

Handwritten Digit Classification Using Cascading Neural Network  
Ensembles

Nancy Zaramian

**A Thesis**

**In**

**The Department**

**Of**

**Computer Science**

**And**

**Software Engineering**

Presented In Partial Fulfillment of the Requirements  
For the Degree Of Master Of Computer Science at  
Concordia University  
Montréal, Québec, Canada

March 2006

© Nancy Zaramian, 2006



Library and  
Archives Canada

Bibliothèque et  
Archives Canada

Published Heritage  
Branch

Direction du  
Patrimoine de l'édition

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file* *Votre référence*

*ISBN: 0-494-14341-X*

*Our file* *Notre référence*

*ISBN: 0-494-14341-X*

#### NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

#### AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

---

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

  
**Canada**

# ABSTRACT

Handwritten Digit Classification Using Cascading Neural Network Ensembles

Nancy Zaramian

In the problem of handwritten digit classification, difficulties are encountered when there is ambiguity among the digits to be classified. It is desirable to detect this confusion and either reject the classification or attempt to make a better decision using post-processing methods. In the proposed method, a cascading neural network model is used to do the latter. Each level contains an ensemble of neural networks trained on different features. This generates classifiers that complement each other and help identify samples that are difficult to classify. Experiments were done on the MNIST database. This database has 60000 training images and 10000 test images that contain segmented handwritten digits. The results from the experiment show an improvement in the classification accuracy with the addition of each level of neural networks. Out of the 10000 test images 2206 of the samples were rejected from the cascading neural network model and were sent to post-processing. Among the 7794 of the accepted samples not sent to post-processing, only 52 were falsely classified. The overall classification rate of the system, including post-processing is 96.58%.



# Contents

<b>List Of Figures .....</b>	<b>vii</b>
<b>List Of Tables.....</b>	<b>viii</b>
<b>1. Introduction.....</b>	<b>1</b>
1.1 Problem Description.....	1
1.2 The Importance of Handwritten Digit Recognition .....	3
1.3 System Overview .....	4
1.4 Thesis Organization.....	6
<b>2. Background .....</b>	<b>7</b>
2.1 Classifiers .....	8
2.2 Cross-validation .....	14
2.3 Confidence Scoring.....	15
2.4 Ensembles.....	16
2.5 Combination Strategies .....	18
2.6 Ensemble Methods .....	20
<b>3. Proposed Method.....</b>	<b>23</b>
3.1 Training .....	24
3.2 Classification .....	26
3.3 Generalization at each level .....	28
3.4 Speed .....	29

3.5 Complementing Classifiers .....	29
3.6 Ensemble as a Decision Tree .....	31
<b>4. Implementation .....</b>	<b>33</b>
4.2 Neural Network Structures.....	35
4.3 Neural Network Training .....	36
4.4 Ensemble Training .....	36
4.6 Classification .....	37
4.7 Post-Processing .....	37
<b>5. Experimental Results .....</b>	<b>39</b>
5.1 Training Databases.....	39
5.2 Testing.....	40
5.3 Classification Results .....	43
5.4 Speed .....	45
5.5 Image Samples .....	45
5.6 Types of Agreements and Disagreements.....	48
5.7 Additional Experiments .....	49
<b>6. Suggested Improvements .....</b>	<b>51</b>
<b>7. Conclusion.....</b>	<b>54</b>
<b>8. References .....</b>	<b>55</b>

## List Of Figures

Figure 1: OCR System Overview .....	4
Figure 2: Cascading Neural Network .....	24
Figure 3: Ensemble as a Decision Tree.....	32
Figure 4: Implemented Neural Network Ensemble .....	33
Figure 5: Largest Connected Components.....	34
Figure 6: Horizontal Distance Features of the digit 2.....	38
Figure 7: Images Accepted by Agreements with Very High Confidence .....	46
Figure 8: Images Rejected by all Levels.....	47

## List Of Tables

Table 1 Neural Network Structure .....	35
Table 2 Training Databases.....	39
Table 3 Accepted and Rejected Samples .....	40
Table 4 Correct and False Accepts .....	41
Table 5 Disagreements.....	42
Table 6 Agreements with Low Posterior Probability .....	42
Table 7 Classification Rates.....	44
Table 8 Classification, Rejection and Error Trade-offs.....	45
Table 9 Types of Agreements and Disagreements .....	48
Table 10 Results for Types of Disagreements .....	49
Table 11 Classification Results from Additional Experiments .....	50



# 1. Introduction

## 1.1 Problem Description

Handwritten digit recognition is a type of optical character recognition where we are asked to associate the correct digits to images that contain handwritten numerals. In this particular case, the focus is on off-line handwritten digit recognition where the data has already been captured and the digits have been segmented. Therefore, the problem consists of associating the correct digit class to the digit images given as input.

There are problems particular to handwritten digit recognition since there is no predefined constraint on the input. However, there are some implied constraints such as certain fields that must contain digits, and therefore there are only ten possible values. A limit on the number of sriptors and handwriting styles might also be some additional constraints. However, these are minimal and a good recognition system should be able to deal with different writing styles. Obviously, the fewer constraints there are the more difficult recognition becomes. Dealing with different writing styles is an important aspect of achieving a successful system.

Due to the very limited constraints imposed on handwritten characters, learning machines are usually used to build the recognition systems. A learning machine computes a function that allows the mapping of pattern inputs to their corresponding categories. A set of patterns, called the training set, which is representative of future inputs, is used during the learning process. The samples in the training set are described by a set of values, called the feature vector, which represent the important attributes of the samples. The learning system uses the feature vectors of the training samples along with the labels associated to them to compute the function. Classification occurs when the function computed by the learning system is used to predict the class of a test pattern. To do so, the same feature vector is extracted to describe the test sample and is given to the function. The output is the predicted class.

Research has focused on the problem of handwritten digit classification with good results. However, the current technology does not perform as well as humans. As described in [13] there are three main sources of error in a classification problem: “Class ambiguity” arises when two samples from different classes have identical feature values. “Imperfectly modeled boundary” occurs when the classifier does not model the optimal decision boundary between the classes. That is, the classifier does not partition the feature vector representations of the samples into their corresponding classes in an optimal way. “Small sample

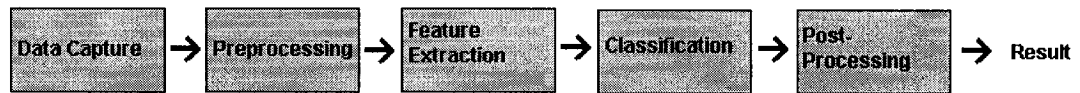
effect and feature space dimensionality” occurs when the training set for the classifier does not accurately represent test samples.

It would be useful to, as much as possible, automatically identify classification errors and be able to make the best possible decision on those samples. The method proposed in this paper deals with handwritten digit classification, focusing on the detection of samples that are difficult to classify and the improvement of the error rate by attempting to make a better decision on those samples.

## **1.2 The Importance of Handwritten Digit Recognition**

Systems with very high recognition rates allow for the automation of certain tasks. In particular, handwritten digit recognizers can be used for the processing of bank checks where it is required to read fields such as the courtesy amount and the date. Sorting mail by address and postal code and the automation of data entry are just some of the other types of applications. Therefore, handwritten digit recognition has a wide range of interesting applications. Since these applications concern millions of documents then it becomes of great economic interest.

### 1.3 System Overview



**Figure 1: OCR System Overview**

OCR systems generally follow a certain processing pattern. The first step is data capture where the source of the data is usually a document or an object containing characters. Information is captured either by a scanner or another method that digitizes the information. Since the quality of the data is important for the identification of characters, this step usually attempts to make the data as clear as possible. However, this is not always the case and often times distortions and artifacts can be introduced. To deal with this problem, the digitized image is then passed on to a preprocessing step.

Various types of preprocessing are applied depending on the data given to the OCR system. Filters are used to remove noise, enhance the features of an object or to simplify the image. The goal is to improve the quality of the data without losing critical information. For documents that contain many lines of characters the baseline for each one is detected. Depending on whether the system will recognize individual characters or complete words, the preprocessor will segment the data accordingly. The holistic approach attempts to recognize a

complete word at a time and is usually applied to cursive handwriting. The segmentation is based on the spacing between words and characters and often times this can be made difficult by inconsistent separations. Different fields are identified as being numerical or alphanumeric or both and this information is passed on to the following stage.

Following preprocessing is feature extraction. At this stage, the goal is to extract important information from different parts of the data later used for classification. Features should be chosen so that the data can be generalized properly. Interesting features for characters are number of corners, holes, endpoints, chain codes, pixel values and pixel distance features. Once the features have been chosen, different methods are applied to extract them. A feature vector is then used to represent the character.

During the classification phase we would like to associate the correct class to the character. In the case of digit recognition the classes are from 0 to 9. A function is applied to the feature vectors and the output is the class associated to the input. The classifier outputs the predicted class of the input and some classifiers also give a confidence value that will help determine the certainty of the classification.

Once a class is associated to an input, additional post-processing methods can be applied. Some systems apply contextual information to provide more data on the sample being classified. A measure of confidence can also be used to identify classifications with low certainty. If the previous steps of the system allow for the identification of misclassified characters then other methods can be applied to deal with the more difficult cases.

## **1.4 Thesis Organization**

In an attempt to deal with the complexities of handwritten digit recognition, the main focus of this thesis will be on the detection and classification of digits that are harder to classify. The goal is to improve classification accuracy by identifying samples that are confusing and to use post-processing methods to make a final decision on these samples. In chapter 2 I will give an overview of the current state of the art followed by a description of the proposed method in chapter 3. I will then discuss the implementation in chapter 4, experimental results in chapter 5, and give some suggestions on additional improvements in chapter 6.

## 2. Background

The general problem of classification consists of predicting the class of some case or object based on some measurements [3]. The goal is to find a classifier that accurately predicts the required class. It receives the measurements as input, and outputs the class to which the input belongs. Some output a set of scores each of which is a measurement of how likely it is the input belongs to that category. Generally, the one with the highest score is associated with the given input.

As mentioned in the introduction, dealing with different writing styles is important for a successful recognition system. Therefore, classifiers used for handwritten digit recognition usually have a learning stage. The idea behind using a method that “learns” comes from statistics. We gather as much data as necessary to represent future inputs then we attempt to find a function that generalizes the data without over-fitting. Over-fitting occurs when the function resulting from the learning process has not only learned the general characteristics of the training patterns, but has also learned information specific to the samples which are not necessary to the generalization. This can result in false predictions in test samples.

This section gives an overview of different classification methods currently applied to handwritten digit recognition. Experiments done in [16] give a comparison of the different classification rates. Additional background information will be given on neural networks and neural network ensembles and other works that relate directly to the research presented in this thesis.

## **2.1 Classifiers**

Classifiers applied to handwritten digit recognition range from very simple to more complex. Simple classifiers generally have the advantage of requiring less training time, however a compromise is usually made in the classification rate. The following gives an overview of the classifiers currently used in applications of handwritten digit recognition.

### **Linear Classifier**

A linear classifier, the simplest kind of learning method, uses a linear function on the given input to determine its class. Output units are determined by a weighted sum of the input feature values. The output unit with the highest value indicates the class of the input. Experiments done in [16] show it is the classifier that yields the highest error rate for handwritten digit recognition. The deficiencies of linear classifiers are documented in [8].



## **K-Nearest Neighbor Classifier**

Another simple method is the K-Nearest Neighbor Classifier [7]. Features are extracted from the training set and plotted in the multi-dimensional feature space. The learning stage consists of forming feature vectors of the objects in the training set and associating them to the class of the object. When classifying unknown samples the same features are extracted from the new samples, and the geometric distance is computed with all the prior feature vectors from the training set. The majority of the classification results of the K shortest distance is taken to be the class of the new sample. This method has the advantage that no training time is required.

## **Radial Basis Function Network**

The radial basis function network is a specific type of neural network [17]. It is made up of the input layer, one hidden layer and an output layer. The neurons in the hidden layer use nonlinear activation functions and the output layer uses linear activation functions. The idea is based on Cover's theorem on the separability of patterns [6]. It states that nonlinearly separable patterns can be separated linearly if the pattern is cast nonlinearly into a higher dimensional space.

The hidden neurons contain the radial basis function that is usually based on a Gaussian distribution. This function, which takes the center and width as parameters, has a peak when the input center is at zero distance from the center.

## **Convolutional Network**

The idea behind convolutional networks, introduced in [16], is to allow the neurons of the network to extract local features. Local features such as end-points and corners have been shown to be useful in identifying characters. To extract these local features the units in one layer receive inputs from a small neighborhood of units in the previous layer called local receptive fields. Once the local features have been extracted they are combined by subsequent layers and used to detect higher order features. Since the position of features can vary it is important to normalize the size and center the digits.

The elementary features extracted can be useful on another part of the image. Since units have receptive fields located at different places of the image, then assigning identical weight factors to a set of units allows the detection of the same features at different parts of the image. The back-propagation algorithm is used to learn the weights.

LeNet-5 described in [16] is a typical convolutional network for recognizing characters. Experiments have shown that convolutional networks yield very high recognition rates.

## **Support Vector Machines**

Support Vector Machines are a method for generating functions from the labeled training set [29]. The output for a classification function is binary. It determines whether the input belongs to a category or not. The method operates by finding a hyper surface in the space of possible inputs such that the distance from the hyper surface to the nearest positive and negative examples is the largest. The method achieves very high recognition rates, however the computation cost is very high.

## **Neural Networks**

A neural network is a learning system that is often used for classification problems. A network is composed of several layers of nodes, and connections between these nodes. The number of layers, the number of nodes in each layer, and the connections between the nodes, determine the structure of a neural network. Networks have an input layer and an output layer. Nodes receive a weighted sum of inputs and determine an output based on an activation function. Initially, the weights are chosen randomly and the goal of the learning method is to adjust these weights according to statistical data.

A supervised learning scheme is used to train the neural network. A database of sample patterns and their associated labels are given, and a training algorithm is used to modify the weights of the neural network until it has learned

the training sample set. The learning algorithm extracts relevant information from the database in order to properly generalize and correctly classify future input patterns that have similar properties to the samples already seen [18].

Fully connected feed-forward neural networks have been studied and used extensively. In this type of network, the connections from one layer to another go in one direction. That is the information is propagated from one level to another but never back to a previous level. Also, every node from one level is connected to every other node in the next level. That is the output from a node on a given level is propagated to every other node in the following level.

Back-propagation is a learning algorithm that is often used for this type of network [21]. On every iteration of the algorithm, the training samples are run through the network. For every sample, the output of the network is a vector of values. The algorithm computes the error between the target output and the output actually computed by the neural network. The error is then propagated upwards through the network and weights are modified to reduce the error. The algorithm iterates in this way until the output of the network on the training samples correspond to the target outputs. The algorithm attempts to minimize the number of errors over the training sample set. Therefore, it searches for a local minimum on the error surface of the training data.

## **Decision Trees**

Decision trees are also used for classification [3]. Like neural networks, they take as input a set of measurements of an object and output a decision. The decision output can be binary or a larger set of classes can be represented. The decisions are taken at each node and they are based on tests done on the values of the attributes. The leaf nodes specify the output of the decision tree.

The structure of the decision tree is not known until it is built. Building a decision tree consists of deriving a set of rules from the training data in order to classify them. The idea is to generalize as much as possible at each node so the class of the input can be narrowed down as quickly as possible. The process of deriving the classification rules from the samples is called discrimination, which is a way of partitioning the data into smaller subsets. The elements in the subsets are grouped together because they all share a certain property. The key to building a good decision tree is to determine the best discrimination at each node and unless the tree is pruned it learns 100% of the training data. Applying the rules to new objects of unknown class is classification.

Although decision trees can learn the complete training set, over-fitting can occur, in which case the classification accuracy on test samples can decrease. Therefore, the tree is often pruned and a compromise is made between the accuracy on the training data and the accuracy on the test data.

## 2.2 Cross-validation

In any learning algorithm there is a risk of over-fitting. That is, the classifier loses the ability to generalize properly by over-learning the training set, which can contain samples with noise. This occurs when the number of parameters of the classification function is more than what is required to generalize over that data. In this case, the classifier not only learns about the required features of a class but also learns about the features that are not necessary to associate a sample with the class it belongs to. For test samples, over-fitting can lead to unpredictable results.

The overall goal for a learning method is to minimize errors on test samples. Therefore, instead of using the classification rate on the training set, cross-validation can be used to determine the ability of the classifier to generalize. Cross-validation is a standard way of determining how much training is required to increase the classification rate on future samples [27]. To do so, the training database can be partitioned into specific training and validation sets. During the training of a classifier, a validation error can be computed on the validation set to determine whether to continue training. One known method called “early stopping” stops training when the error rate on the validation set starts to go up. One disadvantage to cross-validation is stopping training too early and therefore not finding a global minimum on the error surface. In this case, it would be possible to achieve a better recognition rate by continuing training, however since

we rely on the result of the validation, then we are “stuck” at a local minimum with a higher error rate.

### **2.3 Confidence Scoring**

Recognition systems often require automatic methods of detecting misclassifications. These systems must decide whether to accept or reject the result of a classification. Methods have been developed such as in [9] to optimize the error-reject trade off.

Some classifiers provide additional information on the classification result. They provide a measure on the certainty of the classification. Confidence scoring is a way of measuring how sure we are of the result obtained from the classifier. In a neural network that outputs values for each class, the maximum is usually taken as the class of the input. However, the scores associated to every other possible class can also give us information about how much we can rely on the classification. Therefore, many measures consider not only how sure we are of the hypothesized class but also how sure we are the input does not belong to another class. Many applications use the confidence score as a way to determine whether the result is accepted or rejected. Therefore, a good way of measuring the certainty of a classification is important for post-processing operations.

Confidence measures such as likelihood ratio and estimated posterior probability are discussed in [13]. Both of these measures make use of the scores associated with more than one class and give a confidence value on the best hypothesis. In the case of the posterior probability, once a hypothesis  $h_1$  is determined, the following formula is used to determine the probability of a valid classification:

$$P(h_1) = \text{score}(h_1) / \sum \text{score}(h_k), \quad k=1 \text{ to } \# \text{ of classes}$$

## 2.4 Ensembles

Ensembles of classifiers are a way of combining the classification of multiple experts. The classifiers in an ensemble can be the same type or different experts can be combined such as in hybrid systems. Experiments have shown that the combination of more than one classifier can yield better results than just one. There are many variations on the structure of an ensemble. Decisions such as the number and type of classifiers affect the increase in the classification accuracy. Once the structure of the ensemble has been chosen then the combination of the classifier outputs must also be chosen. More detailed descriptions of ensemble methods are given in section 2.6.



## Neural Network Ensembles

The basic idea of a neural network ensemble is to use more than one network to determine the class of an input. We obtain the classification result from each network in the ensemble and use a consensus scheme to decide the collective vote. A consensus scheme is when the final classification result is based on the agreement of the majority of the networks. The motivation for using more than one neural network for a classification problem is the possibility of error independence among the networks. This occurs when neural networks in an ensemble do not misclassify the same samples. They can complement each other by making different classification errors.

As mentioned earlier, training a neural network consists of searching for the right weights associated with the links between the nodes. We are seeking a local minimum on the error surface where the parameters are the weights. However, the selection of the weights is an optimization problem with many local minima. Because the initial weights are chosen randomly, the chances of two neural networks trained on the same data converging to the same local minimum is unlikely. This would imply that they would most probably generate different errors. It has been shown that, in an ensemble of neural networks, if each network can get the right answer more than half the time and network responses are independent then the more networks used the less the likelihood of an error by a majority decision rule [4].

In [4] the authors argue that the collective decision made by the ensemble is less likely to be in error than an individual network. They also give different models for the correlation of the networks in the ensemble.

Most neural network ensemble methods make use of error independence to build classifiers that complement each other. Two known methods that promote error independence among the individual classifiers are bagging and boosting which will be explained in section 2.6.

## **2.5 Combination Strategies**

Once the structure of the ensemble is determined and the classifiers have been trained, then a combination strategy needs to be used. A combination strategy is how the classifier results in an ensemble are combined to make a final decision. Usually the results of all classifiers are considered. These combination methodologies are described in [26].

### **Highest Confidence**

When using the highest confidence the classifiers in the ensemble are ranked and the class with the highest one is chosen to be the correct one.

## **Majority Voting**

Majority voting works by counting the number of votes for each digit and selecting the digit with the highest number of votes. Prior performance is not considered when equal weight is given to the results of all classifiers. However, weights can be assigned to certain classifiers based on prior performance. In the event of a tie the confidence values can be used if they are available.

## **Borda Count**

The Borda count is a consensus function. For each class we determine the sum of the number of classes ranked below it by each classifier. The ranking is determined by arranging the classes so that their Borda counts are in descending order. The class with the highest Borda count is considered to be the correct classification.

## **Bayesian Combination Rule**

This method uses information derived from the training set to determine the combination strategy. It takes into consideration the performance of each expert on the training samples of each class. The confusion matrix is used to derive this information. The confusion matrix for the classification results of digits, is a two dimensional matrix where the columns and rows range from 0 to 9. The rows represent the actual class and the columns the predicted class. Each entry in the matrix is the number of samples that were predicted to be in the class represented

by the column. This matrix is useful in determining the types of mistakes that a particular classifier tends to make.

## **2.6 Ensemble Methods**

As mentioned earlier, many methods have been developed to increase the error independence among the classifiers in an ensemble. These methods are employed to determine the structure of an ensemble and this section gives an overview of the ones used most often.

### **Bagging**

Bagging attempts to reduce error correlation among the classifiers by training them on different datasets of the training set [4]. The datasets are formed by random selection with replacement of the training samples.

These are bootstrap replicates of the original training set. For a training set with  $N$  examples a bootstrap replicate is formed by resampling  $N$  examples with replacement. Some examples may appear more than once and others may not appear in the sample at all. The training sets are independent and the classifiers could be trained in parallel. Bagging performs well when the classifiers are unstable, that is, when changing the learning set changes the behavior of the classification.

## **Boosting**

Boosting is a general method applied to learning algorithms to generate highly accurate composite classifiers [22]. The method combines classifiers that are considered to give “weak hypotheses” because they are only required to do moderately well on the training set.

AdaBoost is a well-known boosting algorithm that has been applied to neural networks [16]. The algorithm trains a predetermined number of classifiers  $T$ . To do so,  $T$  training sets are generated sequentially and the classifiers are trained on these sets. In each round  $t$ , the classifier is trained with respect to a probability distribution on the training set. Initially, a uniform probability distribution is assigned. Each training example has an equal chance of being chosen. After the first classifier is trained, a new set is generated by classifying the training set with the initial classifier and assigning new weights to the samples. The weight on a given sample is determined according to whether the result was correct or not. The probability of incorrect samples is increased and the probability of correct samples is reduced. This is done to put more emphasis on harder to learn samples in subsequent classifiers. A weighted voting scheme, based on the classifier’s performance on the training set, is used when classifying test samples.

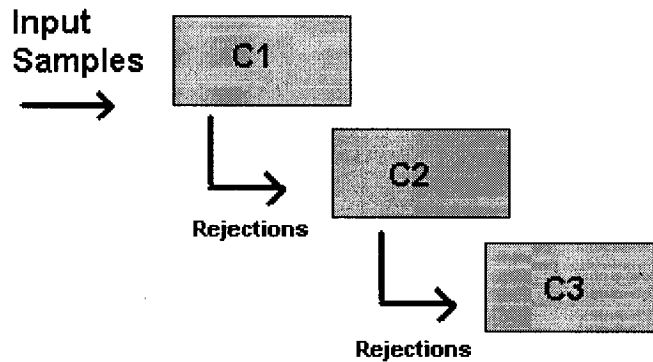
## **Random Subspace Method**

A method called Random Decision Forests proposed in [5] describes an ensemble method for decision trees. The trees in the ensemble are built using randomly selected subspaces of the feature space and the combination has been shown to monotonically improve classification accuracy as the number of trees is increased. A discriminant function is used to combine the outputs of the tree classifiers. The selection of random subspaces of the feature space has been shown to promote error independence among classifiers. Each tree learns 100% of the training data however the errors in generalization are different due to the fact that different subspaces of the feature space have been used.

### 3. Proposed Method

The proposed method uses an ensemble of neural networks to improve the classification accuracy of handwritten digits. In this section I will describe how the ensemble is built and how the outputs are later combined. I will also give additional explanations on the possible advantages of the method.

The purpose of the method is to use error independence among the classifiers to detect samples that are harder to learn or classify. The structure of the ensemble is based on a cascading neural network model. Figure 2 is an example of such a model. The first level, C1, consists of a pre-determined number of neural networks trained on the complete training set. Each neural network is trained with different feature subspaces. Once the training of each neural network is completed, a new database of samples is generated from the training database used on the first level. The neural networks on the second level are trained on this new database each using a different feature subspace. This procedure is followed until it is determined to stop training by the results of cross-validation.



**Figure 2: Cascading Neural Network**

The idea is to use error independence among the neural networks to identify samples that are difficult to learn and to allow a subsequent level to learn these samples. This allows the neural networks on the first level to learn the more general cases while the subsequent classifiers learn the exceptional cases. The following sections describe the training and classification procedures used in the method.

### **3.1 Training**

Cross-validation is used to determine how much each network should be trained. To do so, a subset of the training set which is not used for training is set aside. After a certain number of iterations of the learning algorithm the classifier is used to classify the samples in the validation set. Training of each classifier is stopped once the classification rate decreases.



Once the networks on a certain level have been trained, the training set associated to that level is classified using the neural networks of that level. If at least one network disagrees with another network then the sample is considered to be hard to learn and is put into a different database. If all classifiers agree then the posterior probability of each network is calculated and the average is taken. If this value is below a threshold then the sample is also rejected and put into the database. The optimal threshold is determined using the validation set. The goal is to allow the following levels to learn more about the samples that harder to learn.

If the classifiers on one level complement each other perfectly then they should only agree on correct classifications. If they do agree on incorrect classifications then the confidence determined by the average posterior probability should not be high.

The networks on the following level are trained on the new database. Since the number of samples in this new database is less than the previous one we do not need the same number of parameters for the new neural networks. The number of hidden nodes is decreased proportionally to the reduced size of the database. The process continues until either the complete training set has been learned or cross-validation has determined to stop training. The following gives a description of the training method:

$i=0$   
 $D_i = \text{Complete Training Set}$

Repeat:

1. For every feature space  $F_j$ , train classifier  $C_{ij}$  on  $D_i$  until cross-validation decides to stop training.
2. Classify training set  $D_i$  using classifiers  $C_{ij}$  for  $j=0$  to  $j=\text{\#of feature spaces}$ .
3. If at least one classifier disagrees or the average posterior probability of the agreements is less than a threshold then reject the sample and put it in database  $D_{i+1}$ .
4.  $i=i+1$

Until  $(|D_i| == 0)$  or cross-validation of the complete ensemble decides to stop the training.

The structure of the cascading neural network is determined dynamically.

We do not know in advance how many neural networks and how many levels there will be. The ensemble is considered to have converged when it is determined that additional levels are not required.

### 3.2 Classification

Classification of test samples does not always require all neural networks in the ensemble. It is a sequential process that starts by classifying the sample using all the neural networks on the first level. An agreement occurs when all the classifiers on the same level agree on the classification result. If the classifiers do

not agree or if they agree but have an average posterior probability below a threshold then the sample is rejected and passed on to the following level.

Otherwise, the classification result is accepted. Therefore, an acceptance occurs when all classifiers on a given level agree with an average posterior probability greater than the determined threshold. The process continues until the sample is either accepted by a level or rejected by all levels. In the latter case post-processing can be applied to determine the class of the sample. The following gives a description of the classification method:

For every sample in the classification set:

$i=0$

While the class of the sample has not been determined and ( $i < \#$  of levels):

1. For every feature space  $F_j$ , classify the sample with classifier  $C_{ij}$
2. If all classifiers agree with posterior probability greater than or equal to a threshold then accept the classification. In that case, the class of the sample has been determined.
3.  $i=i+1$

If the class of the sample has not been determined then apply post-processing to determine the class.

Like Boosting, the additional networks during the training phase focus on samples that are harder to learn. However, a sample is considered to be hard to learn if there is either at least one disagreement among the networks on one level or the average posterior probability of the agreement is below a threshold.

During the classification of test samples, each level allows for the identification of samples that are harder to classify. The same rejection method is used as in the training phase. This allows for an increase in the classification accuracy by extracting images that are more difficult to classify given the complexity of the classifiers, the selected feature spaces and the training data. By focusing only on these samples we increase the chances of correctly classifying samples that are more difficult to classify while reducing the chances of misclassifying samples that are easy to classify.

### **3.3 Generalization on each Level**

The generalization ability of the neural networks is reduced at each level since the training set is reduced during training. However if the classifiers have been shown to complement each other well, then different generalization errors should occur at each level and the misclassifications should be detected and rejected. We count on the fact that the networks at each level will make different mistakes.

### 3.4 Speed

Unlike Boosting, we do not always need all neural networks during classification. If the training set is representative of test samples then most classifications should be accepted at the initial levels. As the focus is narrowed down to the more difficult samples more processing power is required.

### 3.5 Complementing Classifiers

As demonstrated in [5], where decision trees in an ensemble are trained on different feature subspaces, as the number of classifiers increase, the error rate decreases monotonically. Since this is true, then the number of incorrectly accepted classifications using the rejection method proposed in this paper, should also decrease as the number of classifiers is increased. This is because the chance of at least one classifier making the correct decision is increased and therefore a disagreement is generated.

For the method to work properly, the neural networks on each level must complement each other so well that there is at least one disagreement on all samples that are hard to classify and the networks only agree on samples that are correctly classified. If all classifiers have a high classification accuracy and they complement each other, i.e. there is a low correlation among the classifiers, then if the training set is representative of test patterns, the classifiers should correctly agree on most classifications and correctly disagree on most misclassifications.

If  $e_i$  is the error rate of classifier  $i$  then the maximum rate of rejections at each level would be the sum of  $e_i$ . Since the error rate of each classifier is usually quite low and classifiers usually misclassify the same difficult patterns then the rejection rate should also be low. The number of undetected misclassifications can be a measure of the correlation among the classifiers in the ensemble.

The method used in [5] learns 100% of the training samples but relies on different generalization errors to increase generalization accuracy. It increases generalization while avoiding over-fitting. It is a coverage optimization method. The method proposed in this thesis learns until either the complete training set has been learned, or cross-validation of the complete ensemble decides to stop training. It increases generalization accuracy by rejecting samples on a given level and learning more about the rejected set in subsequent layers. Error independence among the classifiers ensures that a disagreement will be generated for misclassified samples.

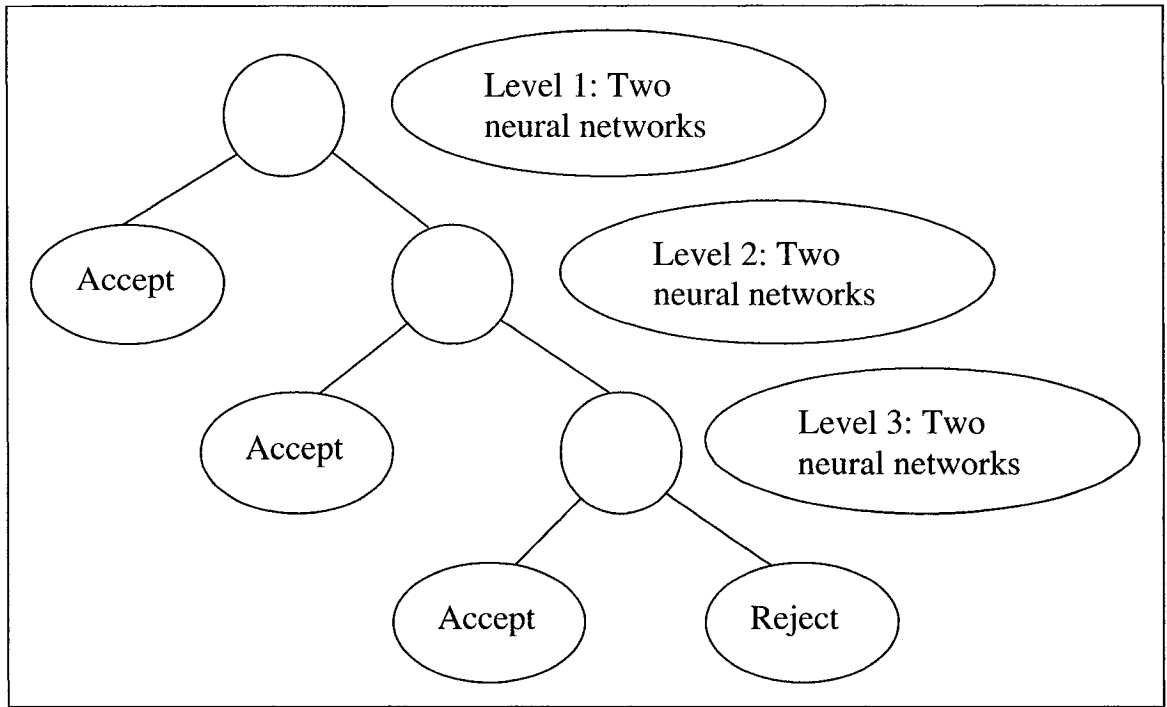
Since the training set is reduced at each level it allows for the generalization on a smaller subset of the training set. This smaller subset represents patterns that deviate from the general case. We are able to concentrate on cases that are harder to learn and training neural networks on these subsets allows for an increase in the confidence of the validity of a classification. The chance of a set of neural

networks with low error correlation agreeing with very high confidence on a classification is very low. For networks that complement each other very well, the pattern must look like the digit as it was classified.

In the end, we are left with some undetected misclassifications at each level and digits that have been identified as “difficult to classify”. We can reduce the undetected misclassifications by adding networks that complement each other at each level. We can increase the classification accuracy by concentrating on the more difficult patterns.

### **3.6 Ensemble as a Decision Tree**

The cascading neural network ensemble can be seen as a decision tree. Figure 3 demonstrates this. The decision at each level is whether to accept or reject the classification by the neural networks of that level. The training of the tree is done by learning the samples of the training set and determining an optimal rejection threshold with the use of the validation set. If the complete training set is learned, the method partitions the training set into smaller subsets such that each one has been learned by a set of neural networks on one level of the tree. These neural networks agree on the classification of the training sample within the determined threshold.

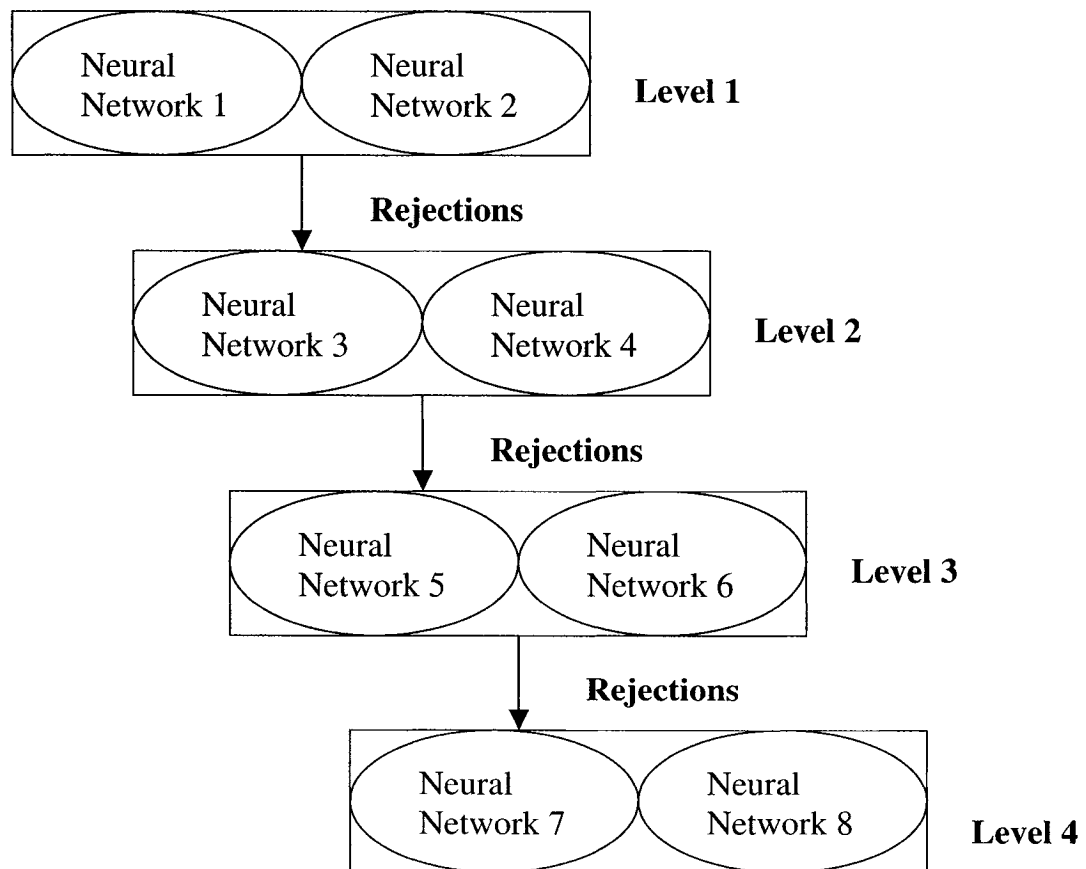


**Figure 3: Ensemble as a Decision Tree**



## 4. Implementation





A cascading neural network ensemble was implemented using the method described in the previous section. Figure 4 gives an overview of the structure of the implemented ensemble. As can be seen there are 4 levels, with two neural networks per level. The two networks on each level have been trained on different features. The chosen feature spaces remain the same for all levels. In this section I will describe the implementation of the recognition system used for experimentation.



**Figure 4: Implemented Neural Network Ensemble**

#### 4.1 Preprocessing and Feature Extraction

Before extracting the features, each image containing one digit was preprocessed. Since the background of the image is black (i.e. pixel values were 0) then a simple binarization was applied by setting every value greater than 0 to 1. Following that, isolated pixels were set to 0 to remove noise. A thinning operation, which removes a layer of pixels around the digit, was also applied. The first neural network on each level was trained on these pixel values. For the second neural network on the same level, the largest connected component in each image was detected and the chain codes were extracted. Examples of largest connected components are shown in figure 5.

Handwritten Digit	Largest Connected Component
	
	

**Figure 5: Largest Connected Components**

The chain codes of an object in an image are a sequence of numbers that describe the shape of the object. Each value of the chain code gives information on the

direction of the next pixel connected to the object. This sequence of numbers was given as input to the second neural network.

## 4.2 Neural Network Structures

The neural networks are feed-forward and fully connected. Table 1 describes the structure of the neural networks on the first level. The structure of the two networks on the same level is identical.

<b>Levels</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>
# of inputs	784	784	784	784
# of hidden layers	1	1	1	1
# of outputs	10	10	10	10
# of nodes in hidden layer	70	33	15	11

**Table 1 Neural Network Structure**

The number of inputs is the number of pixels per image sample. The number of outputs is 10 since we would like to represent the number of digits. Initially, there are 70 nodes in the hidden layer. For each subsequent layer the number of nodes in the hidden layer is decreased proportionally to the reduced size of the database.

### **4.3 Neural Network Training**

The back-propagation learning algorithm is used to learn the training set. A subset of the training set is kept for validation. A minimum of 1000 training iterations was done for each neural network. After this initial training period, the neural network was tested on the validation set after every 50 iterations. If the error rate on the validation set increased by any amount or did not decrease within the last 50 iterations, then training of the neural network was stopped. Testing on the validation set allows for good generalization ability by each neural network. If some over-fitting does occur then error independence among the classifiers on the same level allows for the rejection of misclassified samples.

### **4.4 Ensemble Training**

Once the neural networks on a given level are trained, harder to learn images are rejected from the training set. This is done by classifying the training set associated to the level using the networks on that level. If the two networks disagree on the classification then the sample is rejected and put into the next database. If the networks agree but have an average posterior probability less than 0.9995 then they are also rejected and put into the next database. This was the optimal threshold found by testing the neural networks on the validation set. If the sample is accepted then the image is excluded from the next level. The procedure was stopped at the 4<sup>th</sup> level.

## 4.6 Classification

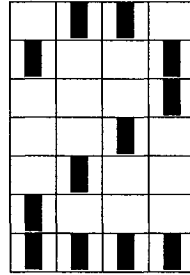
For classification of test samples the process starts with the first level. The classification is done sequentially until a set of neural networks on one level agrees on the classification of the sample with average posterior probability greater than or equal to 0.9995. If none of the levels meet the requirement, then the sample is passed on to post-processing.

## 4.7 Post-Processing

During classification of test samples, if an image has not been accepted on any of the levels of the ensemble then it is rejected and passed on to the post-processor. An additional neural network was trained for this stage. The neural network was trained on the horizontal distance features of the training set (not including the validation set). The horizontal distance feature of a pixel in an image gives the direction and the distance to the nearest white pixel on the same horizontal line. An example of this is given in figure 6.

The classification results of the two neural networks on the first level, along with this new neural network are used to make a final decision. If all three networks agree on the classification then it is accepted as the result. Otherwise, if one classifier disagrees but the other two agree then a decision is made based on the previous performance of the classifiers. If all three classifiers disagree then for the experiment, the one with the highest posterior probability was chosen to be the

result. However, the sample can be rejected since there is no definitive consensus on the result.



-1	0	0	1
0	1	-1	0
-3	-2	-1	0
-2	-1	0	1
-1	0	1	2
0	1	2	3
0	0	0	0

**Figure 6: Horizontal Distance Features of the digit 2**

## 5. Experimental Results

Experiments were done on the MNIST database. This database contains 60000 images for training and 10000 images for testing. Each image is 28x28. Of the 60000 training samples, the experiments done for this paper used 50000 images for training and the remaining 10000 images were set aside for validation. After a minimum number of 1000 training iterations, the neural networks were tested on the validation set after every 50 iterations. If the error rate increased by any amount or if the classification rate did not improve then training was stopped. All 10000 testing samples were used to test the neural network ensemble.

### 5.1 Training Databases

The following table shows the number of images in each database associated to the levels of the neural network ensemble:

Level	# Training Images in Database
1	50000
2	23864
3	11024
4	7470

**Table 2 Training Databases**

As expected, most of the training samples were learned by the first two levels. The easier samples can be generalized relatively well. However, as the levels in the ensemble increase the samples become harder to learn and generalize. This can be seen by the small decrease in the number of training samples between levels 3 and 4.

## 5.2 Testing

Table 3 shows the number of samples accepted and the number of samples rejected at each level. As expected, there are more samples accepted on the first level than any other level. This is because the neural networks on the first level were trained on the largest training set which allows for better generalization. When comparing the classification result on the complete test database between the neural networks trained on the pixel values from the first level to the one on the second level we obtain a classification rate of 94.43% versus 93.37%.

Level	# Samples Accepted	# Samples Rejected
1	4912	5088
2	2127	2961
3	466	2495
4	289	2206

**Table 3 Accepted and Rejected Samples**



A sample has been correctly accepted if all neural networks on a given level correctly agree on the classification with a very high posterior probability. A sample has been falsely accepted if at a given level, all classifiers agree on the incorrect classification with a very high confidence level. This is important because once a classification has been accepted then it will no longer be reconsidered for classification. The more accepted misclassifications there are on a given level of the cascading classifier the higher the error rate will be.

<b>Level</b>	<b># Correct Accept</b>	<b># False Accept</b>
1	4901	11
2	2112	15
3	454	12
4	275	14
<b>Total</b>	<b>7742</b>	<b>52</b>

**Table 4 Correct and False Accepts**

It would also be interesting to see the number of samples that were rejected because of a disagreement, as opposed to the number of samples rejected because of a low average posterior probability. The following table shows the number of disagreements at each level:

<b>Level</b>	<b># of Disagreements</b>
1	1469
2	1526
3	1570
4	1585

**Table 5 Disagreements**

The following table shows the number of agreements with low average posterior probability at each level:

<b>Level</b>	<b># Classifications with Low Average Posterior Probability</b>
1	3619
2	1435
3	925
4	621

**Table 6 Agreements with Low Posterior Probability**

The number of disagreements has changed very little between the top and bottom levels. However, the number of agreements with low posterior probability

has decreased dramatically. As the levels increase, the average posterior probability among agreements increases and samples are accepted.

In the end we are left with 52 falsely accepted samples and a total of 2206 rejected samples sent to post-processing. This indicates that the system was able to reduce the test set to a smaller subset of samples that are more difficult to classify while only leaving behind 52 misclassified samples out of the 7794 accepted samples.

### **5.3 Classification Results**

The classification rate of the neural network trained on 50000 images with chain codes features of the largest components is 86.86%.

The classification rate of the neural network trained on 50000 images with pixel value features is 94.43%.

When applying the cascading neural network model to the classification, the post-processing method described in section 4.7 was applied. The following table shows the improvements in the classification rates with post-processing:

<b>Level</b>	<b>Classification Rate</b>
1	94.43%
2	96.51%
3	96.58%
4	96.58%

**Table 7 Classification Rates**

Therefore, the overall classification rate of the system is 96.58%. Although most classification systems reach an accuracy rate of over 99%, the proposed method reduces the training set to a smaller subset of images that are harder to classify. That is, the samples that are rejected from the cascading neural networks are considered to be difficult to classify. The classification rate of the overall system can be increased if the post-processor has high classification accuracy on this smaller subset.

Table 8 gives the classification results obtained from the system and the possible trade-offs. The results show that by rejecting images where the neural networks on each level disagree on the classification results, the error rate decreases by more than 23%.

<b>Performance</b>	<b>No rejections</b>	<b>Reject if all classifiers disagree</b>
Best Recognition %	96.58	95.61
Rejection %	0	1.8
Error %	3.42	2.61

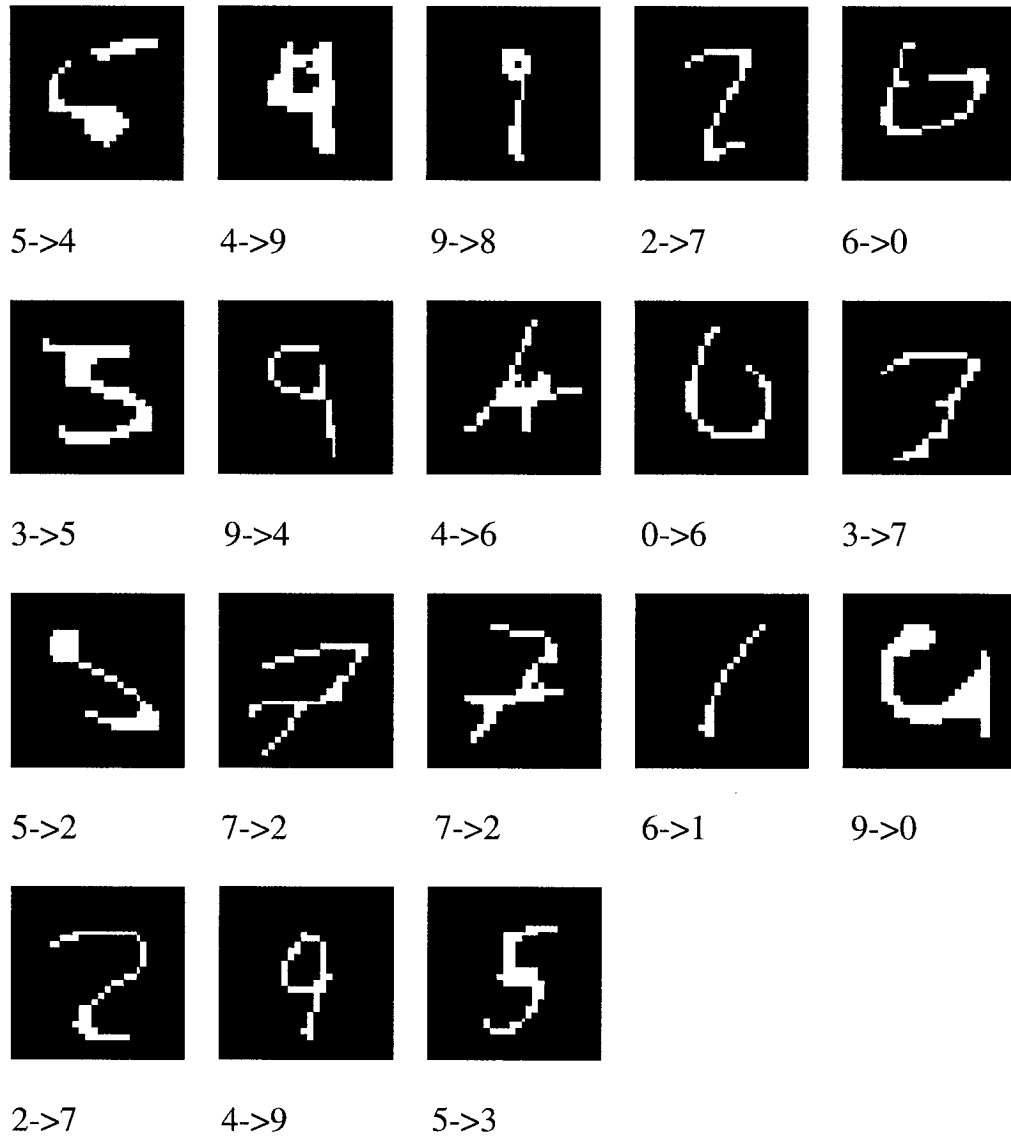
**Table 8 Classification, Rejection and Error Trade-offs**

#### **5.4 Speed**

For samples accepted on the first level the classification time is 93 samples per second on an Athlon AMD 1600+ processor. This means that the classification of each image accepted on the first level takes about 11 milliseconds. Every additional layer takes approximately the same time. Therefore, for a sample accepted at level  $t$  the classification time is  $t \cdot 11$  milliseconds.

#### **5.5 Image Samples**

Two types of images are considered especially difficult to classify. The first are samples that were accepted by an agreement among neural networks with a very high average posterior probability. The following are some images from this category. The digit on the left is the label and the one on the right is the classification result:

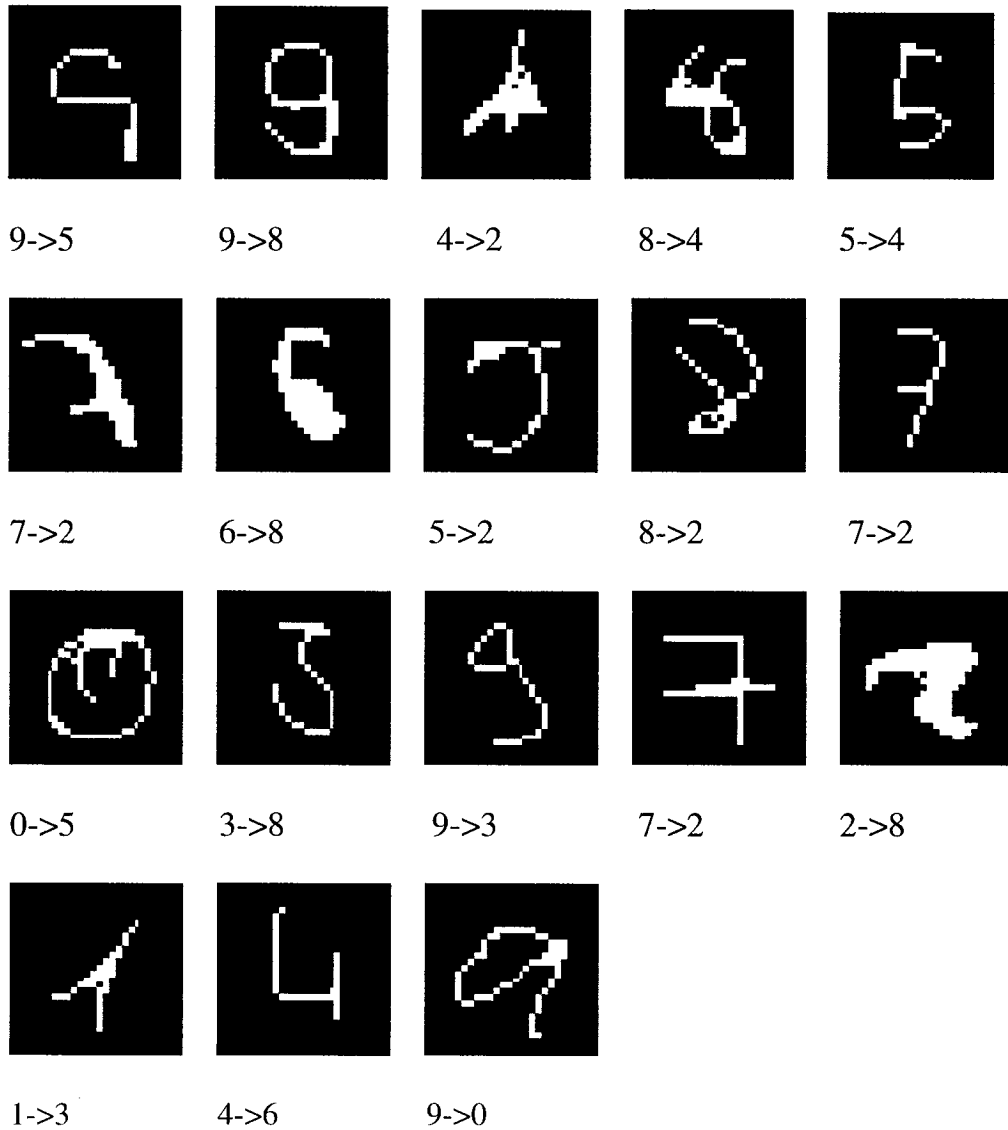


**Figure 7: Images Accepted by Agreements with Very High Confidence**

As can be seen, many of these images are in fact confusing. However, with better feature extraction there is room for improvement in at least raising a doubt on the classification result. Some of these images are clear enough to be classified properly. Using the largest connected component might not be enough information for cases like the first sample in figure 7. The top of the number 5 would be excluded and the digit would look more like the number 4.

The second category is when there are no agreements among the classifiers.

The following are some images from this category. The digit on the left is the label and the one on the right is the classification result which is obtained by taking the result of the classifier with the highest confidence:



**Figure 8: Images Rejected by all Levels**

It is clear that some of these images are more difficult to classify however there are enough identifying features to extract better information and classify them correctly. Interesting information on these types of images would be geometric features such as the number of holes, number of corners, number of endpoints, number of curves and the location of the curves.

### 5.6 Types of Agreements and Disagreements

There are different categories of agreements and disagreements generated on each level. Each level contains two neural networks and table 9 shows the types of agreements and disagreements generated by them:

<b>Classification Result from Neural Network 1</b>	<b>Classification Result from Neural Network 2</b>	<b>Agreement</b>	<b>Disagreement</b>
Correct	Correct	Yes	No
Incorrect	Incorrect (same result as Neural Network 1)	Yes	No
Incorrect	Incorrect (different result from Neural Network 1)	No	Yes
Correct	Incorrect	No	Yes

**Table 9 Types of Agreements and Disagreements**



Obviously, the goal is to reduce the second case where we have two incorrect classifications that are in agreement. However, it would also be interesting to see how many of the disagreements lie in the third case. That is, among the disagreements, how many of them were incorrectly classified by both classifiers and how many were correctly classified by at least one classifier. The results are given in the following table:

<b>Level</b>	<b>Correctly Classified by at Least One Classifier</b>	<b>Incorrectly Classified by All Classifiers</b>
1	1027	442
2	1068	458
3	1021	549
4	943	642

**Table 10 Results for Types of Disagreements**

From the results given in Table 10, we can see that most of the samples were correctly classified by at least one neural network. This suggests there may be a way to identify samples that are consistently misclassified by including additional classifiers on each level trained on different feature subspaces.

## 5.7 Additional Experiments

Additional experiments were done with different features given to the second network on each level. Like the original experiment, a cascading neural network ensemble was constructed with two neural networks on each level trained on different features. The first neural network on each level was again trained on the pixel values and the second one on the horizontal pixel distance features. There were a total of 4 levels and table 11 shows the classification results as the additional levels were added. The same behavior is observed as in the original experiment. That is, the classification accuracy increases as the number of levels increase. There were no rejections in this experiment and table 11 shows the error rates accordingly.

<b>Level</b>	<b>Classification Rate</b>	<b>Error Rate</b>
1	95.91%	4.09%
2	96.31%	3.69%
3	96.33%	3.67%
4	96.34%	3.66%

**Table 11 Classification Results from Additional Experiments**

## 6. Suggested Improvements

This section provides suggestions for improving the classification accuracy and the rejection rate of the proposed method. I will discuss some improvements required in the pre-processing and post-processing methods and also suggest ways to improve the classification accuracy by changing the structure of the ensemble.

Many experiments have shown that good pre-processing greatly improves the classification accuracy of an OCR system. Specifically, introducing normalizations such as shifting, and scaling of training samples has yielded very good results. Also, deslanting the digits will most likely result in better classification results. Experiments with the proposed method are required to demonstrate this.

Improvements in the post-processing stage can also be made. As mentioned in a preceding section, support vector machines yield very good classification results. Since we are able to reduce the test set to a smaller set that is more difficult to classify, then a hybrid ensemble system can be used. Support vector machines have a high computation cost. However, it may be used on a smaller subset of the test set. As long as the false accepts are kept at a minimum at each level then the additional classifier has a chance of correctly classifying the sample. Contextual information can also be used to confirm classification results.

For example, in bank check processing systems where we are trying to classify the digits in the courtesy amount field, the information extracted from the legal amount can be used to confirm the classification.

Also, certain images require additional information taken from the context to confirm that the input is in fact a digit. Otherwise, the input might not be a digit but the classification system will try to classify it as one. However, this can be a limitation on the database used for the experiments, such as for MNIST, and in that case it is assumed that the input is in fact a digit.

Improvements on the structure of the ensemble can also be made. The most obvious one is increasing the number of neural networks at each level. Each network should be trained on different feature subspaces. There has been quite a bit of research on the selection of feature subspaces that result in highly accurate classifiers. Since the classification accuracy increases in [5] as more classifiers are added, then the rejection accuracy should also increase at each level in the proposed method. There is a trade-off between the number of classifiers and the rejection rate. Too many classifiers might mean too many rejections if the individual classifiers do not have high classification accuracy. But if most classifiers have high classification accuracy then most should agree on correct classifications.

Another improvement on the structure would be to train the neural networks on features which differ from one level to another. That is, the neural networks on one level can be trained on different features from the networks on the previous level. This can help since the rejected samples from the previous level might have features that were not considered by the networks on the previous level.

As the number of classifiers on each level is increased, a more tolerant rejection criteria can be used. There can be a threshold set by considering the number of required agreements in the majority of the classifications. Samples can be rejected if the number of agreements does not reach the threshold on each level. Also, boosting can be used to improve error independence on each level and different feature subspaces can be used on every level.

## 7. Conclusion

The proposed method uses complementing classifiers to detect samples that are harder to learn or classify. The neural networks are trained on different feature subspaces and this allows for the detection of the majority of misclassifications. It uses a cascading neural network model to focus on samples that are more difficult to learn and classification accuracy increases when the method is applied to test samples.

It is necessary to do more experiments with additional neural networks trained on different features on each level to help extract as many misclassifications as possible. Since many samples are accepted within the first two levels, we can take advantage of the speed of neural networks while applying more processing power on the more difficult samples. This method is a good complement to support vector machines that require a higher computational power.

## 8. References

- [1] E. Bauer, R. Kohavi, "An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants," *Machine Learning*, 36 (1-2): 105-139, 1999
- [2] B.E. Boser, I.M. Guyon, V.N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, D. Haussler, Ed., 144-152, ACM Press, Pittsburg, PA, 1992
- [3] L. Breiman, J.H. Friedman, R. A. Olshen, and C.J. Stone, "Classification and Regression Trees," Wadsworth International, Monterey, California, 1984
- [4] L. Breiman, "Bagging Predictors," *Machine Learning*, 24 (2): 123-140, 1996
- [5] L. Breiman, "Random Forests," *Machine Learning*, 45 (1): 5-32, 2001
- [6] T.M. Cover, "Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition," *IEEE Trans. Electronic Computers*, 14: 326-334, 1965
- [7] T.M. Cover, P.E. Hart, "Nearest Neighbor Pattern Classification," *IEEE Transactions on Information Theory*, Volume IT-13 (1): 21-27, 1967
- [8] R. Duda, P. Hart, "Pattern Recognition and Scene Analysis," John Wiley and Sons, 1973
- [9] N. Gorski, "Optimizing error-reject trade off in recognition systems," In 4th Int. Conf. on Document Analysis and Recognition, 2: 1092-1096, Ulm, Germany, 1997
- [10] L. K. Hansen, P. Salamon, "Neural Network Ensembles," *IEEE Trans. Pattern Anal. Mach. Intell.*, 12 (10): 993-1001, 1990
- [11] T. K. Ho, "Random Decision Forests," *ICDAR*, 278-282, 1995
- [12] T. K. Ho, "The Random Subspace Method for Constructing Decision Forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(8): 832-844 , 1998

- [13] T. K. Ho, "Complexity of Classification Problems and Comparative Advantages of Combined Classifiers," *Multiple Classifier Systems*, 97-106, 2000
- [14] T. K. Ho, "Multiple Classifier Combination: Lessons and Next Steps," *Hybrid Methods in Pattern Recognition*, A. Kandel and H. Bunke, eds., World Scientific, 2002
- [15] C. Kaynak, E. Aplaydin, "MultiStage Cascading of Multiple Classifiers: One Man's Noise is Another Man's Data," *ICML*, 455-462, 2000
- [16] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 86 (11): 2278-2324, 1998
- [17] Y. Lee, "Handwritten Digit Recognition Using k-Nearest Neighbor, Radial-Basis Functions, and Backpropagation Neural Networks," *Neural Computation*, 3(3): 440-449, 1991
- [18] E. Levin, N. Tishby, S.A. Solla, "A statistical approach to learning and generalization in layered neural networks," *Proceedings of the IEEE*, 78 (10): 1568-1574, 1990
- [19] C. M. Nunes, A. de Souza Britto Jr., C. A. A. Kaestner, R. Sabourin, "Feature Subset Selection Using an Optimized Hill Climbing Algorithm for Handwritten Character Recognition," *SSPR/SPR*, 1018-1025, 2004
- [20] J. F. Pitrelli, M. P. Perrone, "Confidence-Scoring Post-Processing for Off-Line Handwritten-Character Recognition Verification," *ICDAR*, 278-282, 2003
- [21] D.E. Rumelhart, G.E. Hinton, R.J. Williams, "Learning internal representations by error propagation," *Parallel Data Processing*, Vol.1 D. Rumelhart and J. McClelland, editors. The M.I.T. Press, Cambridge, 318-362, 1986
- [22] R. E. Schapire, "The Strength of Weak Learnability," *Machine Learning*, 5(2): 197-227, 1990
- [23] R. E. Schapire, "A Brief Introduction to Boosting," *IJCAI*, 1401-406, 1999
- [24] H. Schwenk, Y. Bengio, "AdaBoosting Neural Networks: Application to on-line Character Recognition," *ICANN*, 967-972, 1997



- [25] H. Schwenk, Y. Bengio, "Boosting Neural Networks," *Neural Computation*, 12(8): 1869-1887, 2000
- [26] C. Y. Suen, L. Lam, "Multiple Classifier Combination Methodologies for Different Output Levels," *Multiple Classifier Systems*, 52-66, 2000
- [27] G. T. Toussaint, "Bibliography on estimation of misclassification," *IEEE Transactions on Information Theory*, 20: 472-479, 1974
- [28] K. Tumer, J. Ghosh, "Error Correlation and Error Reduction in Ensemble Classifiers," *Connect. Sci.*, 8(3): 385-404, 1996
- [29] V. Vapnik, "The Nature of Statistical Learning Theory," Springer, New York, 1995