut is made at the same gene for both parents in each of the two chromosomes. A first child receives the first slice from the first parent and the second from its other parent. A second child receives an inverse copy ($1 \rightarrow 0$, $0 \rightarrow 1$) of the resulting gene, rendering the procedure position-independent. When no cross-over takes place, a Bernoulli-trial determines from which parent the first child receives its genes. The second child receives again an inverse copy of that gene. The resulting population of parents and children then enters a tournament that determines the parent population of the next generation by pair wise competition.

When parents reproduce children, the mutation-rate determines the likelihood by which mutation-operators change the genetic code of the children. This rate reflects the likelihood that on average, one gene is mutated for each child. With regard to the first chromosome describing the feature-selection, two genes at different positions in the gene-string can be swapped or a gene suffers a bit-flip. Often used mutation-operators such as insertion and deletion do not apply to the first chromosome since the network's gate-size is fixed to the size of the features produced by the extraction method. The second chromosome, which

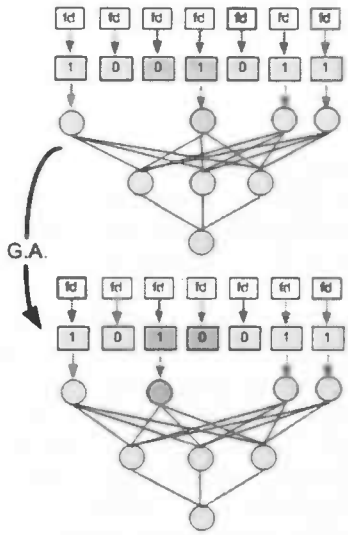


Figure 3.1: Tuning gated neural networks. A genetic algorithm transforms a random base classifier to a more optimal configuration.

describes the number of hidden units of the network, is susceptible for both flip- and swap operators, as well as insertion and deletion.

Each evolution is initiated with a population of random instantiations of gated neural networks, with $\sum_i \gamma \approx \frac{1}{12} \times 400$ feature values, and h is constrained by $h \leq \sum_i \gamma$.

3.2.2 Results & Discussion

The discussed results are derived from the evolutionary process regarding gated-networks that attempt to discriminate between the handwritten abbreviated months *Jan* and *Jun*, and *Maart* and *April* (lit.). Figure 3.2 shows the tasks represented as Bongard problems. Figures 3.3 and 3.4 show the evolutionary process of tuning for *Maart* vs. *April*, and *Jan* vs. *Jun*, respectively. This process is described by the average error, number of selected features, and fitness, as well as the best of each of these variables separately.

For comparison, the dichotomy *Jan* vs. *Jun* is presented to the feature selection algorithm RELIEF that is implemented according to RELIEF-A as presented in [52].

1. Initiate a vector with weights with length equal to the number of features N to zero: $w = 0$
2. For $t = 1..T$,
   (a) pick a random example x from the set of labeled feature vectors
   (b) for $i = 1..N$,
      i. $w_i = w_i + (x_i - \mathbf{nearmiss}(x)_i)^2 - \mathbf{nearhit}(x)_i)^2$, where $\mathbf{nearhit}(x)$ and $\mathbf{nearmiss}(x)$ denote the nearest (Euclidean) point to x with the same or different label, respectively.

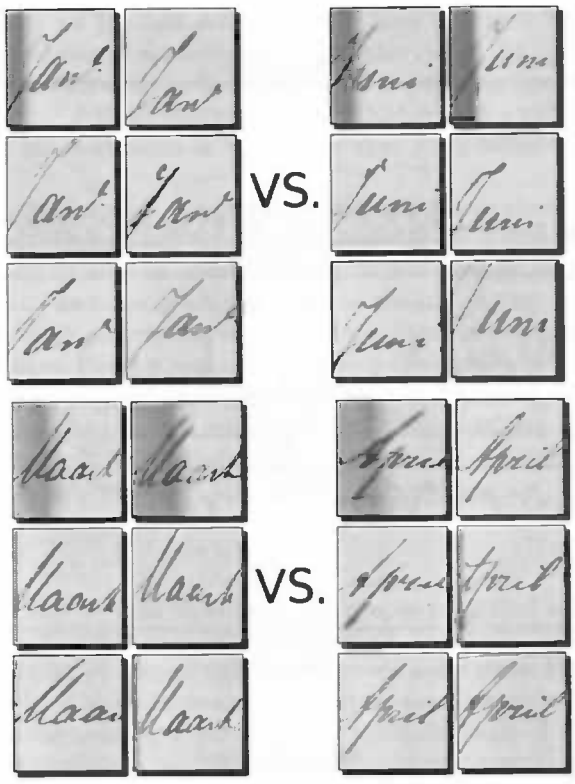


Figure 3.2: The dichotomies of *Jan* vs. *Jun* (top) and *Maart* vs. *April* (bottom), presented as Bongard problems. The evolving gated neural networks try to tackle these problems by using the feature extraction method described in chapter 2.

3. The chosen feature set is $\{i|w_i > \tau\}$, where τ is a threshold

To obtain performance results a 1 nearest-neighbor classifier (described in chapter 2) is used. Since the feature selection one obtains with RELIEF is highly dependent on the order in which - and which of - the data is presented, the performances are assessed using 10-fold cross-validation. Figure 3.5 shows the results of feature selection with a parametrization of 120 iterations, 10 train-samples per class, 447 + 449 test-samples for *Jan* and *Jun*, respectively, and a threshold ranging from $-.2$ to .9.

Maart vs. *April*

The most salient feature of the graphs presented in figure 3.3 is the flawless performance from the onset of the evolutionary process. In each generation more and more individuals show similar characteristics to this initial ('lucky') guess. The performance of this individual is maintained while the feature selection decreases in size, resulting in merely one attribute (prototype patch) necessary for flawless classification. On average, the resulting gated neural networks have hidden layers with 3.66 hidden nodes. Of the resulting 100 individuals, 83

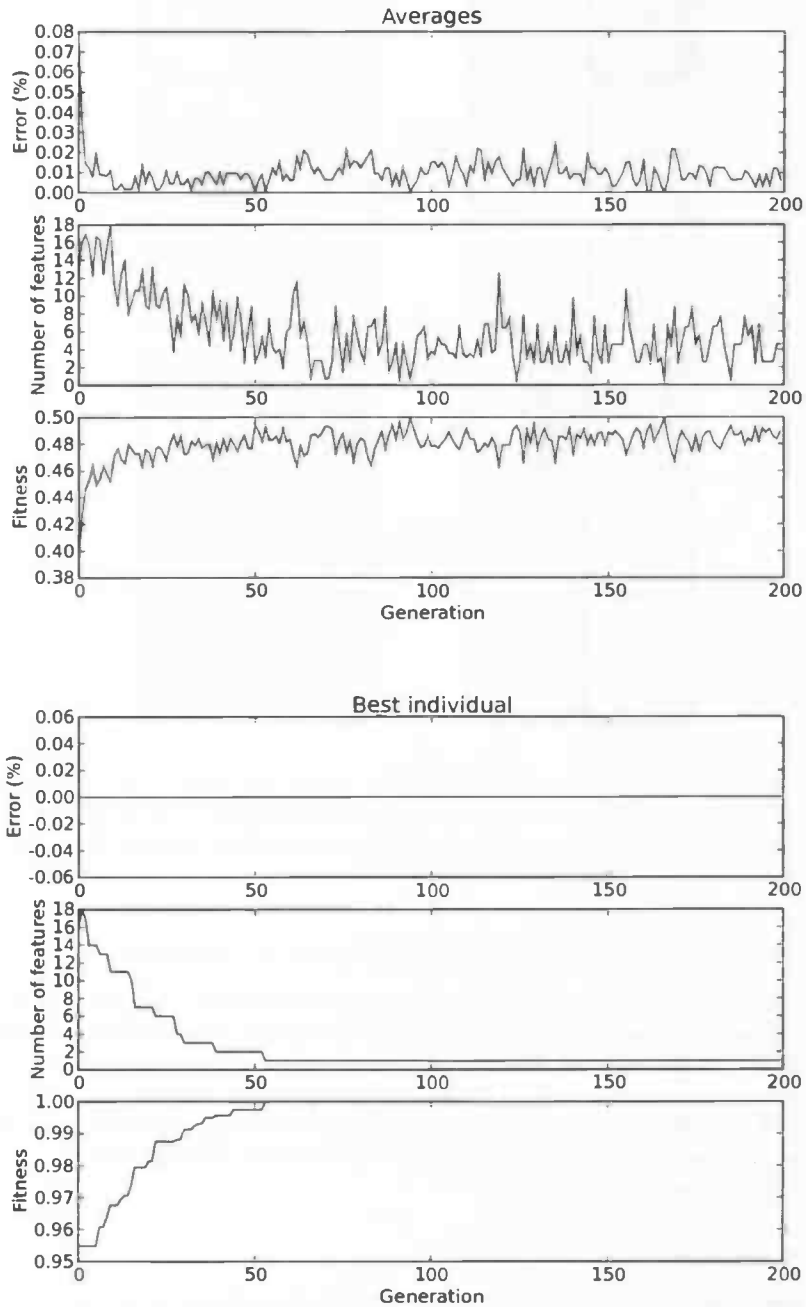


Figure 3.3: *Maart* vs. *April*. Evolutionary process of gated neural network tuning towards the problem of *Maart* vs. *April*. The top figure shows the average error, number of selected features, and fitness per generation. The bottom figure shows the smallest error and number of selected features, and best fitness for each variable separately for each generation.

perform flawlessly on the test set of which 6 have a hidden layer with only 1 node. These gated neural networks are thus linear, since the output nodes only transform the activation of the hidden node to the appropriate target value (o_1 responds with $r_{o_1} \geq 0$ if the activation of h_1 is higher than a certain value θ, and with $r_{o_1} < 0$ if the activation of h_1 is lower than θ; o_2 behaves like an inverse copy of 0_1).

The graphs that present the average error and fitness show a rapid adjustment to the first flawless individual, demonstrated by a rapid increase of the average error and fitness. A short while after, the average size of the feature selection starts to decrease gradually down to the size of one.

The evolutionary process has now tuned the gated neural networks for flawless performance on the test set, and concluded that, given its experience, only one prototypical patch is necessary for classification. Moreover, the evolutionary process found that the problem appears to be linear. Verifying the performance of the best classifier on the validation set again resulted in flawless classification.

Jan vs. *Jun*

Concerning the *Jan* vs. *Jun* dichotomy, the graphs presented in figure 3.4 show a slightly different course of events. First of all, the 'lucky' guess made when assessing the *Maart* vs. *April* dichotomy leaves some room for improvement, as can be seen by a slow decrease of the error-rate, although this might have been caused by the training-durations of 50 epochs instead of 100. Within 100 generations, the error-rate never reaches flawless performance, but still is near-perfect. The graph representing the best individual in each generation show that first the size of the feature selection is minimized, after which the performance can be increased considering certain extra features.

The evolutionary process results in 100 gated neural networks of which 50 perform with an accuracy of just above 99%. The average number of hidden nodes in the hidden layer is 3.36, and of the 50 best individuals, 4 use only one hidden node. Again, the problem is reduced to a linear one. Also with regard to this dichotomy, the result is flawless when the performance of the best classifier is verified on the validation set.

The resulting feature selection concerning the *Jan* vs. *Jun* dichotomy can be compared with another feature selection method: RELIEF. Figure 3.5 shows that although RELIEF can reduce size of the feature selection to approximately one, the performances have to suffer. This suggests that with correct parametrization, RELIEF did not find the right attribute that can render the classification problem linear, and performance near-perfect. This can be explained by the fact that RELIEF is a heuristic method of feature selection, while genetic feature selection is empirical. A major drawback of RELIEF is - as can be seen in figure 3.5 - that it takes careful tuning of the threshold (determining the amount of selected features) and many trials with different examples and orders of presentation to obtain a robust feature selection (the dots in the top graph of figure 3.5 show a relatively high variance between different trials). With the current parametrization, RELIEF obtains similar performance (be it with a 1 nearest-neighbor classifier) as genetic tuning, but still needs a feature selection of approximately 50 features. More iterations and more examples for training might improve feature selection with RELIEF. For now, genetic tuning

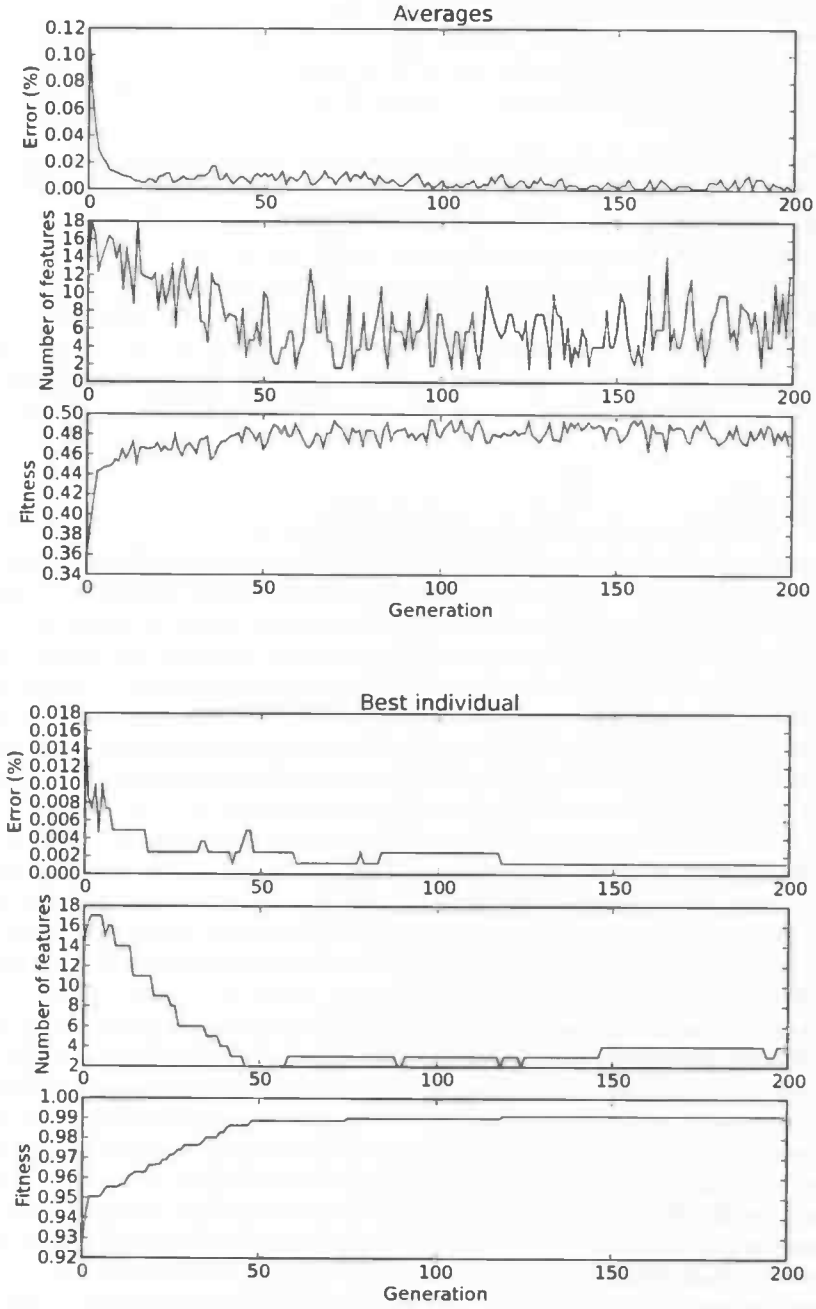


Figure 3.4: *Jan* vs. *Jun*. Evolutionary process of gated neural network tuning towards the problem of *Jan* vs. *Jun*. The top figure shows the average error, number of selected features, and fitness per generation. The bottom figure shows the smallest error and number of selected features, and best fitness for each variable separately for each generation.

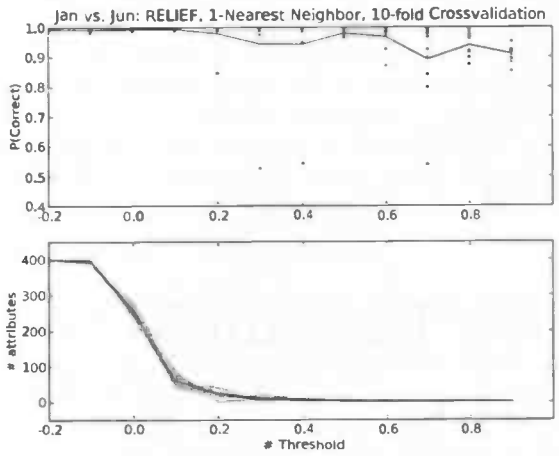


Figure 3.5: Feature selection using RELIEF on the *Jan* vs. *Jun* dichotomy with a threshold ranging from $-.2$ to .9. The top graph shows the average accuracy (black line) and each of the samples (dots). The bottom graph displays the number of attributes (patches) used for classification. The black curve shows the average number of attributes, the yellow lines represents the separate samples in the cross-validation

of gated neural networks appears to be a more appropriate, and additionally, more biologically plausible method.

Chapter 4

Adaptive Ensemble Generation

The previous chapters have demonstrated a biologically motivated approach to the steps of feature extraction, feature selection, and model choosing. The self-adaptive characteristic that each of these approaches show results in less parametrization, sparse coding and high domain-specific performance on handwritten texts. Chapter 2 demonstrated the power of the self-adaptive feature extraction method by the use of a 1 nearest-neighbor classifier and showed that the representations resulting from the extraction method were sufficiently discriminative to achieve high classification performance for each of the classes. Chapter 3 showed how genetic feature selection and model choosing can result in high classification performance using very sparse representation coding, but only for binary classification problems.

Nearest-neighbor classification can be implemented within connectionist terms. In order to perform such computations, a distance calculator is needed that allows for comparison between previously seen and new examples. This type of computation is often described in terms of multi-dimensional Gaussian functions (Radial Basis Functions) that compute the difference between two activation patterns, but how neural networks would calculate Euclidean distances has been an open question. A solution to this problem lies within the possibility to measure the similarity between two normalized vectors with both distance and the angle between the vectors (i.e., $|\mathbf{x}-\mathbf{w}|^2 = -2\mathrm{x}\mathrm{w}+1+|\mathrm{w}|$, if $|\mathbf{x}|=1$). This suggests that radial basis functions can be implemented by means of normalized scalar product operations. A combination of a divisive normalization network [32] and a scalar product network that normalizes afferents by division of the collective response of these afferents with inhibitory feedback connections [53] could therefore be used to perform nearest-neighbor classification.

A major problem with this speculated neural substrate of a classification mechanism is that this mechanism would need much too many resources: at least one such network per concept, even though decision making is reduced to merely determining which of the networks respond with the highest activation. A more likely view on a classification or recognition mechanism is described by the following metaphor presented in Douglas Hofstadter's book, *I am a strange loop* [35], slightly changed for present purposes:

Consider a chain of falling dominoes. In addition to this familiar notion, it is stipulated that each domino is spring-loaded in a clever fashion, so that whenever it gets knocked down by its neighbor, after a short 'refractory' period it flips back up to its vertical state, all set to be knocked down once more. With such a system it is possible to implement a mechanical computer that works by sending signals down stretches of dominos that can bifurcate of join together; thus signals can propagate in loops, jointly trigger other signals, and so forth. The basic idea is that one can imagine a network of precisely timed domino chains that amounts to a computer program for carrying out a particular computation.

Say this domino chainium can be used to recognize handwritten words. Using a particular representation of a handwritten word, a couple of dominos (constituting the input stretch) are tipped over and a series of events take place in which domino after domino falls over. As a consequence various loops will be triggered testing whether the input stretch is consistent with particular memorized features. The loops will maintain to be triggered if they are, otherwise they die off. Triggered loops can again trigger new loops testing for other consistencies and when a stretch can be seen falling it is known which consistencies are there and which are not.

Suppose an observer is standing by when the domino chainium is given a representation of *Jan* as an input. The observer, who has not been told what the chainium was made for, watches keenly for a while, and sees a growing pattern of falling dominos. When the pattern of falling dominos does not change anymore, he points to one of the dominos in the *Jan* consistency stretch and asks, "How come that domino here is never falling?". In contrast with a meaningless answer such as, "Because its predecessor never falls", the most reasonable answer would be, "Because the input-stretch represents *Jan*" and not any other.

In this view on classification or recognition, the input underlies a neural firing pattern that makes the brain 'aware' of *Jan* but also explains why researchers find 'grandmother'-cells, a side-effect of the overall pattern. The metaphor is important here because in addition, the overall pattern of falling dominos is analogous with problem solving within the divide and conquer paradigm, dividing a difficult problem into subproblems that are solved more easily. Moreover, the subproblems are ordered hierarchically, according to the inclusion of sets of classes into bigger sets of classes. Shortly after tipping the initial dominos over, general consistency tests are triggered, which later, by means of positive feedback, trigger more specific consistency tests until we can observe the pattern of falling dominos from which it can be derived that the pattern must be representing *Jan*.

This hierarchical divide and conquer type of inference can be referred to as hierarchical pattern rejection. Hierarchical pattern rejection has the advantage of allowing the use of the binary classifiers presented in the previous chapter, whereas other combination strategies of combining binary classifiers such as one-against-all combination would be less appropriate. Suppose the objective would be to recognize one out of thousands of words (a task humans usually perform when reading), then training a neural network with two possible responses (*Jan* or 'Other') would simply suffer from unevenly divided experience because it for instance already achieves low error-rates by simply always answering 'Other'.

Another advantage of the paradigm is that the strategy allows for hierarchical feature representations of the stimuli that need to be recognized, one for

each easily solvable subproblem that needs to be addressed. It is thus possible to employ a gated neural network, specifically tuned for the subproblem, on the task of discriminating between different sets of classes.

This chapter presents a classification experiment on handwritten month-words with tuned gated neural networks and the hierarchical pattern rejection strategy, whereas the approaches of the two previous chapters are exploited to complete the design cycle of pattern recognition. Although certainly at the level of classification self-adaptation can play an important role, it would have exceeded the scope of this thesis to add this characteristic here as well. Instead, the experiment presented in this chapter involves tuned gated neural network combination by hierarchical pattern rejection according to a more or less arbitrary predetermined decision tree. The further research section of Chapter 5 will provide some suggestions on how to add self-adaptivity to classifier combination, which should subserve a fully automated design-cycle of visual pattern recognition.

4.1 Hierarchical Pattern Rejection

Fritz and Finke [26] argue that a hierarchical approach for modularizing classification tasks is crucial in applying artificial neural networks in case of a high number of classes: With an increasing number of classes to be distinguished, neural network estimations of posterior probabilities (see section 1.1.4) fail to provide a good approximation mainly because of two reasons, of which the first is of most importance to this chapter. Real world problems, such as handwritten text recognition that involve a myriad of classes exhibit an extremely non-uniform distribution of priors [54] while many learning algorithms have problems with non-uniformly distributed classes. Particularly the distribution of infrequent classes are approximated poorly. The second reason involves the use of monolithic neural network classifiers, for which it is unfeasible to train them with a myriad of output-neurons. In addition, when applying a monolithic network, the number of possible conflicts grows when the number of classes increases.

The number of classes dealt with in this thesis certainly do not pose any problems of this kind, but it would not be generic to use monolithic neural networks if the number of classes is below an arbitrary threshold, and the divide and conquer paradigm if the number of classes exceeds the threshold. Thus for reasons of generality the paradigm is used also when the number of classes is relatively small. Moreover, in pattern recognition, any classification algorithm is capable to provide a binary output, whereas not all algorithms can cope with multiple classes without classifier combination.

4.1.1 Meta-classification

Hierarchical pattern rejection involves meta-classification, decisions regarding decreasingly smaller sets of classes. Pattern rejection operates with so-called rejectors instead of recognizers, which are employed to the easier task of eliminating some classes instead of all but one. Baker and Nayar [4] provide the following characteristics of pattern rejection wherein rejectors are used to eliminate most classes from further consideration:

1. The definition of correctness for a rejector is less constraining than that for a recognizer. As a result, rejectors can be constructed that are more efficient than recognizers.

2. Although in general a single rejector does not provide a final solution to the pattern recognition problem, it can significantly reduce the number of possible classes.

3. A collection of rejectors may be combined in a tree-like structure to form a more effective one, which is termed a *composite rejector*. Each node of the tree is a single rejector, tuned to the small set of classes which were not eliminated by a previous rejector.

4. It is possible to analyze the performance of composite rejectors. For instance, conditions can be derived that guarantee logarithmic time complexity in terms of the total number of classes involved.

With regard to the first characteristic, rejectors are less constraining than classifiers by their definitions: Assuming a classification space $S = \Re^d$, and a finite collection of classes $W_1, W_2, ..., W_n$ (subsets of S), a rejector is an algorithm ψ that, given an input $x \in S$, returns a set of class labels $\psi(x)$, such that $x \in W_i \Rightarrow i \in \psi(x)$. A classifier is an algorithm ϕ that, given an input $x \in S$, returns the class label i for which $x \in W_i$. This less constraining characteristic implies that rejectors allow the freedom to choose relatively simple decision boundaries, allowing the rejectors to be efficient [4].

Here a composite pattern rejector is used that can be represented by a directed acyclic graph (decision tree) for which each node ψ has only two possible outputs and the cardinality of these outputs is equal. In addition, the two outputs do not intersect. This leads to a worst case time complexity of $\log_2(n)+c_\psi$, where c_ψ corresponds to the time it takes to eliminate a set of classes from further consideration.

4.1.2 On the role of Prototypes

At each node of the composite pattern rejector, a specific feature-subspace needs to be identified together with a decision rule [72]. Since each node constitutes a decision task, the gated neural networks answer to this query by performing feature selection and provide a tuned decision rule after the networks are specialized in the node's task. This implies that for each decision a different sparse set of prototype-patches may be used to reject possible classes. Although patches do not appear in the size of characters, this property is nicely described by an initial rejection of all classes with J's, then all M's, etcetera, until for instance *Dec* remains. The results section shows that this characteristic is indeed beneficiary for recognition.

The above allows for some predictions concerning the number of patches gated neural networks at each level of tree use. At each node of the tree a balanced binary answer is needed and half of the classes need to be rejected. This suggests that ideally the tuned gated network uses those patches that are common in each of the two class-sets, and not the other. Additionally, the network will preferably use patches that do not occur in one of the two sets at all, but do occur in the other. Considering the level in the tree, finding such a feature-set is most likely easier when the node's task involves less classes.

4.1.3 Decision-committees

A major pitfall of using decision trees for classification is early commitment. Once a bad decision is made, there is no point attending nodes below the one that produced the erroneous decision since any decision of those nodes will be nonsense. It is therefore key that each node makes decisions with absolute certainty. Furthermore, assuming independence between the nodes (although faulty answers of predecessing nodes do affect the probability of a successing node's correct answer), the probabilities of answering correctly multiplies for each node in the path, quickly decreasing the probability of the final answer to be correct. Although a nonsense-input recognizer could provide feedback for the predecessing node, making it aware of its nonsense answer, this nonsense-input recognizer would functionally be exactly the same as the predecessing node.

Suppose the nonsense-input recognizer of level $L+1$ is somewhat different from the decision function of level L. For instance, the nonsense-input recognizer uses a different feature selection or different training sets [50]. Effectively, asking at a certain level L for feedback one level $L+1$ below from a nonsense-input recognizer would constitute the same as adding one decision-maker to level L, preferably a somewhat different one [1].

Feedback of this kind can therefore be added to the decision tree, without turning it into a directed cyclic graph, when at each node of the tree an ensemble of decision-makers is employed. Moreover, in the limit of adding 'different' decision-maker to the ensemble, the error-likelihood decreases towards zero [49], which is subserving avoidance of faulty early commitment and obviously increases the probability of a correct final answer.

4.2 Towards multi-class recognition

4.2.1 Methods

As the final stage of automating the design-cycle of pattern recognition with an adaptive object recognition approach, the tuned gated neural networks using self-selected features need to be combined to construct a full-fletched word recognizer. The approach here is set within the divide and conquer paradigm employing a composite pattern rejector in the form of a decision tree wherein each node consists of an ensemble of rejectors. The role of such a rejector is fulfilled by the tuned gated neural networks of chapter 3.

The decision tree used here is shown in figure 4.1, where can be seen that the tree consists of four levels, henceforth referred to as level *6 vs. 6*, *3 vs. 3*, *2 vs. 1*, and *1 vs. 1*, from top to bottom. Each node decides the unknown pattern to belong to one of two arbitrary sets of classes, whereas these sets are determined considering solely the number of train samples available for the class-set, where in turn these numbers had to be approximately equal for each of the two class-sets.

Each node involves an ensemble of 100 tuned gated neural networks, obtained from complete population in the 100th generation of the evolutionary process. Since the genetic algorithm described in chapter 3 uses an elitist selection procedure, the algorithm itself will not ensure that the resulting gated neural networks are indeed different considering their feature selections; as the

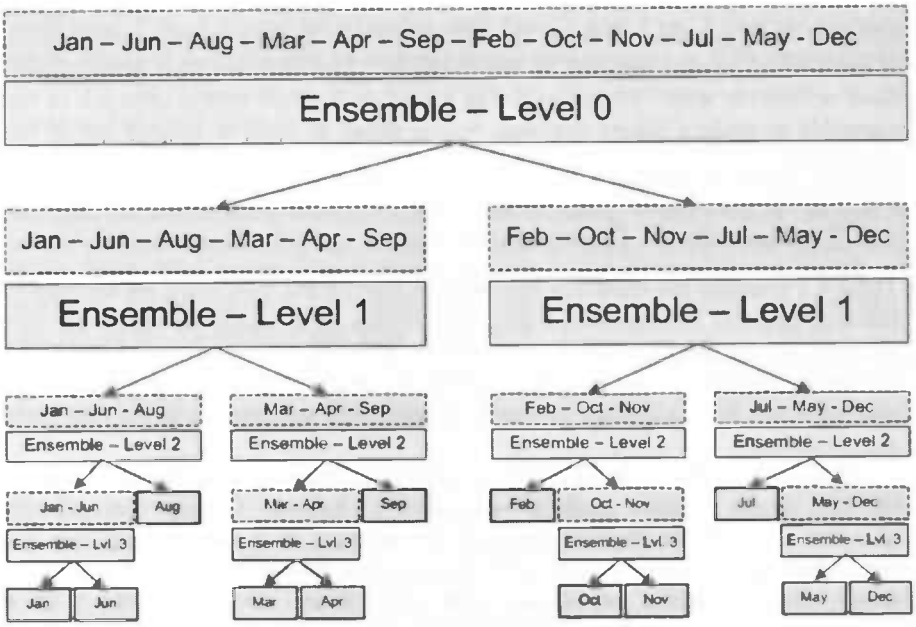


Figure 4.1: Decision tree for handwritten month-words classification. At each level an ensemble of evolved gated neural networks is employed to decide which meta-class to reject. The classes are divided over the meta-classes so that each of the two meta-classes at each level contains a similar amount of examples for training and testing in the evolutionary process.

individuals in each population converge towards an optimal solution, the individuals become more and more alike. Therefore the 100th population instead of a higher numbered population is chosen.

The number of training-epochs with regard to the gated neural networks is set to 50 for all nodes, except from the root level (100 epochs) which deals with the most difficult decision. For this reason, also the initial random feature selection concerning this node is set to 50 features, whereas for the rest of the nodes evolutionary ensemble generation is initiated with 25 selected features. A balance between computational load and good performance determined these numbers. More epochs, would lead to very long training times without improving the performance much further.

The dataset used here is the handwritten month-words dataset also used in the second and third chapter. The train set used for training the gated neural networks consisted of 50 examples per class, randomly sampled before the first generation is initiated. The number of test samples is described by the rightmost column of table 2.4, and the validation set consists of 40 examples per class, kept apart from any of the previously described experiments.

After constructing the decision tree, the validation set is used to assess its performance. Each node comes to a decision by means of a combination-rule, which is varied. The combination-rules used in this experiment are max-activation ($\sum \alpha$), majority voting (MV), and using the opinion of the gated neural network with the best error-rate on the test-set (Best). Max-activation

adds for each possible class-set the activation of the appropriate output node of each gated neural network in the example. The class-set that accumulated most activation wins. Majority voting forces each gated neural network in the ensemble to make a binary decision, voting either in favor or against one of the class-sets.

4.2.2 Results & Discussion

Table 4.1 presents the results of the decision tree on the validation set for each of the tree possible combination-rules. The results look disappointing, especially considering the flawless classification of the relatively simple dichotomies of the previous chapters. It appears that indeed the main pitfall of a decision tree is playing a role here, although precautions have been undertaken to prevent early commitment.

To elucidate the origin of errors made by the decision tree, figure 4.2 shows the first wrong decision made in each level of the tree. Clearly, the majority of errors are made in the first decision. In addition, it is also striking, but not surprising, that the amount of errors decreases when the sub-problems become easier, that is, concern less classes. The gated neural networks in the *6 vs. 6* node either should have been trained better, should have been assigned an easier dichotomy (possibly not one with balanced class-sets), or should have evolved for a longer time.

With regard to the features used in each level of the tree, the following can be said. At level *6 vs. 6* the gated neural network with the highest fitness uses twelve features and at level *3 vs. 3* the gated neural network with the best fitness uses nine. One level lower, at *2 vs. 1* either two or one feature is used, whereas all fittest gated neural networks of the *1 vs. 1* level merely use one feature. Surprisingly, figure 4.3 reveals that at level *6 vs. 6* there is almost no diversity among the individuals with regard to the feature selection. That is, the best features to discern this dichotomy are almost always used ($P(open) = 1$). Figure 4.4 indicates that longer training would have been beneficial since the performance which was near-perfect in early generations, but later on suffers from selective pressure to decrease the amount of features. The other levels also suffer small diversity;

Table 4.1: Tuned GNN Decision Tree: Months

Class	MV	$\sum \alpha$	Best
Jan	0.950	0.950	0.925
Feb	1.000	1.000	1.000
Mar	0.775	0.775	0.800
Apr	0.900	0.900	0.875
May	0.825	0.825	0.825
Jun	1.000	1.000	1.000
Jul	0.875	0.875	0.900
Aug	0.825	0.825	0.800
Sep	0.925	0.925	0.925
Oct	0.975	0.975	1.000
Nov	1.000	1.000	0.950
Dec	0.925	0.925	0.925
Total:	0.915	0.915	0.910

since level *3 vs. 3* contains two nodes, and *2 vs. 1* and *1 vs. 1* contain 4 nodes, small diversity corresponds to $p(open) = 0.5$ and $p(open) = 0.25$ respectively. Probably the ensembles at each node were not diverse enough to avoid wrong turns in the path from root to leaf.

Figure 4.3 shows an additional interesting result. Different features are needed to perform different meta-classification tasks. That is, which features are informative depends strongly on the task at hand, which level of the tree is attended, and, considering the maximal $p(open)$-values, which node of the tree is attended. This result is very intuitive, meaning that when starting with an unknown visual stimulus, the stimulus is recognized by focusing on particular patches of the visual field one after another.

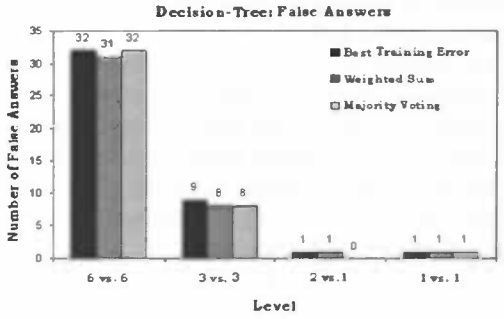


Figure 4.2: First false answers in each level of the decision tree for each of the three combination rules.

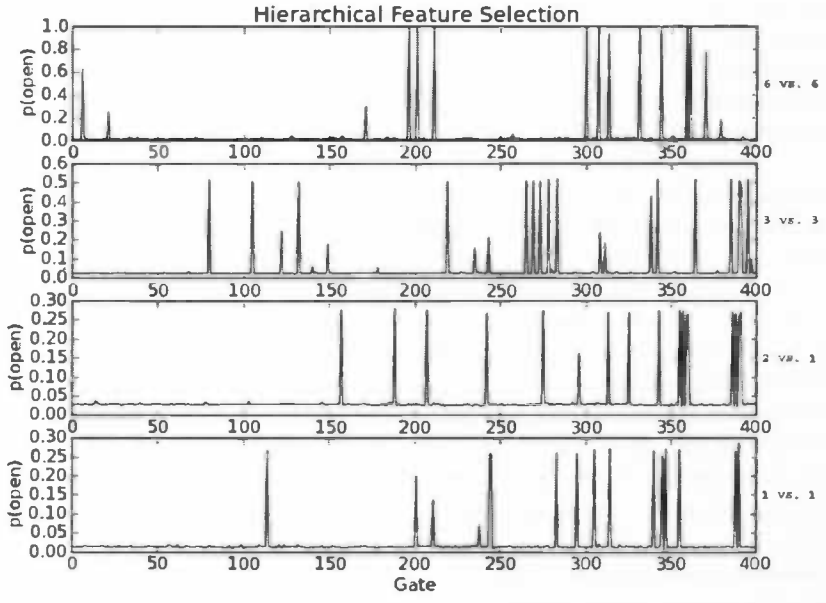


Figure 4.3: Hierarchical feature selection. The horizontal axis represents the initial feature vector, the vertical axis represents the probability the a feature is used (gate open) for each level in the decision tree. Values are obtained by counting which features are used by the collective of ensembles at each level of the tree.

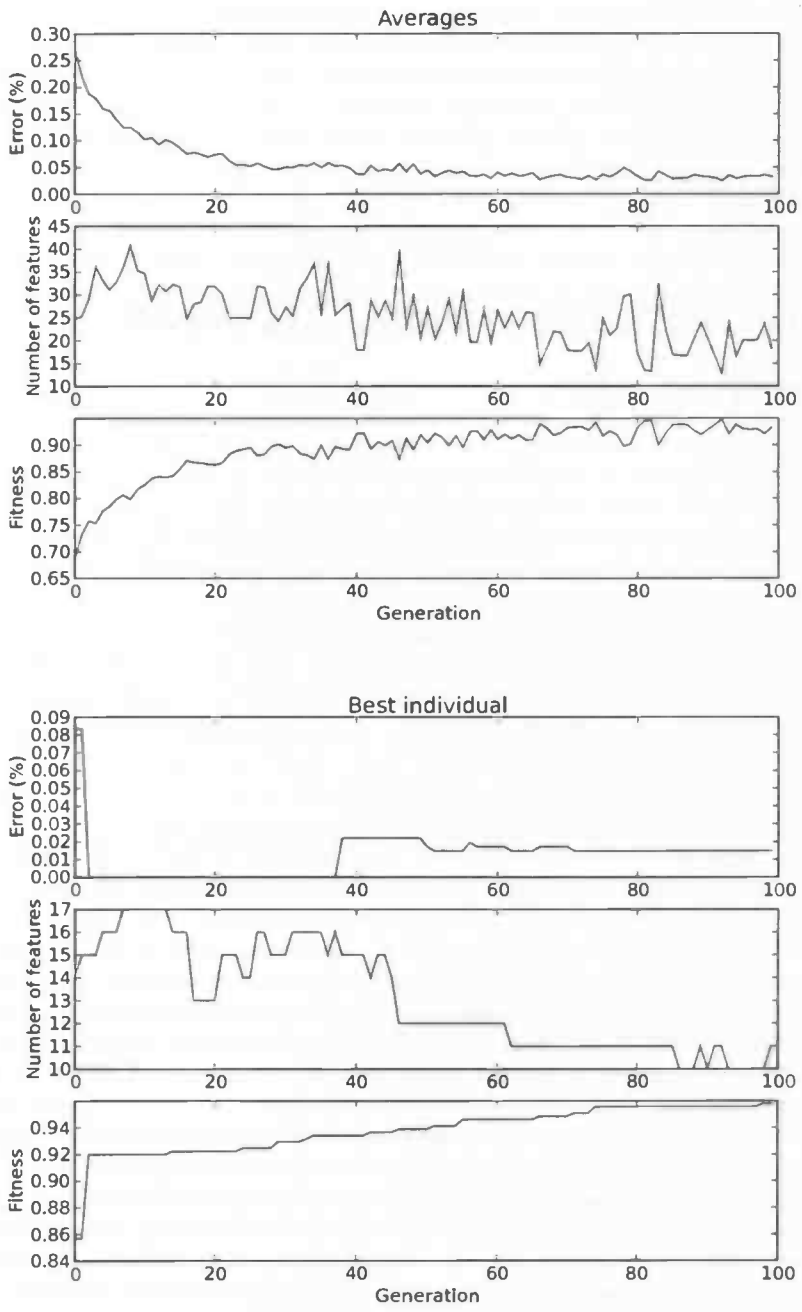


Figure 4.4: *Jan, Jun, Aug, Mar, Apr, Sep* vs. *Feb, Oct, Nov, Jul, May, Dec.* Evolutionary process of gated neural network tuning towards the problem of *6* vs. *6*. The top figure shows the average error, number of selected features, and fitness per generation. The bottom figure shows the smallest error and number of selected features, and best fitness for each variable separately for each generation.

Chapter 5

Discussion and Further Research

This thesis proposed a visual object recognition approach for handwritten text recognition based on three assumptions. First, neuroplasticity and neurogenesis hint to the existence of specialization mechanisms for a vast variety of visual recognition tasks, and it is assumed that also handwritten text recognition is a specialized form of object recognition. Secondly, it is assumed that each such specialized visual recognition mechanism is subserved by a single basic hierarchical visual processing mechanism, which allows specialization at higher levels of processing. Thirdly, specialization of such a basic processing mechanism for handwritten text recognition can, so it is assumed, be modeled by an evolutionary algorithm, since such an algorithm can mimic cortical competition.

The three assumptions provided the ingredients for a three-staged development of a handwritten word recognizer. Moreover, they subserve automation of design when addressing visually presented pattern recognition in general. Chapter 2 addressed the basic visual processing mechanism. Key characteristics of this mechanism are its self-adaptive behavior, its possible function as a feature extractor, and its biological plausibility. The hierarchical model of visual processing as proposed by Serre et al. [78, 79], which is based on earlier models of processing by Poggio et al. [73, 70] fulfills this role with surprising results. Originally designed for object recognition, the model provides an excellent feature extractor for handwritten text recognition as well, as chapter 2 demonstrates.

Supplied with a feature extractor, the common next step in developing a pattern recognizer is feature selection, a processing step that is also present in primate visual object recognition [67] where in V4 and beyond sparse population codings constitute object shapes. IT cells in primate brains tend to get tuned to particular stimuli with experience as well, resulting in so-called view-tuned (and later even view-invariant) cells, responding only to specific stimuli when a specific task is addressed [57]. In chapter 3, gated neural networks, tuned by a genetic algorithm, that integrate feature selection and network parametrization, mimic these phenomena while considering the assumption regarding cortical competition. Provided with an excellent representation resulting from the feature extractor, tuning resulted in an again excellent task-specific performance, not only reducing a 400-dimensional feature space to a couple of dimensions,

but additionally still performing flawlessly on unseen data.

However, unlike nearest-neighbor classification, which effortlessly discriminates between multiple classes, the resulting tuned gated neural networks of the third chapter do not allow for classification in such a setting without combining multiple classifiers. In order to maintain biological plausibility, and considering the limitations of artificial neural network learning algorithms, the chosen combination strategy that finds itself within the divide and conquer paradigm is hierarchical pattern rejection. Chapter 4 showed that this combination strategy provides an intuitive recognition inference, whereas the nature of the word is derived by focusing on general discriminative features first (literally parts of handwritten words), and on more specific parts of the text later. However, the pitfalls of using decision trees were not avoided, resulting in disappointing performances.

The following subsections will discuss these issues in more detail, additionally addressing questions for further research. Although the results of chapter 2 and 3 were excellent, this thesis merely presented exploratory research. Many improvements may be expected taking the discussed issues below into account. The disappointing results of the fourth chapter in comparison to chapters two and three are explainable (see the results & discussion section of chapter 4) and further experimenting may give room for improvement. However, classification or recognition in a setting of a myriad of classes is a very difficult problem, which should be addressed while closely watching nature, as has been done for visual processing. The combination strategy presented in the fourth chapter did not show any adaptive behavior except from its ingredients which certainly do show self-adaptation. A suggestion how to add such adaptivity will be presented as well.

5.1 Visual processing for handwritten text recognition

Employing the standard model of visual processing as a features extractor for handwritten texts is based on the assumption that basic visual processing subserves handwritten text recognition to a great extend. The key characteristic of such visual processing is the use of radial basis functions, in this case matching memorized patches of text with 'interesting' patches of text in particular areas of an image. Size and position do not have to be exact, and hence the visual processor produces a vector of match-values which is robust to within-class variance, but discriminative enough to elucidate between-class variability. Therefore the model complies to the criterion of stability and sensitivity regarding object representations as postulated by Marr [56] [1], resulting in the performances on handwritten texts reported in chapter 2.

The model used here takes the same parametrization as the one employed in visual object recognition tasks by Serre et al. [79], making sure indeed a general visual processing model is employed on the task of handwritten text recognition.

[1] Regarding handwritten texts the model complies to this criterion, but it does not for general visual processing in the view of Marr, since this criterion involves the debate surrounding view-point invariance, which is not achieved by the model as used here. However, the model performs consistent with findings of [55] regarding view-tuned neurons. Handwritten texts usually have only one view, hence the discussion of viewpoint-invariance is irrelevant here

Filters in four orientations and of particular sizes determine what is interesting in the image, and from intensity maps of training images, indicating these interesting parts, patches are sampled randomly. When these memorized patches are later matched with parts of a new image, further processing disregards information about the relative positions of these interesting parts (in contrast to Recognition by components [7]). Maintaining positional information is hypothesized to improve the results even further as well as changing the parametrization of the model.

In [83] it is postulated that experience also affects mechanisms of processing and that these mechanisms are task-specific. That is, visual processing procedure in case of handwritten text recognition has adapted itself towards this specific task. This feat can be achieved by a simple adjustment regarding selecting which patches apply as memorized patches. Instead of random patch-sampling, by modeling competition between memory connections, the matching procedure can be specialized. Starting of with an abundance of RBF cells with arbitrary memories as their centers, competition between these cells specializes the feature extraction method.

The standard model can be considered a part-based model but results in case of its use for handwritten text recognition in holistic recognition, rather than recognition by parts. Chapter 2 made it appear that capital characters and digits are harder to discriminate than words with the representation resulting from the visual processing model. Perhaps this was due to the origin of the examples, more likely it is due to the informational content of the stimuli. Possibly tuning the gated neural networks for digit and character recognition results in better performance. Perhaps relative positional information is more important for recognition of stimuli of this type.

Concerning word recognition, it is key to find the right parametrization (either manually or by means of self-adaptation) so higher level complex cells get tuned for parts of the word that resemble characters. If this could be achieved, the resulting feature representation after visual processing already codes the class-label itself.

5.2 Sparse representation of handwritten texts

Chapter 3 addressed the tuning of gated neural networks that integrate feature selection and classifier. This process resulted in very sparse representations: the dichotomies presented in this chapter can be discriminated with only one feature and a simple linear neural network.

Although modeling cortical competition with an genetic algorithm is justified, it might not been the best tool. Taking neuronal development and plasticity into account when developing a recognizer opens a whole new field of research, and more detailed knowledge about cortical competition mechanisms can eventually lead to the proposal of different tools such as competitive networks with biologically plausible network dynamics. As a proof of concept, the use of a genetic algorithm shows already high explanatory power, for instance as it allows to set up behavioral experiments for verification:

Suppose tuning was carried out in the context of for instance handwritten digit recognition. Additionally suppose that a one-against-all combination scheme is used, then tuning results in a total set of features so that each di-

chotomy in the combination scheme can be addressed. Most likely, this total set of features would not be big, and since the one-against-all scheme is used it is possible to reconstruct the (deteriorated) shape of the digits. This minimal shape representation of the digits can then for instance be compared with an behavioral study on degraded digit shape recognition.

5.3 Recognizing one out of a myriad of classes

The decision tree of chapter 4 finishes the handwritten word recognizer by enabling recognition of multiple classes. Unfortunately, the precautions of employing an ensemble at each node of the tree did not prevent faulty early commitment. As the sub-problems assessed by the tree become more simple, this does not make too much of a difference. The most difficult problem at the root of the tree can be considered the main bottle-neck. The ensemble in the root-node performs at best with 93.5% accuracy, which is then also the ceiling of the performance of the three. Chapter 4 concluded that this is due to a lack of diversity within the ensembles and the premature ending of the evolutionary process of the classifiers in the root node. These are problems that can be easily solved by extending the duration of the evolutionary process and concerning diversity employing bigger populations.

Another reason for the unsatisfactory results presented in the fourth chapter is that the dichotomies assessed by the tree here are completely arbitrary. Obviously, some dichotomies are easier to assess than others and most likely ample other dichotomies would have resulted in a better performance. But, as the number of classes would increase, the dichotomies close to the root of the tree will become more and more difficult to assess. A nicely balanced tree as used here therefore is probably not a possible combination strategy anymore.

Thus considering the parts of which it is built, the combination strategy should in theory recognize words well. However, the tree lacks an appropriate dichotomy choice and structure. In line with the adaptive approach explained in chapter 3, structural plasticity can address these two issues. The resulting recognition system would then with increasing experience first generate an abundance of gated neural networks. That is, when the recognizer encounters a dichotomy, specialization takes place to assess this dichotomy. Meanwhile connections between the gated neural networks grow or die off, again with experience. Possibly evolutionary programming can model this process of evolving decision trees, pooling from a repository of gated neural networks as building blocks.

5.4 Sparse Data

The adaptive approaches, and especially the use of evolving neural networks, need many examples to form a specialized recognizer. Unfortunately, data is always sparse and a solution for this problem needs to be found. Watching a page of handwritten words, or a pedestrian crossing the road, one can wonder how many, almost similar 'static' images are seen. That is, the question of how much experience we obtain in the real world is of the essence, and answering it could be a helpful indication on how to extract data. Currently, real-time visual pro-

cessing together with learning and recognizing are beyond the possibilities of the field of artificial intelligence, but mimicking this real-time learning to recognize behavior off-line (for instance, by mimicking reading eye-movements and record the input with a camera, or constantly slightly transforming already obtained data) might provide the necessary data to employ the proposed approaches and allow a handwritten word recognizer to develop itself.

5.5 Conclusion

The results of the experiments provide a proof of concept for the adaptive visual object recognition approach for handwritten text recognition. A great deal of research in the fields of visual processing and perception offers ample findings that allow for extension and improvement of the proposed approach, which in this exploratory research is already supported by surprising results. The approach differs from common handwritten text recognition techniques in that no attempt is made to manually design an 'adult' recognizer and that the approach is strongly motivated by neuroscientific findings, while it can still be fitted in the design framework which has proved itself to be extremely useful. Further research should involve the neural correlates of a developing recognizer, more specifically modeling cortical competition, and the neural correlates of visually presented handwritten word processing. In addition, a major outstanding question of concepts are represented needs to be addressed, since a main objective should be the recognition of a myriad of classes in a fast and efficient manner. Lastly, it is necessary to investigate how humans gather sufficient data for learning how to recognize and specialize their recognition abilities. These are all majorly difficult and important research questions, but new findings concerning these issues accumulate rapidly. In addition, it has been relatively easy to obtain satisfactory results without addressing these major issues, and new neuroscientific evidence surely boosts the performance of future handwritten text recognizers when approaching the problem within the paradigm proposed here.

Bibliography

[1] K.M. Ali and M.J. Pazzani. On the link between error correlation and error reduction in decision tree ensembles, 1995. Technical report 95-38, ICS-UCI.

[2] H. Almuallim and T.G. Dietterich. Learning with many irrelevant features. In *AAAI*, pages 547–552, 1991.

[3] Atkinson, J. et al. Contrast sensitivity of the human infant for moving and static patterns. *Vision Research*, 17:1045–1047, 1977.

[4] S. Baker and S.K. Nayar. A theory of pattern recognition, 1995. Columbia university technical report, CUCS-013-95.

[5] R.E. Bellman. *Adaptive Control Processes.* Princeton University Press, 1961.

[6] J.O. Berger. *Statistical Decision Theory and Bayesian Analysis.* Springer-Verlag, New York, second edition, 1985.

[7] Biederman, I. et al. Recognition by components: A theory of human image understanding. *Psychological Review*, 94(2):115–147, 1987.

[8] C.M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, Inc., New York, NY, USA, 1995.

[9] A. et al. Blumer. Occam's razor. *Inf. Process. Lett.*, 24(6):377–380, 1987.

[10] M. Bongard. *Pattern Recognition.* Hayden Book Company, 1970.

[11] Braun, C. et al. Dynamic organization of the somatosensory cortex induced by motor activity. *Brain*, 124:2259–2267, 2001.

[12] L. Brothers and B. Ring. A neurethological framework for the representation of minds. *Journal of Cognitive Neuroscience*, 4(2):107–118, 1992.

[13] J.P. Changeux, P. Courrege, and A. Danchin. A theory of epigenesis of neural networks by selective stabilization of synapses. In *Proceedings of the National Acadamy of Sciences USA*, volume 70, pages 2974–2978, 1973.

[14] J.P. Changeux and A. Danchin. Selective stabilization of developing synapses as a mechanism for the specification of neural networks. *Nature*, 264:705–712, 1976.

[15] F Coulmas. *The Blackwell encyclopedia of writing systems.* Oxford Blackwell, 1996.

[16] B.V. Dasarathy, editor. *Nearest neighbor patter classification techniques.* IEEE, 1990.

[17] Vandewalle J. Dellaert, F. Automatic design of cellular neural networks by means of genetic algorithms: Finding a feature detector. *Proceeding of the third IEEE International workshop on Cellular Neural Networks and Their Applications*, pages 189–194, 1994.

[18] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society (Series B)*, 39(1):1–38, 1977.

[19] R. DeValois, D. Albrecht, and L. Thorell. Spatial frequency selectivity of cells in macaque visual cortex. *Vision Research*, 22:545–559, 1982.

[20] R. DeValois, E. Yund, and N. Hepler. The orientation and direction selectivity of cells in macaque visual cortex. *Vision Research*, 22:531–544, 1982.

[21] Lowe D.G. Object recognition from local scale-invariant features. *Proceeding of the International conference on computer vision*, pages 1150–1157, 1999.

[22] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis.* Wiley, 1973.

[23] G.M. Edelman. *Neural Darwinism: The theory of neuronal group selection.* New York: Basic Books, 1987.

[24] G.M. Edelman. *Topology: an introduction to molecular embryology.* New York: Basic Books, 1988.

[25] Elman, J. et al. *Rethinking Innateness: A connectionist perspective on development.* Bradford, MIT Press, Cambridge, Massachusetts, 1996.

[26] J. Fritz and M. Finke. Applying divide and conquer to large scale pattern recognition tasks. *Lecture Notes in Computer Science*, 1524:315–342, 1998.

[27] D. Gabor. Theory of communication. *Journal IEE*, 93:429–459, 1946.

[28] Gauthier, I. et al. Expertise for cars and birds recruits brain areas involved in face recognition. *Nature Neuroscience*, 3(2):192–197, 2000.

[29] M.A. Goodale and A.D Milner. Seperate visual pathways for perception and action. *Trends in Neuroscience*, 15:20–25, 1992.

[30] R.M. Gray. *Source Coding Theory.* Kluwer academic publishers, Norwell, MA, USA, 1989.

[31] Whitley D. Guerra-Salcedo, C. Genetic approach to feature selection for ensemble creation. *Proceedings of the Genetic and Evolutionary Computation Conference*, 1:236–243, 1999.

[32] D. Heeger. Modelling simple-cell direction selectivity with normalized, half squared, linear operators. *Journal of Neurophysiology*, 70:1885–1898, 1993.

[33] Heisele, B. et al. Categorization by learning and combining object parts. pages 1239–1245, 2001.

[34] R. Held. *Early visual development: Normal and Abnormal*, chapter The stages in the development of binocular vision and eye alignment. Cambridge, MA: MIT press., 1993.

[35] D. Hofstadter. *I am a strange loop.* Basic Books, 2007.

[36] J. Holland. *Adaptation in Natural and Artificial Systems.* The University of Michigan Press, Ann Arbor, 1975.

[37] D.H. Hubel and T.N. Wiesel. Receptive fields, binocular interaction, and functional architecture of the cat's visual cortex. *Journal of Psychology*, 160:106–154, 1962.

[38] P.J. Huber. Projection pursuit. *Ann. of Statist.*, 13(2):435–475, 1985.

[39] A. Hyvrinen. Survey on independent component analysis, 1999.

[40] A.K. Jain, R.P.W. Duin, and J. Mao. Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000.

[41] A.K. Jain and D. Zongker. Feature selection: Evaluation, application and a small sample performance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:153–158, 1997.

[42] G.H. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *International Conference on Machine Learning*, pages 121–129, 1994.

[43] M.H Johnson and J. Morton. *Biology and Cognitive Development: The Case of Face Recognition.* Oxford: Blackwell, 1991.

[44] I. T. Jolliffe. *Principal component analysis.* Springer Verlag, New York, 1986.

[45] J.P. Jones and L.A. Palmer. An evaluation of the two dimensional gabor filter model of simple receptive fields in cat striate cortex. *Journal of Neurophysiology*, 58:1233–1258, 1987.

[46] J.W. Kalat. *Biological Psychology.* Wadsworth, 2001.

[47] K. Kira and L.A. Rendell. The feature selection problem: Traditional methods and a new algorithm. In *AAAI*, pages 129–134, 1992.

[48] K. Kira and L.A. Rendell. A practical approach to feature selection. *Assorted Conferences and Workshops*, pages 249–256, 1992.

[49] J. Kittler. Improving recognition rates by classifier combination: a theoretical framework. *Frontiers of handwriting recognition*, (5):231–247, 1997.

[50] Kittler, J. et al. On combining classifiers. *IEEE transactions on pattern analysis and machine intelligence*, 20(3):226–239, 1998.

[51] T. Kohonen. *Associative memory: a system theoretic approach.* Springer-Verlag, Berlin, 1978.

[52] I. Kononenko. Estimating attributes: Analysis and extensions of RELIEF. *Lecture Notes in Computer Science*, 784:171–182, 1994.

[53] M Kouh and T. Poggio. A general mechanism for tuning: Gain control circuits and synapses underly tuning of cortical neurons, 2004. AI Memo 2004-031/CBCL Memo 245, Massachusetts Institute of Technology, Cambridge.

[54] S. Lawrence, I. Burns, and A. Back. Neural network classification and prior class probabilities. In *Neural Networks: Trick of the Trade*, pages 299–313, 1998.

[55] N.K. Logothetis and J. Pauls. Psychophysical and psychological evidence for viewer centered object representations in the primate. *Cerebral Cortex*, 5(3):270–288, 1995.

[56] D. Marr. *Vision.* New York: Freeman, 1982.

[57] A. Martin. The representation of object concepts in the brain. *Annual review of Psychology*, 58:25–45, 2007.

[58] B.D. McCandliss, L. Cohen, and S. Deheane. The visual word form area: experise for reading in the fusiform gyrus. *Trend in cognitive sciences*, 7(7):293–299, 2003.

[59] G.J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition.* Wiley, New York, 1992.

[60] R.E. Michod. Darwinian selection in the brain. *Evolution*, 43:694–696, 1989.

[61] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *Proceedings of the international conference on computer vision and pattern recognition*, 2:257–263, 2003.

[62] A. Mohan, C. Papageorgiou, and T. Poggio. Example-based object detection in images by components. *IEEE Transactions on Pattern analysis and Machine Intelligence*, 23:349–361, 2001.

[63] Oliveira et al. Feature selection for ensembles: A hierarchical multi-objective genetic algorithm approach. *Proceedings of the 7th International Conference on Document Analysis and Recognition*, pages 676–680, 2003.

[64] N. Otsu. A threshold selection method from gray level histograms. *IEEE Trans. Systems, Man and Cybernetics*, 9:62–66, 1979.

[65] S.L. Pallas. Intrinsic and extrinsic factors that shape neocortical specification. *Trend in Neurosciences*, 24(7):417–423, 2001.

[66] A. Pasupathy and C.E. Connor. Population coding of shape in area v4. *Nature Neuroscience*, 5(12):1332–1338, 2002.

[67] J.J. Peissig and M.J. Tarr. Visual object recognition: Do we know more now than we did 20 years ago? *Annual Review of Psychology*, 58:75–96, 2007.

[68] Peterson, M.A. et al. Object memory effects on figure assignment: Conscious object recognition is not necessary or sufficient. *Vision Research*, 40:1549–1567, 2000.

[69] T. Poggio and E. Bizzi. Generalization in vision and motor control. *Nature*, 431:768–774, 2004.

[70] T. Poggio and S. Edelman. A network that learns to recognize three-dimensional objects. *Nature*, 343:263–266, 1990.

[71] M.J.D. Powell. Radial basis functions for multivariable interpolation: A review. In *Algorithms for Approximation*, pages 143–167, Oxford, 1987. Clarendon Press.

[72] A.F.R. Rahman and M.C. Fairhurst. Multiple classifier decision combination strategies for character recognition: a review. *International Journal on Document Analysis and Recognition*, 5:166–194, 2003.

[73] M. Riesenuber and Poggio T. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025, 1999.

[74] J. Rissanen. Stochastic complexity and modeling. *Annals of Statistics*, 14:1080–1100, 1986.

[75] D. Rumelhart and J. McLelland. *Parallel Distributed Processing*. MIT Press, 1986.

[76] P.H. Schiller, B.L. Finlay, and S.F. Volman. Quantative studies of single cell properties in monkey striate cortex ii. orientation selectivity and ocular dominance. *Journal of Neurophysiology*, 39(6):1334–1351, 1976.

[77] P.H. Schiller, B.L. Finlay, and S.F. Volman. Quantative studies of single cell properties in monkey striate cortex iii. spatial frequency. *Journal of Neurophysiology*, 39(6):1334–1351, 1976.

[78] Serre, T. et al. A theory of object recognition: Computations and circuits in the feedforward path of the visual stream in primate visual cortex, 2005. AI Memo 2005-036/CBCL Memo 259, Massachusetts Institute of Technology, Cambridge.

[79] Serre, T. et al. Robust object recognition with cortex-like mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):411–426, 2007.

[80] C.J. Shatz. The developing brain. *Scientific American*, 267(9):60–67, 1992.

[81] A. Slater. The visual constancies in early infancy. *The Irish Journal of Psychology*, 13(4):411–424, 1992.

[82] M.J. Tarr and Y.D. Cheng. Learning to see faces and objects. *Trends in Cognitive Sciences*, 7(1):23–30, 2003.

[83] M. Turenhout, T. Ellmore, and A. Martin. Long-lasting cortical plasticity in the object naming system. *Nature*, 3(12), 2000.

[84] S. Ullman, M. Vidal-Naquet, and E. Sali. Visual features of intermediate complexity and their use in classification. *Nature Neuroscience*, 5(7):682–687, 2002.

[85] L.G. Ungerleider and M. Mishkin. *The analysis of visual behavior*, chapter Two Cortical Visual Systems. Cambridge, MA: MIT press., 1982.

[86] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[87] G. Voronoi. Nouvelles applications des paramtres continus la thorie des formes quadratiques. *Journal fur die Reine und Angewandte Mathematik*, 133:97–178, 1907.

[88] C.S. Wallace and P.R. Freeman. Estimation and inference by compact coding. *J. R. Statist. Soc B*, 49(3):240–265, 1987.

[89] M. Weber, W. Welling, and P. Perona. Unsupervised learning if models of recognition. In *European conference of Computer vision*, pages 1001–1108. Massachussetts Institute of Technology, 2000.

[90] X. Yao. Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9):1423–1447, 1999.

[91] Yoshida, H. et al. Orientation selectivity is present in the first month and subsequently sharpens. *Investigative Ophthalmology & Visual Science supplement*, 31:8, 1990.