Experiments with RST, A Rotation, Scaling and Translation Invariant Pattern Classification System

Cem Yüceer and Kemal Oflazer

Department of Computer Engineering and Information Science
Bilkent University, Bilkent, 06533 Ankara, Türkiye

Abstract

This paper includes an overview and experimental results on RST, the hybrid pattern classification system. It can recognize patterns even when they are deformed by a transformation like rotation, scaling, and translation or a combination of these [11]. The system is formed of a Karhunen-Loéve transform based pattern preprocessor, an artificial neural network classifier and an interpreter. After a description of the system architecture, experimental results are provided from three different classification domains: classification of letters in the English alphabet, classification of the letters in the Japanese Katakana alphabet, and classification of five main geometric figures. The system is general purpose and has a reasonable noise tolerance.

1 Introduction

The recent interest in artificial neural networks, machine learning, and parallel computation has led to renewed research in the area of pattern recognition. Pattern recognition aims to extract information about the image and/or classify its contents. Systems having pattern recognition ability have many possible applications in a wide variety of areas, from simple object existence checks, through identity verification, to robot guidance in space exploration. Pattern classification, a subfield of pattern recognition, is concerned with determining whether the pattern in an input image belongs to one of the predefined classes. Many researchers studied parametric Bayesian classifiers where the form of input distributions is assumed to be known and parameters of distributions are estimated using techniques that require simultaneous access to all training data. These classifiers, especially those that assume Gaussian distributions, are still the most widely used since they are simple and are clearly described in a number of textbooks [4, 5]. However, the thrust of recent research has changed. More attention is being paid to practical issues as pattern classification techniques are being applied to speech, vision, robotics, and artificial intelligence applications where real-time response with complex real world data is necessary. In all cases, pattern classification systems should be able to learn while or before performing, and make decisions depending on the recognition result.

Developing pattern recognition systems is usually a two-stage process. First, the designer should carefully examine the characteristics of the pattern environment. The result is a set of features chosen to represent the original input image. Second, the designer should choose from a variety of techniques to classify the pattern which is now in feature representation. The stage of feature determination and extraction strictly determines the success of the system, since from thereon the image is represented by this feature form. Therefore, it is highly desired that the classification system itself should extract the necessary features to differentiate the example patterns that represent each class. In other words, the system should be automated to work by itself and should not depend on the human designer's success in defining the features. Further, these features should be chosen such that they should tolerate the differentiation between the patterns in the same class. The system should also have the ability to perform the classification in a rotation, scaling, and translation invariant manner. This effect is typical when the scanning device, suppose a camera, changes its orientation or distance from the specimen. Hence the image fed to the system may contain a pattern that is rotated, scaled, or translated compared to its original form when it was first presented to the system. For such a case, either the system should employ features that are invariant to such transformations or there should be a preprocessor to maintain the rotational, scaling, and translational invariancy. Even for a limited system designed for classifying only a determined type of patterns – an optical character classifier, or an identity verifier – it is hard to find features that extract useful information while maintaining the mentioned invariancies. The problem will be impractical if such a system is intended for general purpose classification, or to say it is aimed to classify any type of patterns. Artificial neural networks have recently been used for automatic feature extraction and/or pattern classification mainly owing to their learning algorithms, generalization ability and noise tolerance [1, 2, 3, 6, 8, 10, 11].

2 The Pattern Classification System

Selecting good features for relatively complex patterns, like the human face or finger prints, turns out to be impractical or even impossible [9]. The problem is more acute when there is no prior knowledge on the patterns to be classified. Therefore, a system to automatically extract the useful features is essential. Artificial neural networks extract information during the training process. RST, the pattern classification system first presented in [11] has a modular structure consisting of three main blocks, a preprocessor, a classifier, and an interpreter. The blocks are cascaded in order such that the original image is first preprocessed, then classified, and finally the results are interpreted.

2.1 PREP1: The Preprocessor with Radial Scaling Correction

The preprocessor has three cascaded blocks *R-Block*, *S-Block*, and *T-Block*. The R-Block maintains rotational invariancy, S-Block maintains scaling invariancy, and T-Block maintains translational invariancy. The order in which the blocks are cascaded is determined mainly by the functional dependencies between these blocks. In the first implementation of the preprocessor, PREP1, the T-Block comes first, S-Block second, and R-Block last.

The T-block maintains translational invariancy by computing the center of gravity of the pattern and translating the image so that the center of gravity coincides with the origin. The resulting image is passed to the S-Block. The center of gravity, (x_{av}, y_{av}), is computed by averaging the x and y coordinates of the on-pixels. The mapping function for the translation invariant image is:

$$f_T(x_i, y_j) = f(x_i - x_{av}, y_j - y_{av}) \tag{1}$$

where function $f(x, y)$ gives the value of the pixel at the coordinates (x, y). For digitized binary-valued 2-D images this function will be either 0 or 1.[1]

The S-Block maintains scaling invariancy by scaling the image so that the average radius for the on-pixels is equal to one-fourth of the grid size. The term *radius* for a pixel is defined to be the length of the straight line connecting the pixel and the origin. The scale factor is:

$$s = \frac{R}{\frac{1}{\sum_{i=1}^{N}\sum_{j=1}^{N} f_T(x_i, y_j)} \sum_{i=1}^{N}\sum_{j=1}^{N} f_T(x_i, y_j) \cdot \sqrt{x_i^2 + y_j^2}} \tag{2}$$

where R is equal to one-fourth of the grid size. The mapping function for the scaling invariant image is:

$$f_{TS}(x_i, y_j) = f_T(s \cdot x_i, s \cdot y_j) \tag{3}$$

The R-block maintains rotational invariancy by rotating the image so that the direction of maximum variance coincides with the *x-axis*. The derivation of the function is based on the Karhunen-Loéve transformation which has been used in some applications [7, 9, 10, 11]. The transformation exploits the following: given a set of vectors, the eigenvector that corresponds to the largest eigenvalue of the covariance matrix calculated from the set of vectors, points in the direction of maximum variance [7, 9]. This property is used to maintain rotational invariancy since detection of the maximum variance direction will also reveal the rotation angle. Define:

$$T_{xx} = \sum_{i=1}^{N}\sum_{j=1}^{N} f_{TS}(x_i, y_j) \cdot x_i^2 \qquad T_{yy} = \sum_{i=1}^{N}\sum_{j=1}^{N} f_{TS}(x_i, y_j) \cdot y_j^2 \qquad T_{xy} = \sum_{i=1}^{N}\sum_{j=1}^{N} f_{TS}(x_i, y_j) \cdot x_i \cdot y_j \tag{4}$$

The *sine* and *cosine* of the rotation angle will be:

$$\sin\theta = \frac{(T_{yy} - T_{xx}) + \sqrt{(T_{yy} - T_{xx})^2 + 4 \cdot T_{xy}^2}}{\sqrt{2 \cdot \left[\sqrt{(T_{yy} - T_{xx})^2 + 4 \cdot T_{xy}^2} + (T_{yy} - T_{xx})\right] \sqrt{(T_{yy} - T_{xx})^2 + 4 \cdot T_{xy}^2}}} \tag{5}$$

[1]If x or y, the arguments of the function, are not integers then they are rounded to the nearest integer to obtain the pixel coordinates.

$$\cos\theta = \frac{2\cdot T_{xy}}{\sqrt{2\cdot\left[\sqrt{(T_{yy}-T_{xx})^2+4\cdot T_{xy}^2}+(T_{yy}-T_{xx})\right]\sqrt{(T_{yy}-T_{xx})^2+4\cdot T_{xy}^2}}} \tag{6}$$

The mapping function for the rotation invariant image is:

$$f_{TSR}(x_i, y_j) = f_{TS}(\cos\theta\cdot x_i + \sin\theta\cdot y_j\,,\ -\sin\theta\cdot x_i + \cos\theta\cdot y_j) \tag{7}$$

2.2 PREP2: The Preprocessor with Axial Scaling Correction

PREP1, performs radial scaling correction. That is, it uses the same scaling factor along all directions. This approach will perform better in certain applications like patterns being scanned in different resolutions in different dimensions. In preprocessor with axial scaling correction, called PREP2, the main blocks are reordered, such that T-Block comes first, R-Block is second, and S-Block is last. Further, the scaling factors are computed using a different function. In order to maintain rotational, scaling, and translational invariancy PREP2 computes various relevant parameters as:

$$T_{xx} = \left(\sum_{i=1}^{N}\sum_{j=1}^{N} f(x_i, y_j)\cdot x_i^2\right) - P\cdot x_{av}^2 \qquad T_{yy} = \left(\sum_{i=1}^{N}\sum_{j=1}^{N} f(x_i, y_j)\cdot y_j^2\right) - P\cdot y_{av}^2 \tag{8}$$

$$T_{xy} = \left(\sum_{i=1}^{N}\sum_{j=1}^{N} f(x_i, y_j)\cdot x_i\cdot y_j\right) - P\cdot x_{av}\cdot y_{av} \tag{9}$$

$$s_x = \sqrt{\frac{R_x\cdot P}{T_{xx}(\cos\theta)^2 + 2\cdot T_{xy}\cdot\cos\theta\cdot\sin\theta + T_{yy}(\sin\theta)^2}} \tag{10}$$

$$s_y = \sqrt{\frac{R_y\cdot P}{T_{xx}(\sin\theta)^2 - 2\cdot T_{xy}\cdot\cos\theta\cdot\sin\theta + T_{yy}(\cos\theta)^2}} \tag{11}$$

where s_x and s_y are the scaling factors, and R_x and R_y are the desired deviation values along the corresponding axes. R_x and R_y are equal to the grid size.

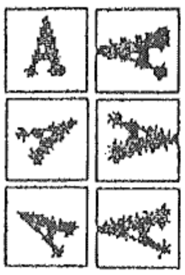
The mapping function for the preprocessor with axial scaling correction is:

$$\begin{aligned} f_{TRS}(x_i, y_j) &= f(s_x\cdot(\cos\theta\cdot(x_i - x_{av}) + \sin\theta\cdot(y_j - y_{av}))\,, \\ &\quad s_y\cdot(-\sin\theta\cdot(x_i - x_{av}) + \cos\theta\cdot(y_j - y_{av}))) \end{aligned} \tag{12}$$

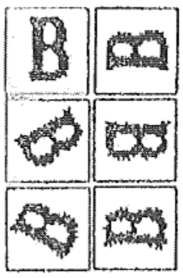
2.3 The Classifier and The Interpreter

The current implementation of the system employs a multilayer feed-forward network for the classifier block. Such a network has a layered structure and only connections between neurons at subsequent layers are permitted. The training algorithm is the widely used *backpropagation algorithm*. Since the input is an image in pixel-map form, the number of nodes in the input layer is fixed and equal to the number of pixels. Further, since the output neurons are organized such that each node represents a class, the number of output neurons is also fixed. An output neuron having a value close to 1 is interpreted as a strong membership, while a value close to 0 will point a loose membership. Hence, given the input and output layer size one should decide on the number of hidden layers and the number of neurons in each hidden layer as well as the learning rate. The general structure of the neural network classifier was presented in [11]. The output of the preprocessor, which is an image in pixel-map form, is converted to a linear array by cascading the rows of the image from the top row to the bottom row. The content of each array entry is the initial input value of the corresponding input node.

The method we have used in the interpreter block is to report *no discrimination* as long as the ratio of the maximum output to the next highest output remains under a predetermined threshold value. When the ratio exceeds the threshold, which means that the maximum output is dominant on the other outputs, the interpreter decides on the class with the maximum output. If not, then the interpreter reports that no unique discrimination could be made. This simple method has been observed to perform well in the evaluation of the classifier outputs.

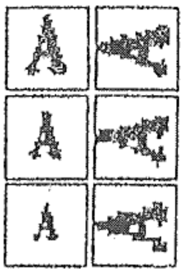


```
Pattern is A with 0.927701
Candidate was Y with 0.021057
Discrimination ratio is 440.6 %

Pattern is A with 0.772268
Candidate was V with 0.053143
Discrimination ratio is 445.3 %

Pattern is A with 0.722949
Candidate was V with 0.029526
Discrimination ratio is 244.8 %
```

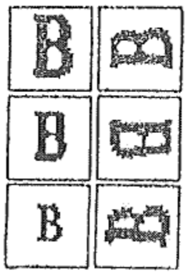


```
Pattern is B with 0.906775
Candidate was D with 0.030358
Discrimination ratio is 298.7 %

Pattern is B with 0.894358
Candidate was D with 0.036271
Discrimination ratio is 246.6 %

Pattern is B with 0.797724
Candidate was E with 0.026937
Discrimination ratio is 296.1 %
```

Figure 1: Classification results with PREP1 for letters A and B rotated by 0, 60, and -60 degrees.

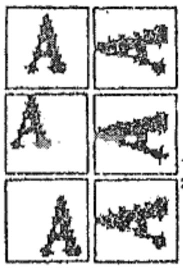


```
Pattern is A with 0.927701
Candidate was Y with 0.021057
Discrimination ratio is 440.6 %

Pattern is A with 0.605121
Candidate was V with 0.040005
Discrimination ratio is 151.3 %

Pattern is A with 0.501417
Candidate was K with 0.168918
Discrimination ratio is 29.7 %
```

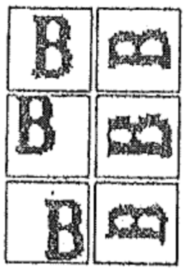


```
Pattern is B with 0.906775
Candidate was D with 0.030358
Discrimination ratio is 298.7 %

Pattern is B with 0.917553
Candidate was D with 0.075526
Discrimination ratio is 121.5 %

Pattern is B with 0.733702
Candidate was E with 0.061952
Discrimination ratio is 118.4 %
```

Figure 2: Classification results with PREP1 for letters A and B scaled by a factor of 1, 0.8, and 0.6.

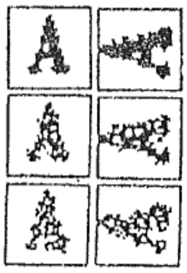


```
Pattern is A with 0.927701
Candidate was Y with 0.021057
Discrimination ratio is 440.6 %

Pattern is A with 0.770125
Candidate was R with 0.041848
Discrimination ratio is 184.0 %

Pattern is A with 0.927701
Candidate was Y with 0.021057
Discrimination ratio is 440.6 %
```

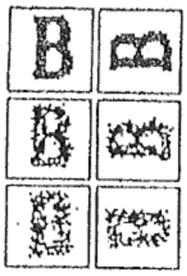


```
Pattern is B with 0.906775
Candidate was D with 0.030358
Discrimination ratio is 298.7 %

Pattern is B with 0.514692
Candidate was F with 0.029781
Discrimination ratio is 172.8 %

Pattern is B with 0.848754
Candidate was D with 0.099396
Discrimination ratio is 85.4 %
```

Figure 3: Classification results with PREP1 for letters A and B translated diagonally by 0, 6, and -6 pixels.

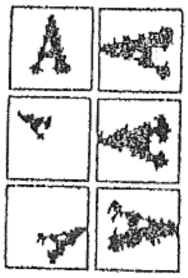


```
Pattern is A with 0.927701
Candidate was Y with 0.021057
Discrimination ratio is 440.6 %

Pattern is A with 0.521594
Candidate was R with 0.035405
Discrimination ratio is 147.3 %

Pattern is F with 0.107739
Candidate was V with 0.047904
Discrimination ratio is 22.5 %
```

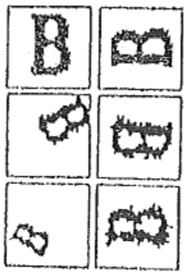


```
Pattern is B with 0.906775
Candidate was D with 0.030358
Discrimination ratio is 298.7 %

Pattern is B with 0.796986
Candidate was E with 0.032866
Discrimination ratio is 242.5 %

Pattern is B with 0.783350
Candidate was R with 0.035924
Discrimination ratio is 218.1 %
```

Figure 4: Classification results with PREP1 for letters A and B with 0%, 20%, and 40% noise.

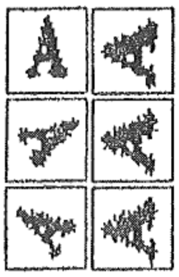


```
Pattern is A with 0.927701
Candidate was Y with 0.021057
Discrimination ratio is 440.6 %

Pattern is A with 0.283534
Candidate was V with 0.081742
Discrimination ratio is 34.7 %

Pattern is A with 0.713344
Candidate was R with 0.023365
Discrimination ratio is 305.3 %
```

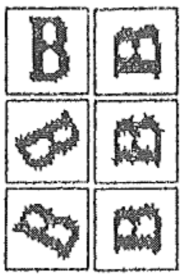


```
Pattern is B with 0.906775
Candidate was D with 0.030358
Discrimination ratio is 298.7 %

Pattern is B with 0.479234
Candidate was R with 0.083784
Discrimination ratio is 57.2 %

Pattern is B with 0.032631
Candidate was R with 0.035072
Discrimination ratio is 237.4 %
```

Figure 5: Classification results with PREP1 for letters A and B with random translation, scaling, and rotation applied.

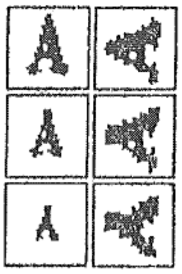


```
Pattern is 1 with 0.911179
Candidate was 22 with 0.046750
Discrimination ratio is 19.5

Pattern is 1 with 0.762654
Candidate was 9 with 0.067812
Discrimination ratio is 11.2

Pattern is 1 with 0.830162
Candidate was 9 with 0.151426
Discrimination ratio is 5.5
```

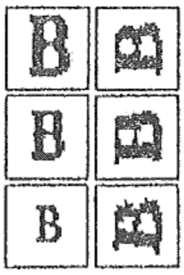


```
Pattern is 2 with 0.779041
Candidate was 19 with 0.072447
Discrimination ratio is 10.8

Pattern is 2 with 0.624327
Candidate was 21 with 0.056638
Discrimination ratio is 11.0

Pattern is 2 with 0.569115
Candidate was 21 with 0.051234
Discrimination ratio is 11.1
```

· Figure 6: Classification results with PREP2 on rotated letters.

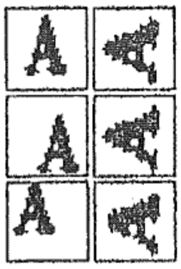


```
Pattern is 1 with 0.911179
Candidate was 22 with 0.046750
Discrimination ratio is 19.5

Pattern is 1 with 0.770966
Candidate was 22 with 0.062864
Discrimination ratio is 12.3

Pattern is 1 with 0.425590
Candidate was 25 with 0.185589
Discrimination ratio is 2.3
```

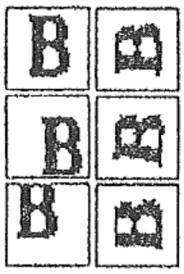


```
Pattern is 2 with 0.779841
Candidate was 19 with 0.072447
Discrimination ratio is 10.8

Pattern is 2 with 0.161977
Candidate was 4 with 0.078825
Discrimination ratio is 2.1

Pattern is 2 with 0.634949
Candidate was 18 with 0.071779
Discrimination ratio is 8.8
```

Figure 7: Classification results with PREP2 on scaled letters.

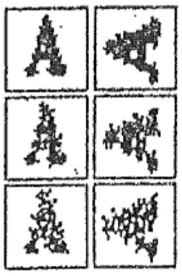


```
Pattern is 1 with 0.911179
Candidate was 22 with 0.046750
Discrimination ratio is 19.5

Pattern is 1 with 0.807283
Candidate was 25 with 0.077792
Discrimination ratio is 10.4

Pattern.is 1 with 0.781698
Candidate was 9 with 0.074611
Discrimination ratio is 10.5
```

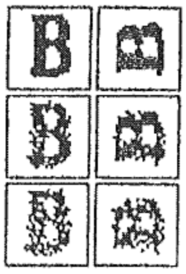


```
Pattern is 2 with 0.779841
Candidate was 19 with 0.072447
Discrimination ratio is 10.8

Pattern is 2 with 0.426662
Candidate was 18 with 0.191974
Discrimination ratio is 2.2

Pattern is 2 with 0.496783
Candidate was 19 with 0.050108
Discrimination ratio is 9.9
```

Figure 8: Classification results with PREP2 on translated letters.

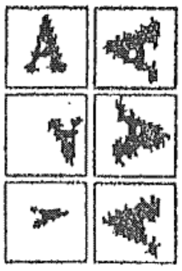


```
Pattern is 1 with 0.911179
Candidate was 22 with 0.046750
Discrimination ratio is 19.5

Pattern is 1 with 0.721102
Candidate was 25 with 0.119924
Discrimination ratio is 6.0

Pattern is 1 with 0.775800
Candidate was 25 with 0.131237
Discrimination ratio is 5.9
```

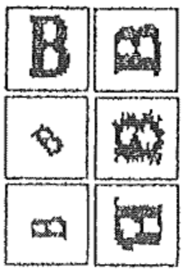


```
Pattern is 2 with 0.779841
Candidate was 19 with 0.072447
Discrimination ratio is 10.8

Pattern is 2 with 0.689617
Candidate was 18 with 0.058586
Discrimination ratio is 11.8

Pattern is 2 with 0.105599
Candidate was 4 with 0.052336
Discrimination ratio is 2.0
```

Figure 9: Classification results with PREP2 on noisy letters.

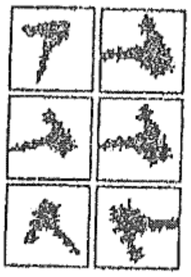


```
Pattern is 1 with 0.911179
Candidate was 22 with 0.046750
Discrimination ratio is 19.5

Pattern is 1 with 0.353124
Candidate was 25 with 0.109775
Discrimination ratio is 3.2

Pattern is 1 with 0.750400
Candidate was 25 with 0.078528
Discrimination ratio is 9.6
```

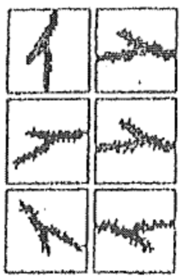


```
Pattern is 2 with 0.779841
Candidate was 19 with 0.072447
Discrimination ratio is 10.8

Pattern is 2 with 0.304866
Candidate was 4 with 0.141030
Discrimination ratio is 2.2

Pattern is 2 with 0.205307
Candidate was 18 with 0.099175
Discrimination ratio is 2.9
```

Figure 10: Classification results with PREP2 on letters with random translation, scaling, and rotation applied.

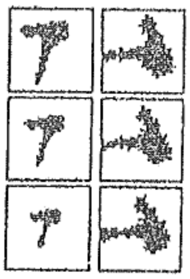


```
Pattern is 1 with 0.820121
Candidate was 14 with 0.075892
Discrimination ratio is 10.8

Pattern is 1 with 0.554672
Candidate was 34 with 0.054674
Discrimination ratio is 10.1

Pattern is 1 with 0.580281
Candidate was 5 with 0.071503
Discrimination ratio is 8.1
```

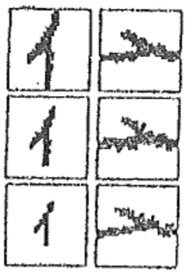


```
Pattern is 2 with 0.902589
Candidate was 13 with 0.048572
Discrimination ratio is 18.6

Pattern is 2 with 0.565567
Candidate was 33 with 0.072772
Discrimination ratio is 7.8

Pattern is 2 with 0.469407
Candidate was 40 with 0.017460
Discrimination ratio is 26.9
```

Figure 11: Classification results with PREP1 for symbols from Class 1 and 2 rotated by 0, 60, and -60 degrees.

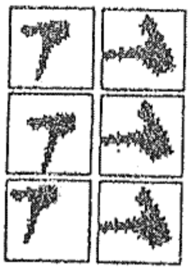


```
Pattern is 1 with 0.820121
Candidate was 14 with 0.075892
Discrimination ratio is 10.8

Pattern is 1 with 0.486271
Candidate was 34 with 0.042003
Discrimination ratio is 11.6

Pattern is 1 with 0.525178
Candidate was 34 with 0.077539
Discrimination ratio is 6.8
```

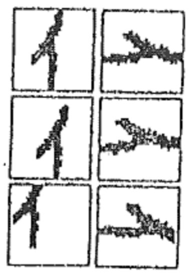


```
Pattern is 2 with 0.902589
Candidate was 13 with 0.048572
Discrimination ratio is 18.6

Pattern is 2 with 0.446612
Candidate was 40 with 0.203349
Discrimination ratio is 2.2

Pattern is 2 with 0.812504
Candidate was 33 with 0.035828
Discrimination ratio is 22.7
```

Figure 12: Classification results with PREP1 for symbols from Class 1 and 2 scaled by a factor of 1, 0.8, and 0.6.

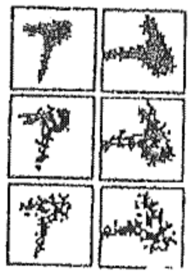


```
Pattern is 1 with 0.820121
Candidate was 14 with 0.075892
Discrimination ratio is 10.8

Pattern is 1 with 0.801266
Candidate was 14 with 0.066163
Discrimination ratio is 12.1

Pattern is 1 with 0.813754
Candidate was 14 with 0.074115
Discrimination ratio is 11.0
```

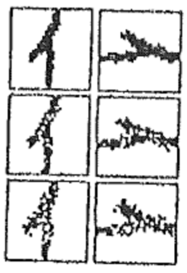


```
Pattern is 2 with 0.902589
Candidate was 13 with 0.048572
Discrimination ratio is 18.6

Pattern is 2 with 0.455013
Candidate was 35 with 0.069810
Discrimination ratio is 6.5

Pattern is 40 with 0.249899
Candidate was 9 with 0.099151
Discrimination ratio is 2.5
```

Figure 13: Classification results with PREP1 for symbols from Class 1 and 2 translated diagonally by 0, 6, and -6 pixels.

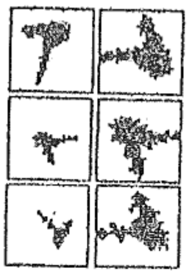


```
Pattern is 1 with 0.820121
Candidate was 14 with 0.075892
Discrimination ratio is 10.8

Pattern is 1 with 0.786378
Candidate was 14 with 0.054121
Discrimination ratio is 14.5

Pattern is 1 with 0.679368
Candidate was 21 with 0.038395
Discrimination ratio is 17.7
```

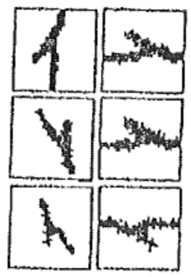


```
Pattern is 2 with 0.902589
Candidate was 13 with 0.048572
Discrimination ratio is 18.6

Pattern is 2 with 0.714926
Candidate was 40 with 0.069100
Discrimination ratio is 10.3

Pattern is 2 with 0.265836
Candidate was 6 with 0.080215
Discrimination ratio is 3.3
```

Figure 14: Classification results with PREP1 for symbols from Class 1 and 2 with 0%, 20%, and 40% noise.

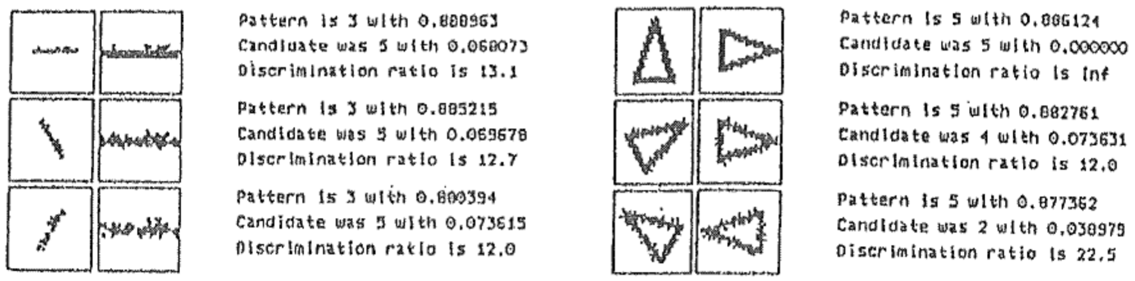


```
Pattern is 1 with 0.820121
Candidate was 14 with 0.075892
Discrimination ratio is 10.8

Pattern is 1 with 0.640245
Candidate was 34 with 0.052577
Discrimination ratio is 12.2

Pattern is 1 with 0.463797
Candidate was 13 with 0.140003
Discrimination ratio is 3.3
```



```
Pattern is 2 with 0.902589
Candidate was 13 with 0.048572
Discrimination ratio is 18.6

Pattern is 2 with 0.572148
Candidate was 13 with 0.103430
Discrimination ratio is 5.5

Pattern is 2 with 0.758859
Candidate was 40 with 0.109365
Discrimination ratio is 6.9
```

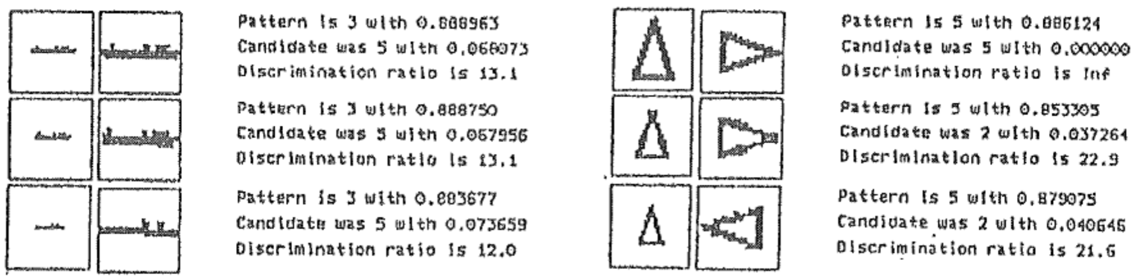
Figure 15: Classification results with PREP1 for symbols from Class 1 and 2 with random translation, scaling, and rotation applied.



```
Pattern is 3 with 0.888963
Candiuate was 5 with 0.068073
Discrimination ratio is 13.1

Pattern is 3 with 0.885215
Candidate was 5 with 0.069678
Discrimination ratio is 12.7

Pattern is 3 with 0.880394
Candidate was 5 with 0.073615
Discrimination ratio is 12.0
```

```
Pattern is 5 with 0.886124
Candidate was 5 with 0.000000
Discrimination ratio is Inf

Pattern is 5 with 0.882761
Candidate was 4 with 0.073631
Discrimination ratio is 12.0

Pattern is 5 with 0.877362
Candidate was 2 with 0.038979
Discrimination ratio is 22.5
```

Figure 16: Classification results with PREP1 for geometric symbols rotated by 0, 60, and -60 degrees.



```
Pattern is 3 with 0.888963
Candidate was 5 with 0.068073
Discrimination ratio is 13.1

Pattern is 3 with 0.888750
Candidate was 5 with 0.067956
Discrimination ratio is 13.1

Pattern is 3 with 0.883677
Candidate was 5 with 0.073659
Discrimination ratio is 12.0
```

```
Pattern is 5 with 0.886124
Candidate was 5 with 0.000000
Discrimination ratio is Inf

Pattern is 5 with 0.853305
Candidate was 2 with 0.037264
Discrimination ratio is 22.9

Pattern is 5 with 0.879075
Candidate was 2 with 0.040645
Discrimination ratio is 21.6
```

Figure 17: Classification results with PREP1 for geometric symbols scaled by a factor of 1, 0.8, and 0.6.

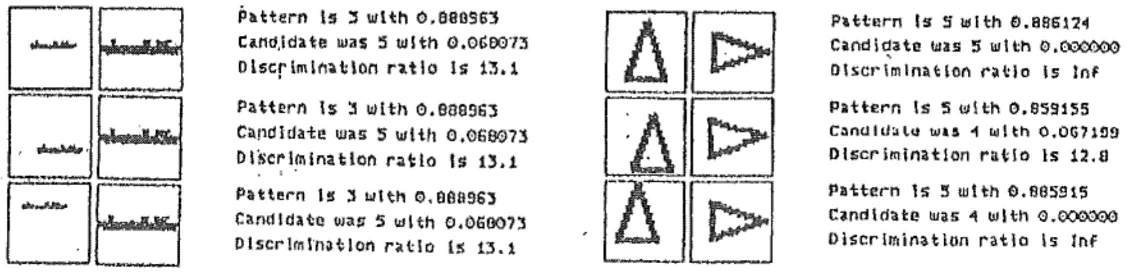
3 Experimental Results

Three problem domains have been experimented: character recognition on the English Alphabet, character recognition on the Japanese Katakana Alphabet and recognition of geometric objects. It should be noted that the R-Block rotates a pattern until the computed orientation coincides with the *x-axis*. Since a pattern and its 180° rotated version will have the same orientation of maximal variance, R-Block will not be able to differentiate between them and will apply the same angle of rotation on both patterns. The resulting mappings will conserve this 180° angle difference. Hence depending on its original orientation, a given pattern will be mapped to one of the two canonical patterns. These two canonical patterns will both represent the class. Hence for each original pattern we have two preprocessor outputs which are used during training.

3.1 Character Recognition on the English Alphabet

This classical problem is the classification of letters in the English alphabet. The number of hidden layers and the number of neurons in each hidden layer has been found by trial-and-error, which is typical for most multilayer feed-forward network applications [1, 2, 3, 6, 8, 10]. From experimentation, a network with 1024 input nodes, 20 neurons in a single hidden layer, and 26 neurons in the output layer performs best for a number of test cases. In the training phase, the network is trained on the canonical example patterns (which are the outputs of the preprocessor) until it manages to successfully classify the letters.

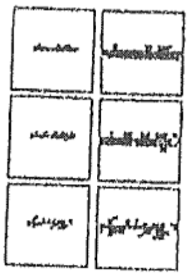
Figure 1 through Figure 10 present examples of the system performance using both preprocessors. Input images images are 32×32 pixels. First column is the original image given to the system. Second column is the preprocessed version of the original image, and finally third column is the resulting decision. The class name and value of the corresponding output neuron are given.



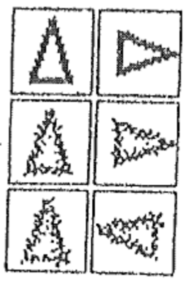
```
Pattern is 3 with 0.888963
Candidate was 5 with 0.068073
Discrimination ratio is 13.1

Pattern is 3 with 0.888963
Candidate was 5 with 0.068073
Discrimination ratio is 13.1

Pattern is 3 with 0.888963
Candidate was 5 with 0.068073
Discrimination ratio is 13.1
```

```
Pattern is 5 with 0.886124
Candidate was 5 with 0.000000
Discrimination ratio is Inf

Pattern is 5 with 0.859155
Candidate was 4 with 0.067199
Discrimination ratio is 12.8

Pattern is 5 with 0.885915
Candidate was 4 with 0.000000
Discrimination ratio is Inf
```

Figure 18: Classification results with PREP1 for geometric symbols translated diagonally by 0, 6, and -6 pixels.

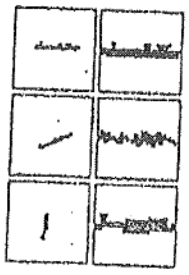


```
Pattern is 3 with 0.888963
Candidate was 5 with 0.068073
Discrimination ratio is 13.1

Pattern is 3 with 0.883824
Candidate was 5 with 0.074029
Discrimination ratio is 11.9

Pattern is 3 with 0.878547
Candidate was 5 with 0.075138
Discrimination ratio is 11.7
```

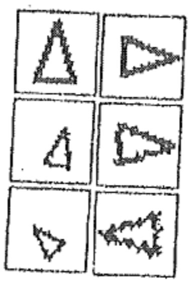


```
Pattern is 5 with 0.886124
Candidate was 5 with 0.000000
Discrimination ratio is Inf

Pattern is 5 with 0.882583
Candidate was 5 with 0.000000
Discrimination ratio is Inf

Pattern is 5 with 0.838724
Candidate was 4 with 0.062111
Discrimination ratio is 13.5
```

Figure 19: Classification results with PREP1 for geometric symbols with 0%, 20%, and 40% noise.



```
Pattern is 3 with 0.888963
Candidate was 5 with 0.068073
Discrimination ratio is 13.1

Pattern is 3 with 0.885966
Candidate was 5 with 0.071690
Discrimination ratio is 12.3

Pattern is 3 with 0.883736
Candidate was 5 with 0.075221
Discrimination ratio is 11.7
```



```
Pattern is 5 with 0.886124
Candidate was 5 with 0.000000
Discrimination ratio is Inf

Pattern is 5 with 0.859256
Candidate was 4 with 0.060094
Discrimination ratio is 14.3

Pattern is 5 with 0.870000
Candidate was 4 with 0.069743
Discrimination ratio is 12.5
```

Figure 20: Classification results with PREP1 for geometric symbols with random translation, scaling, and rotation applied.

3.2 Character Recognition on the Japanese Katakana Alphabet

This problem is the classification of symbols in the Japanese Katakana alphabet. Since the 111 Katakana symbols are in fact combined forms of 66 unique patterns, the system is trained only on these 66 patterns. With some experimentation, a network with only one hidden layer having twenty nodes has been chosen. Figures 11 through 15 give the performance of the system with preprocessor PREP1.

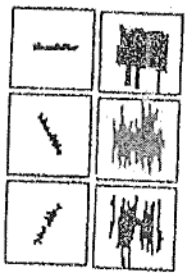
3.3 Classification of Geometric Symbols

This is the problem of classification of five main geometric symbols: circle, cross, line, rectangle, and triangle. Figure 16 through Figure 25 present examples of the system performance using both preprocessors.

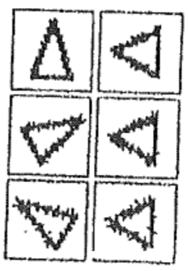
Figure 26 gives the classification results for the two versions of the preprocessor on the distorted versions of the geometric symbols. PREP1 could detect only 50% of the distorted patterns, while PREP2 managed to successfully classify 90%. The performance difference emerges from the axial scaling correction ability of this preprocessor.

Table 1 shows the (average) percentage of English letters, Katakana symbols and geometric symbols correctly classified after undergoing 100 random transformations of the type stated in the first column.[2]

[2]Combined denotes an input that is distorted from the original by a random rotation, scaling and translation. The noise is applied to an undeformed pattern by flipping the on pixels with a certain probability. In the last row, noise is applied to the distorted pattern

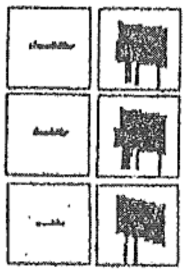


```
Pattern is 3 with 0.871819
Candidate was 5 with 0.031803
Discrimination ratio is 27.4

Pattern is 3 with 0.800754
Candidate was 5 with 0.034429
Discrimination ratio is 23.3

Pattern is 3 with 0.755228
Candidate was 5 with 0.334764
Discrimination ratio is 2.3
```

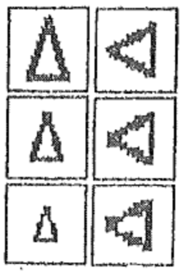


```
Pattern is 5 with 0.589079
Candidate was 3 with 0.041384
Discrimination ratio is 14.3

Pattern is 5 with 0.580918
Candidate was 3 with 0.046978
Discrimination ratio is 12.4

Pattern is 5 with 0.549709
Candidate was 3 with 0.053978
Discrimination ratio is 10.2
```

Figure 21: Classification results with PREP2 for rotated geometric symbols.

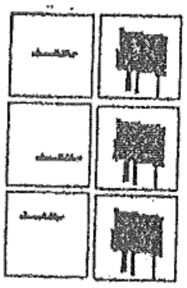


```
Pattern is 3 with 0.871819
Candidate was 5 with 0.031803
Discrimination ratio is 27.4

Pattern is 3 with 0.864028
Candidate was 5 with 0.032874
Discrimination ratio is 26.3

Pattern is 3 with 0.860065
Candidate was 5 with 0.032336
Discrimination ratio is 26.8
```

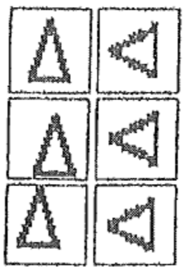


```
Pattern is 5 with 0.589079
Candidate was 3 with 0.041384
Discrimination ratio is 14.3

Pattern is 5 with 0.551313
Candidate was 3 with 0.048864
Discrimination ratio is 11.3

Pattern is 5 with 0.569154
Candidate was 3 with 0.041707
Discrimination ratio is 13.6
```

Figure 22: Classification results with PREP2 for scaled geometric symbols.



```
Pattern is 3 with 0.871819
Candidate was 5 with 0.031803
Discrimination ratio is 27.4

Pattern is 3 with 0.871819
Candidate was 5 with 0.031803
Discrimination ratio is 27.4

Pattern is 3 with 0.871819
Candidate was 5 with 0.031803
Discrimination ratio is 27.4
```

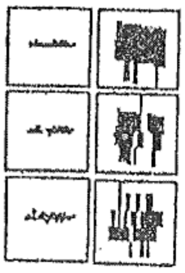


```
Pattern is 5 with 0.589079
Candidate was 3 with 0.041384
Discrimination ratio is 14.3

Pattern is 5 with 0.494082
Candidate was 3 with 0.054646
Discrimination ratio is 9.0

Pattern is 5 with 0.589624
Candidate was 3 with 0.041387
Discrimination ratio is 14.2
```

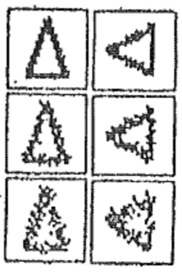
Figure 23: Classification results with PREP2 for translated geometric symbols.

4 Conclusions

In this work we have presented a hybrid a pattern classification system which can classify patterns independent of any deformations of translation, scaling and rotation. The system uses a preprocessor which maps input patterns to a set of canonical patterns which is then classified by a multilayer neural network. The artificial neural network is trained using the popular backpropagation network. The use of the preprocessor reduces the number of training patterns to two per example pattern to be classified instead of a much larger number if the networks were to be trained on all possible distortions of the patterns. Results from three different applications were presented. In classification of letters of the English Alphabet the system was able to correctly classify 89% of the inputs which were deformed by random rotations, translations and scaling. The performance is much better when distortions were only of one kind, with 100% of the inputs distorted by only translations correctly classified. In the recognition of geometric figures, the system was able to correctly classify 88% of the inputs which were deformed by all three kinds of distortions while the performance was almost perfect when the random distortions were only of one kind. The overall performance for recognition of Japanese Katakana alphabet were worse compared to the other two applications. 68% of the inputs with combined deformations were correctly classified. Even though the performance for inputs which are distorted only by translations or scaling is very good, the performance for rotationally distorted inputs was about 75%. The main reason for this loss of performance is that some of the letters in this case are very similar to each other differing by very small visual features. When patterns that are already deformed are processed with the preprocessors, a certain amount of superfluous visual features may be introduced during mapping between images. The system presented here is independent of the application domain and leaves feature extraction to the neural network and thus can be applied in other domains with relative simplicity.

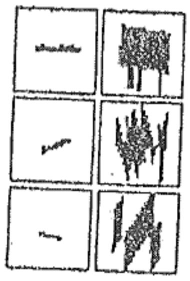


```
Pattern is 3 with 0.871819
Candidate was 5 with 0.031803
Discrimination ratio is 27.4

Pattern is 3 with 0.802999
Candidate was 5 with 0.034289
Discrimination ratio is 23.4

Pattern is 3 with 0.822618
Candidate was 5 with 0.055444
Discrimination ratio is 14.8
```

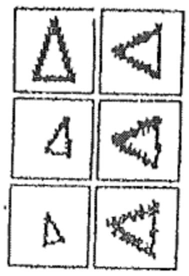


```
Pattern is 5 with 0.589879
Candidate was 3 with 0.041384
Discrimination ratio is 14.3

Pattern is 5 with 0.527329
Candidate was 3 with 0.050612
Discrimination ratio is 10.4

Pattern is 5 with 0.525596
Candidate was 3 with 0.072024
Discrimination ratio is 7.3
```

Figure 24: Classification results with PREP2 for noisy geometric symbols.



```
Pattern is 3 with 0.871819
Candidate was 5 with 0.031803
Discrimination ratio is 27.4

Pattern is 3 with 0.851932
Candidate was 5 with 0.032825
Discrimination ratio is 26.0

Pattern is 3 with 0.865563
Candidate was 5 with 0.029263
Discrimination ratio is 29.6
```

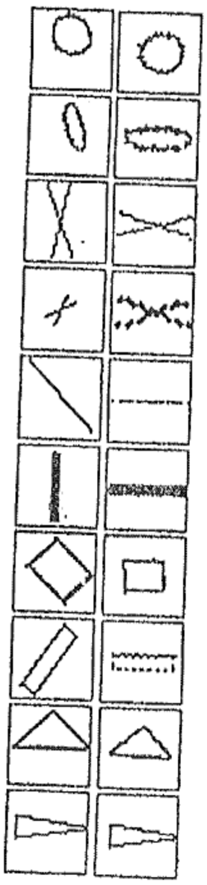


```
Pattern is 5 with 0.589879
Candidate was 3 with 0.041384
Discrimination ratio is 14.3

Pattern is 5 with 0.522750
Candidate was 3 with 0.061037
Discrimination ratio is 8.6

Pattern is 5 with 0.568493
Candidate was 3 with 0.054351
Discrimination ratio is 10.5
```

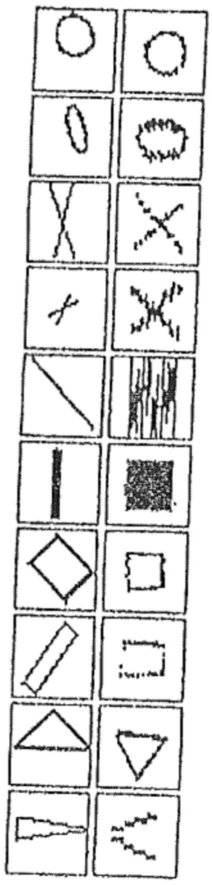
Figure 25: Classification results with PREP2 for geometric symbols with random translation, scaling, and rotation applied.

	English		Katakana	Geometric	
Transformation	PREP1	PREP2	PREP1	PREP1	PREP2
Rotation	91%	89%	75%	98%	94%
Scaling	98%	94%	92%	100%	100%
Translation	100%	100%	100%	100%	100%
Combined	89%	79%	68%	88%	84%
20% noise	98%	96%	93%	100%	100%
40% noise	92%	84%	76%	98%	97%
Combined & 20% noise	77%	60%	57%	89%	78%

Table 1: Percentage of correct classification for English letters, Katakana symbols and geometric symbols under various distortions



```
Pattern is 1 with 0.489718
Candidate was 5 with 0.182937
Discrimination ratio is 2.7

Pattern is 5 with 0.834490
Candidate was 4 with 0.054007
Discrimination ratio is 15.5

Pattern is 3 with 0.683376
Candidate was 5 with 0.106999
Discrimination ratio is 6.4

Pattern is 3 with 0.709030
Candidate was 5 with 0.129486
Discrimination ratio is 5.5

Pattern is 3 with 0.563266
Candidate was 5 with 0.066862
Discrimination ratio is 8.4

Pattern is 3 with 0.888596
Candidate was 5 with 0.068497
Discrimination ratio is 13.0

Pattern is 4 with 0.351126
Candidate was 5 with 0.070720
Discrimination ratio is 5.0

Pattern is 5 with 0.487965
Candidate was 4 with 0.006454
Discrimination ratio is 75.6

Pattern is 5 with 0.419376
Candidate was 3 with 0.104784
Discrimination ratio is 4.0

Pattern is 3 with 0.714873
Candidate was 5 with 0.286256
Discrimination ratio is 2.5
```



```
Pattern is 1 with 0.829573
Candidate was 4 with 0.106937
Discrimination ratio is 7.8

Pattern is 1 with 0.746327
Candidate was 5 with 0.080175
Discrimination ratio is 9.3

Pattern is 2 with 0.807162
Candidate was 3 with 0.208715
Discrimination ratio is 3.9

Pattern is 2 with 0.660773
Candidate was 3 with 0.200848
Discrimination ratio is 3.3

Pattern is 3 with 0.717909
Candidate was 5 with 0.290264
Discrimination ratio is 2.5

Pattern is 3 with 0.757173
Candidate was 5 with 0.045864
Discrimination ratio is 16.5

Pattern is 4 with 0.451725
Candidate was 2 with 0.078806
Discrimination ratio is 5.7

Pattern is 4 with 0.552510
Candidate was 3 with 0.028043
Discrimination ratio is 19.7

Could not discriminate ...
Candidates were 5 with 0.236333
                and 3 with 0.130853

Pattern is 5 with 0.456683
Candidate was 3 with 0.067811
Discrimination ratio is 6.7
```

Figure 26: Classification results, with PREP1 (left) and PREP2 (right), on distorted patterns of the five main geometric symbols.

References

[1] E. Barnard and D. Casasent, "Shift invariance and the neocognitron," *Neural Networks*, vol. 3, pp. 403–410, 1990.

[2] Y. Le Cun et al., "Handwritten digit recognition: Applications of neural network chips and automatic learning," *IEEE Communications Magazine*, pp. 41–46, November 1989.

[3] Y. Le Cun et al., "Handwritten zip code recognition with multilayer networks," in *Proceedings of 10th International Conference on Pattern Recognition, Atlantic City, NJ USA*, pp. 35–40. IEEE Computer Society Press, Los Alamitos, 1990.

[4] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* John Wiley and Sons, NY, 1973.

[5] K. Fukunaga, *Introduction to Statistical Pattern Recognition.* Academic Press, NY, 1972.

[6] K. Fukushima, S. Miyake, and T. Ito, "Neocognitron: A neural network model for a mechanism of visual pattern recognition," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-13, pp. 826–834, 1983.

[7] R. C. Gonzales and P. Wintz, *Digital Image Processing*, pp. 122–130. Addison-Wesley, MA, 1987.

[8] A. Khotanzad and J. Lu, "Classification of invariant image representations using a neural network," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, number 6, pp. 1028–1038, June 1990.

[9] M. Kirby and L. Sirovich, "Application of the karhunen-loéve procedures for the characterization of human faces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, number 1, pp. 103–108, January 1990.

[10] H. A. Malki and A. Moghaddamjoo, "Using the karhunen-loéve transformation in the back-propagation training algorithm," *IEEE Transactions on Neural Networks*, vol. 2, number 1, pp. 162–165, January 1991.

[11] C. Yüceer and K. Oflazer, "A rotation, scaling, and translation invariant pattern classification system," in Mehmet Baray and Bülent Özgüç, eds., *Proceedings of ISCIS VI, The Sixth International Symposium on Computer and Information Science, Side, Türkiye*, volume 2, pp. 859–869, October 1991.