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 presents a hybrid pattern classification system which can classify patterns in a rotation, scaling, and translation invariant manner. The system is based on preprocessing the input image to map it into a rotation, scaling, and translation invariant canonical form, then classifying the preprocessed image and finally interpreting the classification results. Results from a number of classification problems are also presented in the paper.

1 Introduction

In this work, a pattern classification system based on a pattern preprocessor that is rotation, scaling and translation invariant, and an artificial neural network classifier is described. After a brief description of the system architecture we provide results from some sample pattern sets. The system can recognize rotated, scaled and translated patterns correctly in about 90% of the cases and it also has a reasonable noise tolerance. The details of our system can be found in [5, 6]. Some other approaches to the same problem can be found in [1, 2, 3, 4].

2 The Pattern Classification System

Our system consists of three main blocks as depicted in top portion of Figure 1. These blocks are the *preprocessor*, the *classifier* and the *interpreter*. The *preprocessor* has three cascaded blocks as shown in bottom portion of Figure 1. 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 *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. Finally the *R-block* maintains rotational invariancy by rotating the image so that the direction of maximum variance for on-pixels coincides with the *x-axis* of the image. The derivation of the function for this unit is based on the Karhunen-Loéve transformation and is given in [5, 6]. Here only brief formulas are supplied in Appendix A.

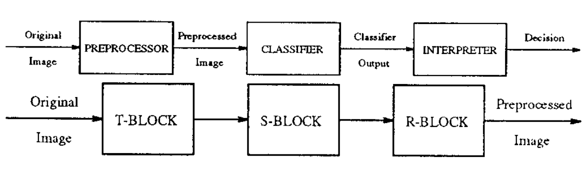


Figure 1: Block diagrams of RST (top) and the preprocessor (bottom).

The *classifier* consists of a *multilayer feed-forward network* and the training algorithm is the widely used *backpropagation* algorithm. The number of nodes in the input layer and the output layer is fixed and equal to the number of pixels in the preprocessor output image and number of pattern classes respectively. Note that the preprocessor output image is directly fed into the neural network classifier.

The *interpreter* block interprets the outputs of the neural network output layer. This block decides on a class if the ratio of the maximum output to the next highest output remains over a predetermined threshold value. Thus even if the activation value for a class is not high but it is sufficiently higher than the nearest one, the interpreter will make a decision.

3 Results from Two Problems

The first problem is the classification of 26 printed characters in the English alphabet. The second is the classification of the 5 geometric symbol patterns for *circle*, *cross*, *line*, *rectangle*, and *triangle*. The number of hidden layers and the number of neurons in each hidden layer is found by experimentation [3, 4]. A single hidden layer having 20 neurons has been chosen for the first application and a single hidden layer with 3 neurons has been used for the second application. In the training phase, the networks are trained on the preprocessor outputs for the input example patterns. Note that since R-Block rotates a pattern so that the maximum variance direction coincides with the *x-axis*, any pattern and a 180° rotated version will have the same orientation for maximum variance. R-Block rotates both patterns by the same amount. The resulting preprocessor outputs which are canonical patterns will differ from each other by a 180° rotation. Thus the classifier is trained with two example patterns for each pattern class, the two being 180° rotated versions

0-8186-2915-0/92 $3.00 © 1992 IEEE

of each other. Hence the two applications use 52 and 10 training patterns respectively.

Figure 2 through Figure 5 show the output of the system in classifying various distorted versions of the letters A and B and the *triangle* and *line* symbols.

For each case, the first image is the 32 by 32 pixel original input image, the second image is the output of the preprocessor and hence the input to the classifier. Third column shows the activation values for the two competing outputs and the decision of the interpreter.[1]

Due to space limitations we present only a limited set of results from two problems. The complete set of results from our four classification problems are in [5].

4 Conclusions

We have presented the high-level architecture of a hybrid pattern classification system which relies on a preprocessor that maps an input pattern image into a canonical form which is then classified by a multilayer neural network. The system can successfully recognize distorted patterns in 90% of the test cases. It also has a reasonable classification performance even in the presence of noise.

A Basic Formulae

Define function $f(x,y)$ to give 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. The terms used and the mapping functions will be:

$$x_{av} = \frac{1}{\sum_{i=1}^{N}\sum_{j=1}^{N} f(x_i,y_j)} \sum_{i=1}^{N}\sum_{j=1}^{N} f(x_i,y_j)\cdot x_i \quad (1)$$

$$y_{av} = \frac{1}{\sum_{i=1}^{N}\sum_{j=1}^{N} f(x_i,y_j)} \sum_{i=1}^{N}\sum_{j=1}^{N} f(x_i,y_j)\cdot y_j \quad (2)$$

$$f_T(x_i,y_j) = f(x_i - x_{av},\, y_j - y_{av}) \quad (3)$$

$$r_{av} = \frac{1}{\sum_{i=1}^{N}\sum_{j=1}^{N} f_T(x_i,y_j)} \sum_{i=1}^{N}\sum_{j=1}^{N} f_T(x_i,y_j)\cdot\sqrt{x_i^2+y_j^2} \quad (4)$$

$$f_{TS}(x_i,y_j) = f_T(\frac{R}{r_{av}}\cdot x_i,\, \frac{R}{r_{av}}\cdot y_j) \quad (5)$$

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

$$\sin\theta = \frac{(T_{yy}-T_{xx})+\sqrt{(T_{yy}-T_{xx})^2+4\cdot T_{xy}^2}}{\sqrt{2\cdot\left[\sqrt{(T_{yy}-T_{xx})^2+4\cdot T_{xy}^2}+(T_{yy}-T_{xx})\right]\sqrt{(T_{yy}-T_{xx})^2+4\cdot T_{xy}^2}}} \quad (7)$$

$$T_{xx} = \sum_{i=1}^{N}\sum_{j=1}^{N} f_{TS}(x_i,y_j)\cdot x_i^2 \quad (8)$$

$$T_{yy} = \sum_{i=1}^{N}\sum_{j=1}^{N} f_{TS}(x_i,y_j)\cdot y_j^2 \quad (9)$$

$$T_{xy} = \sum_{i=1}^{N}\sum_{j=1}^{N} f_{TS}(x_i,y_j)\cdot x_i\cdot y_j \quad (10)$$

$$f_{TSR}(x_i,y_j) = f_{TS}(\cos\theta\cdot x_i+\sin\theta\cdot y_j,\, -\sin\theta\cdot x_i+\cos\theta\cdot y_j) \quad (11)$$

References

[1] 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.

[2] Y. N. Hsu, H. H. Arsenault, and G. April, "Rotational invariant digital pattern recognition using circular harmonic expansion," *Applied Optics*, vol. 21, pp. 4012–4015, 1982.

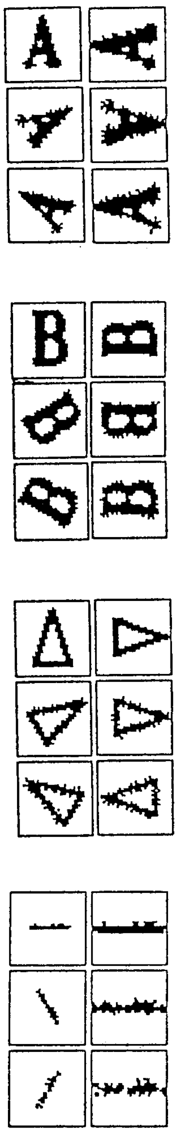
[3] A. Khotanzad and J. Lu, "Classification of invariant image representations using a neural network," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, number 6, pp. 1028–1038, June 1990.

[4] G. L. Martin and J. A. Pittman, "Recognizing hand-printed letters and digits using backpropagation learning," *Neural Computation*, vol. , number 3, pp. 258–267, 1991.

[5] C. Yüceer, "A general purpose rotation, scaling, and translation invariant pattern classification system," Master's thesis, Computer Engineering and Information Science Department, Bilkent University, Ankara, Türkiye, January 1992.

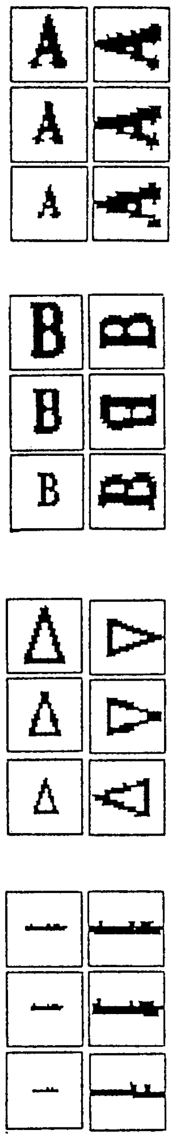
[6] C. Yüceer and K. Oflazer, "A rotation, scaling, and translation invariant pattern classification system," in Mehmet Baray and Bülent Özgüç, eds., *Proceedings of ISCIS VI, The Sixth International Symposium on Computer and Information Science, Side, Türkiye*, pp. 859–869, 1991.

[1] The *triangle* and *line* symbols are named as class 5 and 3 respectively.



```
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 145.3 %

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 %

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

Pattern is 3 with 0.888963
Candidate 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
```

Figure 2: Classification results for rotated versions of letters A and B and the *triangle* and *line* symbols.



```
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 %

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 0.040646
Discrimination ratio is 21.6

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
```

Figure 3: Classification results for scaled versions of letters A and B and the *triangle* and *line* symbols.

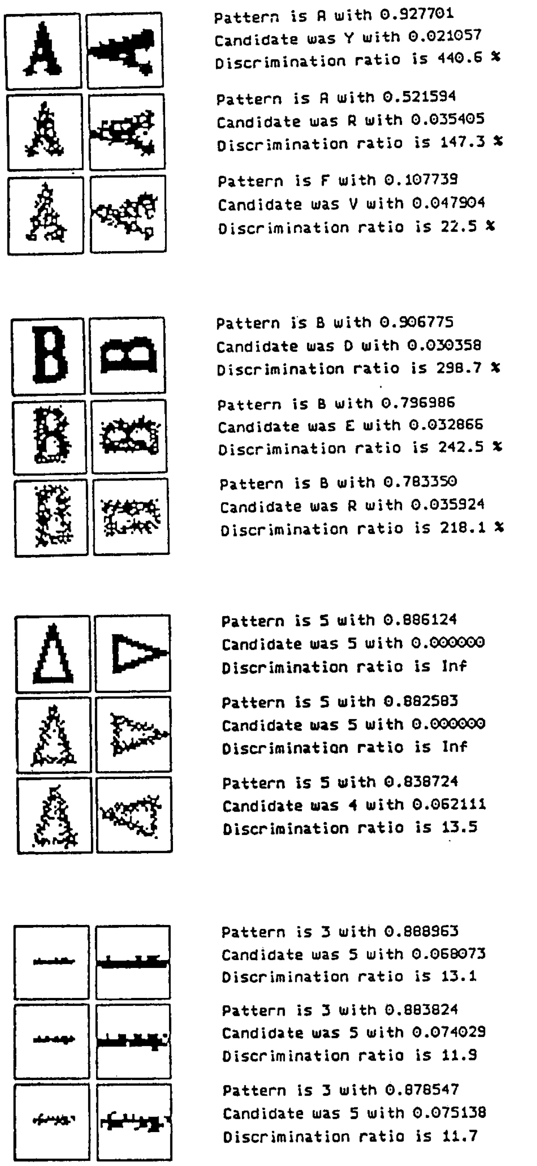


Figure 4: Classification results for noisy versions of letters A and B and the *triangle* and *line* symbols.

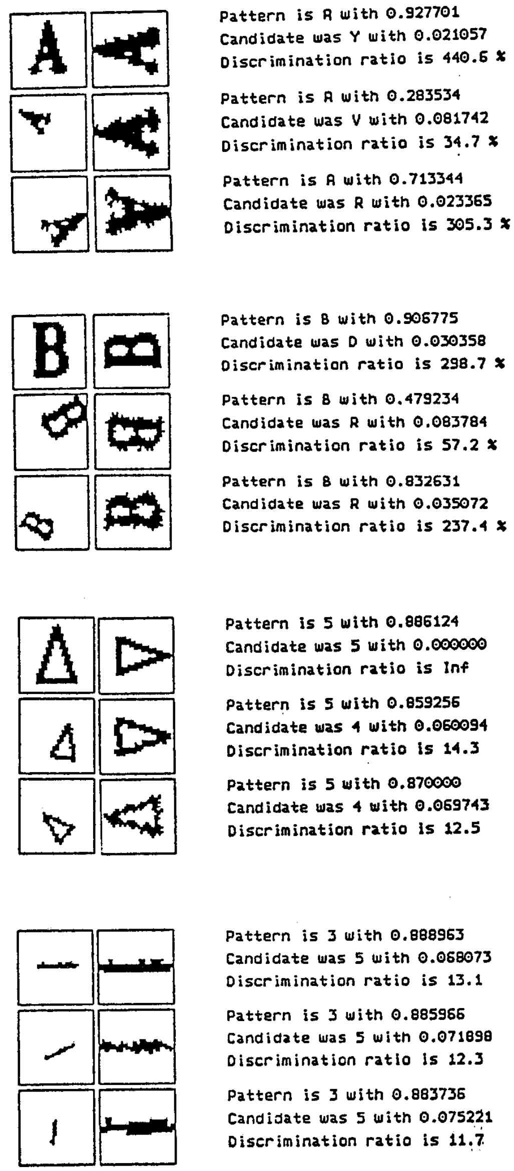


Figure 5: Classification results for randomly transformed versions of letters A and B and the *triangle* and *line* symbols.