NISTIR 4805

FAUST: A Vision-Based Neural Network Multi-Map Pattern Recognition Architecture

C. L. Wilson

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards
and Technology
Technology Administration
Gaithersburg, MD 20899

NIST

NISTIR 4805

FAUST: A Vision-Based Neural Network Multi-Map Pattern Recognition Architecture

C. L. Wilson

U.S. DEPARTMENT OF COMMERCE
Technology Administration
National Institute of Standards
and Technology
Technology Administration
Gaithersburg, MD 20899

March 1992

U.S. DEPARTMENT OF COMMERCE
Barbara Hackman Franklin, Secretary

TECHNOLOGY ADMINISTRATION
Robert M. White, Under Secretary for Technology

NATIONAL INSTITUTE OF STANDARDS
AND TECHNOLOGY
John W. Lyons, Director

FAUST: A Vision-Based Neural Network Multi-Map Pattern Recognition Architecture

C. L. Wilson
National Institute of Standards and Technology, Gaithersburg, MD 20899

Abstract

A new architecture is presented for multi-map, self-organizing pattern recognition which allows concurrent massively parallel learning of features using different maps for each feature type. The method used is similar to the multi-map structures known to exist in the vertebrate sensory cortex. The learning used to update memory locations uses a feed-forward mechanism and is self-organizing. The architecture is described by the acronym FAUST (Feed-forward Association Using Symmetrical Triggering). As a demonstration of the effectiveness of FAUST, a character recognition, fingerprint classification, and forms recognition programs have been constructed on a massively parallel compute. The character recognition program can perform 99% accurate character recognition on medium-quality machine printed digits at a speed of 2.4ms/digit, and 88% recognition on multiple-writer hand print with a 2.3% substitutional error rate. The form recognition program can achieve 94% accuracy on complex forms. The fingerprint classification program classifies 93% of fingerprints correctly with 10% rejection rate. All of the calculations were performed on an Active Memory Technology DAP 510 [1].

1 Introduction

Previous work has demonstrated that it is possible to use adaptive resonance methods [1, 2] such as ART-1 [3] for feature detection in image recognition problems if the images involved have been appropriately preprocessed. In the CORT-X method [4] these filters are formed to approximate known neural sensitivity patterns; in the neocognitron [5] like method [4] the image is segmented into regional features; and in [6, 7] Gabor filters [8] are used to approximate neural receptor profiles. All of these methods require multiple layers of neural processors and include a *priori* assumptions about the nature of the filtering or segmentation required for the pattern recognition problem. The addition of layers of processors decreases recognition speed by lowering the degree of parallelism in the system. *A priori* assumptions can cause the system to be specialized to a narrower range of applications and decreases system flexibility.

[1] Certain commercial equipment may be identified in order to adequately specify or describe the subject matter of this work. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the equipment identified is necessarily the best available for the purpose.

1.1 FAUST Architecture

The FAUST architecture provides a self-organizing method of feature extraction and classification [9]. The FAUST architecture is one of several neural networks which provide self-organizing multi-map capabilities. The structure used is a multi-map procedure similar to those known to exist in the mid-level visual cortex [10]. As in previous work [11, 12, 1] the method must provide a parallel, multi-map, self-organizing, pattern classification procedure. This is achieved using a feed-forward architecture which allows multi-map features stored in weights acting as associative memories to be accessed in parallel and to trigger a symmetrically controlled parallel learning process. A diagram of the FAUST system is shown in figure 1. This method allows features of different data type, such as binary image patterns and multi-bit statistical correlations, to be updated in parallel. This capability is provided by the parallel pattern association and relevance paths shown in figure 1 and by the existence of separate input modules for each path.

A pattern comparison method is used to form a centralized learning control which is contained in the symmetric triggering learning control block. The triggering block gates data into the learning block on the right of figure 1. This combined architecture is described by the acronym FAUST (Feed-forward Association Using Symmetrical Triggering). The three essential features of FAUST shown in this figure are: 1) Different feature classes use individual association rules in the pattern comparison blocks. 2) Different feature classes use individual learning rules as illustrated by the pattern modification blocks. 3) All feature classes contribute symmetrically to learning as illustrated by the functional symmetry of the pattern and relevance paths. The number of feature classes is shown as two in figure 1 for graphic clarity but the architecture is not restricted to any number or type of feature classes.

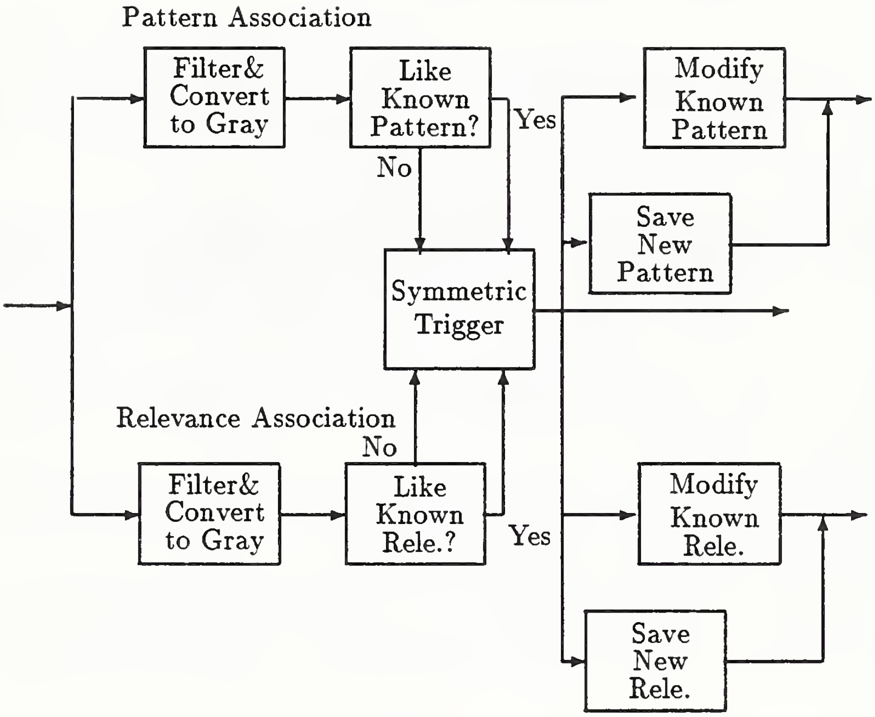


Figure 1: FAUST architecture disgram.

1.2 Outline of Paper

The organization of the paper follows the parallel data flow shown in figure 1.2. Incoming data is filtered and presented, in parallel, to each of the associative memory maps. The filtering methods used are discussed in [9]. The methods of associative recall used to locate the best match are discussed in section 2. After the association strength has been determined for each memory map, learning is triggered by a set of parallel comparisons that gate input image data to the learning modules. This symmetric triggering of learning is discussed in section 3. When learning has been triggered, the learning module for each memory map updates the selected memory locations. The learning methods used are discussed in section 4. As an example of the ability of FAUST to perform self-organizing pattern recognition, two character recognition examples and two pattern classification examples are presented in section 5.

2 Associative Recall

The data stored in the associative memories, $\mathbf{M}$, is compared to the image, $\mathbf{q}$. The association between a memory and an image is 0.0 for no similarity and 1.0 for a perfect match. The associative memories are initialized for the pattern with the memory value $\mathbf{M} = 1$ and are initialized for the relevance with the value $\mathbf{M} = 0$. Everything is true but nothing is relevant. The maximum range on the 8-bit memories is $\pm S$.

The associative functions tested in the present implementation of FAUST are shown in Table 1. Five functions are used: correlation, $1/(1 + \tan^2 \theta$, $1/(1 + d^2)$, Tanimoto similarity, and an offset cosine. Three of these functions, cosine, Euclidian distance, d, and Tanimoto, are discussed in [13]. The correlation based method is the one used in [4]. The tangent based function has a mathematical form similar to Tanimoto with the same properties near a perfect match as the distance based form. All of the indicated sums are carried out over all pixels in the image.

The class of usable similarity functions is very large. Any monotonic, single valued relationship between the image, $\mathbf{q}$, and the memory contents, $\mathbf{M}$, which can be conveniently mapped on to the interval 0 to 1, could be used.

The efficiency of any particular function will depend on the ability to separate correct classifications from weak associations on marginal images. In a character recognition application, the efficiency is partially counter balanced by computational cost.

3 Symmetric Triggering of Learning

During the learning process, images are presented, filtered, and compared with stored patterns using the association strength equations of section 3. If the match is adequate, the image is used to update the memory in the learning phase. If the image is not sufficiently similar to the existing patterns then it is treated as a unique pattern and placed in a new memory location. Symmetric triggering is the concurrent logical operation which uses comparable logical structures to update all of the associative memory blocks in parallel. This parallel operation occurs in an association space which has a dimension equal to the number of feature classes and associative memory blocks.

The most general form of the FAUST triggering logic for N associative memories for each feature classe and M feature classes, for a total of $N \times M$ memories which each have association strengths, $A_{i,j}$, and logical thresholds for triggering, ρ_j, where $j = 1, \ldots, M$ and $i = 1, \ldots, N$ is:

$$\max_i(\sum_{j=1}^{M}(A_{i,j} - \rho_j)^2)\&(A_{i,1} > \rho_1)\ldots\&(A_{i,j} > \rho_j)\ldots\&(A_{i,N} > \rho_N) \qquad (1).$$

The learning is triggered symmetrically in the ith set on N memories, and N features are being learned across M feature classes. This is shown in two dimensions, $M = 2$, in figure 2.

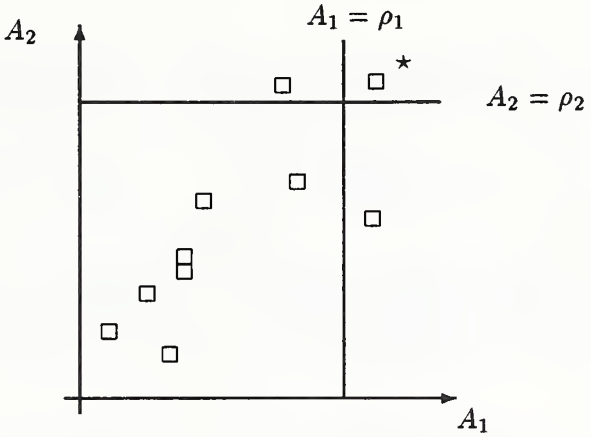


Figure 2: The association space diagram which is used for symmetric triggering. The □'s are for non-matching points. The ⋆ is for the correct classification. The limits $A_{i,j} > \rho_j$ for $j = 1, 2$ are also marked.

Function	8-bit form	binary form
Correlation	$-\Sigma(-1)^{\mathbf{q}}\mathbf{M}$	$\Sigma(\mathbf{q} \equiv \mathbf{M})$
$1/(1+\tan^2\theta)$	$\frac{1}{1+\frac{\mathbf{M}^2\mathbf{q}^2+(S\sum(-1)^{\mathbf{q}}\mathbf{M})^2}{(-\sum(-1)^{\mathbf{q}}\mathbf{M}))^2}}$	$\frac{1}{1+\frac{\mathbf{M}^2\mathbf{q}^2+(\sum(\mathbf{q}\equiv\mathbf{M}))^2}{(\sum(\mathbf{q}\equiv\mathbf{M}))^2}}$
$1/(1+d^2)$	$\frac{1}{1+\frac{\sum(\mathbf{M}+(-1)^{\mathbf{q}}S)^2}{4NS^2}}$	$\frac{1}{1+\sum\sim(\mathbf{q}\equiv\mathbf{M})/N}$
Tanimoto	$\frac{S\sum(-1)^{\mathbf{q}}\mathbf{M}}{\mathbf{M}^2+\mathbf{q}^2+S\sum(-1)^{\mathbf{q}}\mathbf{M}}$	$\frac{\sum(\mathbf{q}\equiv\mathbf{M})}{2N-\sum(\mathbf{q}\equiv\mathbf{M})}$
Offset Cosine	$\frac{\sum(-1)^{\mathbf{q}}\mathbf{M}+N}{2N\sqrt{\sum\mathbf{M}^2}+\sum S^2}$	$(\Sigma(\mathbf{q} \equiv \mathbf{M}) + N)/(2\sqrt{2}N)$

Table 1: Function used for associative comparison of binary input image, $\mathbf{q}$, and memory, M, for 8-bit and binary memory data. Each image has N elements and 8-bit elements have a maximum value S. Binary input are treated arithmetically as being zero or one. Therefore $(-1)^{\mathbf{q}}$ is 1 when $\mathbf{q} = 0$ and -1 when $\mathbf{q} = 1$.

A less general type of triggering, similar to the ART methods, is obtained if X takes on values R and P and $N = 2$. Pattern existence data is represented by n binary map feature classes indexed on j and with i occurrences of each pattern with association strength, $P_{i,j}$. Triggering is initiated by n vigilance parameters, ρ_j. These vigilance parameters are not counted in the left hand distance term and pattern strength. Relevance data is represented by m multi-bit map feature classes indexed on k with i occurrences of each strength class with association strength, $R_{i,k}$, which have $\rho_k = 0$. The generalized FAUST logic for triggering learning takes the form: Find an i and $P = \prod_{j=1}^{n}(P_{i,j}) \times \prod_{k=1}^{m}(R_{i,k})$ using:

$$\max(\sum_{k=1}^{m}(R_{i,k}^2))\&(P_{i,1} > \rho_1)\ldots\&(P_{i,j} > \rho_j)\ldots\&(P_{i,n} > \rho_n) \qquad (2).$$

For the case discussed in the example presented here, $j = 1$ and $k = 1$ so that the FAUST triggering logic reduces to: find an i using:

$$\max(R_i^2)\&(P_i > \rho) \qquad (3).$$

A typical association space diagram for a set of R_i and P_i points are shown in figure 3 for $i = 1, 2, \ldots, 10$. The area P for the maximum case is marked by the dashed lines. The limit $P_i > \rho$ is marked by a vertical line. The case which triggers learning is marked by a bullet, •. Weaker associations which do not result in learning are shown as boxes, □.

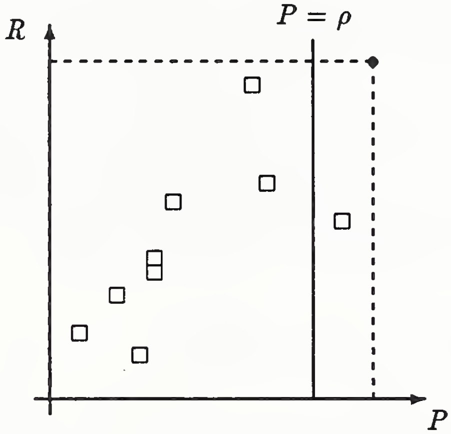


Figure 3: The association space diagram which is used for symmetric triggering. The □'s are for non-matching points. The • is for the correct classification. The area P and the limit $P_i > \rho$ are also marked.

4 Learning Methods

After learning is triggered, information from the image, $\mathbf{q}$, is stored in one of the 8-bit memories used by the relevance feature class, relevance memory, and into one of the binary memories used by the pattern feature class. The location used is determined by the logic discussed in section 2. Any absolutely stable and convergent learning rule may be used. Six different learning rules of varying complexity have been used for the relevance memories; two different rules have been used for the pattern memories. During learning the class of the sample images is not used. The process is self-organizing and requires no knowledge of class to construct the learned images.

4.1 Learning in Multi-Bit Memories

All of the learning methods used for 8-bit relevance memory take the form:

$$M_{i,j}(t+1) = g\left(M_{i,j}(t) + \alpha \Delta M_{i,j}\right) \tag{4}$$

where if $\pm S$ is the scale, maximum range, of $\mathbf{M}$, the limit function g is given by:

$$g(x) = \begin{cases} S & \text{if } x > S \\ x & \text{if } S \geq x \geq -S \\ -S & \text{if } x < -S \end{cases} , \tag{5}$$

α is the learning rate, and the memories are updated from epoch t to epoch $t+1$. The time dependence generates a dynamically stable learning sequence. All of the rules used here involve mechanisms which generate positive feedback between the memory locations and the image. The function g is used to provide a bound on this feedback.

The first rule is the DYSTAL[10] rule given by:

$$\Delta M_{i,j} = \begin{cases} S - M_{i,j}(t) & \text{if } q_{i,j}(t+1) > 0 \\ M_{i,j}(t) - S & \text{otherwise} \end{cases} \tag{6}$$

This rule was developed in [13] to more closely approximate the behavior of neurons than previous rules. This has important consequences for the stability of the learning process. Unlike the other rules used here, this rule is self limiting at the scale values $\pm S$. This rule is only applied vertically; data from each pixel only affect one memory element during learning. When a more extended field is used, image data affects memory elements over some local region referred to as a receptor field. A rule of the DYSTAL type has a distinct stability advantage in the receptor field case over non-linear limit functions.

The second rule used is a simple Hebbian rule [14] of the form:

$$\Delta M_{i,j}(t) = \begin{cases} S & \text{if } q_{i,j}(t+1) > 0 \\ -S & \text{otherwise} \end{cases} \tag{7}$$

This rule is used because it is simpler and faster than the rules used in [15, 16] but can be compared with these rules. The stability on the rule is guaranteed since only vertical interconnections are used.

The third rule used is a vertically distributed Hebbian rule [15] of the form:

$$\Delta M_{i,j}(t) = \sum_{k=i-1}^{k=i+1} \sum_{l=j-1}^{l=j+1} \begin{cases} S & \text{if } q_{k,l}(t+1) > 0 \\ -S & \text{otherwise} \end{cases} \tag{8}$$

The vertical part of the learning is identical to rule two. The field used is a 3 by 3 square.

The fourth rule used is a vertically distributed Hebbian rule and a laterally connected anti-Hebbian rule [16] of the form:

$$\Delta M_{i,j}(t) = \begin{array}{l} \sum_{k=i-1}^{k=i+1} \sum_{l=j-1}^{l=j+1} \begin{cases} S & \text{if } q_{k,l}(t+1) > 0 \\ -S & \text{otherwise} \end{cases} \\ -\sum_{k=i-1}^{k=i+1} \sum_{l=j-1}^{l=j+1} \begin{cases} S & \text{if } \operatorname{sgn}(M_{k,l}(t)) \equiv \operatorname{sgn}(M_{i,j}(t)) \\ -S & \text{otherwise} \end{cases} \end{array} \tag{9}$$

where

$$\operatorname{sgn}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0 \\ -1 & \text{if } x < 0 \end{cases} \tag{10}$$

The vertical part of the learning is identical to rule three. The lateral component is anti-Hebbian.

The fifth rule used is a vertically distributed Hebbian rule with Gaussian weights [15] on a field n_f square which must be summed over $q = (n_f - 1)/2$ elements and takes the form:

$$\Delta M_{i,j}(t) = \sum_{k=i-q}^{k=i+q} \sum_{l=j-q}^{l=j+q} \begin{cases} \exp(-(i^2 + j^2)/\sigma^2)S & \text{if } q_{k,l}(t+1) > 0 \\ -\exp(-(i^2 + j^2)/\sigma^2)S & \text{otherwise} \end{cases} \tag{11}$$

where σ is the envelope variance of the field.

The sixth rule used is a vertically distributed Hebbian rule and a laterally connected anti-Hebbian rule with Gaussian weights [16] on field n_f square which must be summed over $q = (n_f - 1)/2$ elements and takes the form:

$$\Delta M_{i,j}(t) = \begin{array}{l} \sum_{k=i-q}^{k=i+q} \sum_{l=j-q}^{l=j+q} \begin{cases} \exp(-(i^2 + j^2)/\sigma^2)S & \text{if } q_{k,l}(t+1) > 0 \\ -\exp(-(i^2 + j^2)/\sigma^2)S & \text{otherwise} \end{cases} \\ -\sum_{k=i-q}^{k=i+q} \sum_{l=j-q}^{l=j+q} \begin{cases} \exp(-(i^2 + j^2)/\sigma^2)S & \text{if } \operatorname{sgn}(M_{k,l}(t)) \equiv \operatorname{sgn}(M_{i,j}(t)) \\ -\exp(-(i^2 + j^2)/\sigma^2)S & \text{otherwise} \end{cases} \end{array} \tag{12}$$

where σ is again the envelope variance of the field.

4.2 Learning in binary memories

The two rules used for binary pattern learning are a logical "OR" of the positively relevant elements:

$$M_{i,j}(t+1) = \begin{cases} M_{i,j}(t) \vee q_{i,j}(t+1) & \text{if relevance} > 0 \\ M_{i,j}(t) & \text{otherwise} \end{cases} \tag{13}$$

and a logical product of the image and relevant elements.

$$M_{i,j}(t+1) = \begin{cases} 1 & \text{if } q_{i,j}(t+1) > 0 \text{ and relevance} > 0 \\ 0 & \text{if } q_{i,j}(t+1) = 0 \text{ and relevance} < 0 \\ M_{i,j}(t) & \text{otherwise} \end{cases} \tag{14}$$

5 Results

The basic structures of the character recognition, fingerprint classification systems and forms recognition systems are similar. All systems have a loading phase which includes image decompression on the host serial computer, a feature extraction phase, and a recognition phase. For character recognition the isolated character images are loaded directly into the FAUST recognition module which does the feature extraction. The fingerprint classification system uses a ridge-valley based feature isolation and alignment method. A binary ridge-valley map is loaded into the FAUST module and global features are learned. The forms recognition system does a massive downsample to map the page image into a single 32 by 32 binary image.

The function of all systems is similar. A raster scanned image is input to the system and ASCII classifications are returned. For the character recognition system, the input image is a binary image of a page containing 8,000,000 pixels. For the finger print system, the input image is a 512 by 512 8-bit gray level image. The character recognition system returns a page of character classifications. The fingerprint classification system returns an ASCII character representing one of five fingerprint classifications. For the forms recognition system, the input image is a binary image of a page containing 8,000,000 pixels. An ASCII code repersenting one of 20 form types is returned.

5.1 Regular Shapes

5.1.1 Machine Print

For machine-print data with the correct choice of ρ, it is possible to achieve 99.7% recognition on test samples of 10000 characters. The association rules discussed in section 2 affect the sensitivity of learning and the confidence levels in the triggering process discussed in sectin 3. This is caused by the variation in sensitivity of the learnibg functions discussed in section 4 to image similarity differences. The minimum recognition rate is achieved using inverse square distance association and resonance classification and is 2.4ms/character.

5.1.2 Forms

The FAUST method of document recognition is independent of the local variations in forms caused either by printing or by the distributions of user written responses on individual forms. This method involves downsampling the entire form and converting the image into a 32 by 32 binary image. The original form shown in figure 4 is taken from NIST Special Database 2 [17]. A downsampled binary image is shown in figure 5. The binary image is then used, in a way similar to character images, to develop features of each form type using the FAUST recognition module. Testing on 1000 pages of forms achieved 94% recognition with 2% incorrect identifications and 4% unknown forms.

5.2 Irregular Shapes

5.2.1 Handprint Digits

Table 2 shows the classification error for three different samples of 512 hand printed digits. The digits are taken from hand printed digits contained in the NIST hand print database [18]. Data from 18 different individuals were used in each test file. Nine individuals were used in the learning phase, and a different nine individuals were used to test classification. Several different filter types were used on the hand printed characters. Undetected error rates are a minimum of 4.9% and detected error rates are 12.1%. The most effective filter combination [9] is shown to be a shear transform followed by a Gabor filter. These combined filters reduce the substitutional error by a factor of three and the number of unknowns by a factor of two.

Only the shear and Gabor filter types were used since they were the most effective Undetected error rates are a minimum of 2.3% and detected error rates are about 9.4%. The substitutional error has been cut in half with optimal recognition parameters and the recognition rate has improved from 83% to 88%. Since the number of filter types, association types, learning rules, and classification rules form a large set which has not been completely explored, better values for recognition rate may be possible.

5.2.2 Fingerprints

The fingerprint classification system differs from the character recognition system in an important way. Fingerprints are natural objects which have classes designated by humans. These classes merge smoothly into each other. The five classes used are right loop, left loop, whorl, arch and tented arch [19]. The classification is made using a 512 by 512 gray level image.

After the fingerprint image is decompressed and loaded into the array processor, alignment features are extracted. A rule based alignment method is then used to center the print in the image field [20]. After the print has been centered in the image field, classification features are extracted. Both alignment features and classification features are extracted using local ridge slope data.

A binary ridge-valley map is created by assigning a "1" to all ridge-valley values less than 90° and a "0" to all other values. This binary map is related to the original fingerprint as shown in figure 6. These binary maps are used to train and test the FAUST fingerprint classification. The results of this process are shown in table 3. The receptor field size is the parameter q in (11).

6 Conclusions

A new self-organizing architecture, FAUST, has been developed. Examples from image recognition show that the method works for both small scale direct recognition, such as character recognition, and large scale image classification problems, such as form recognition and fingerprint classification. Recognition of both regular objects, machineprint and forms, and irregular objects, handprint and fingerprints, is possible. Higher accuracy is usually possible on regular objects.

Acknowledgement

The author would like to acknowledge the members of the image recognition group research term who contributed to the work described here: J. L. Blue, G. T. Candela, D. L. Dimmick, M. D. Garris, P. J. Grother, and R. A. Wilkinson.

References

[1] C. L. Wilson, R. A. Wilkinson, and M. D. Garris. Self-organizing neural network character recognition on a massively parallel computer. In *Proc. of the IJCNN, volume II*, pages 325–329, June 1990.

[2] C. L. Wilson, R. A. Wilkinson, and M. D. Garris. Self-organizing neural network character recognition using adaptive filtering and feature extraction. *Progress in Neural Networks*, 3, 1991. to be published.

[3] G. A. Carpenter and S. Grossberg. A massively parallel architecture for a self-organizing neural pattern recognition machine. *Computer Vision, Graphics, and Image Processing*, 37:54–115, 1987.

[4] G. A. Carpenter, S. Grossberg, and C. Mehanian. Invariant recognition of cluttered scenes by a self-organizing art architecture: CORT-X boundary segmentation. *Neural Networks*, 2:169–181, 1989.

[5] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shifts in position. *Biological Cybernetics*, 36:193–202, 1980.

[6] M. D. Garris, R. A. Wilkinson, and C .L. Wilson. Analysis of a biolgically motivated neural network for character recognition. In *Proceedings: Analysis of Neural Network Applications*. ACM Press, George Mason University, 1991.

[7] M. D. Garris, R. A. Wilkinson, and C .L. Wilson. Methods for enhancing neural network handwritten character recognition. In *International Joint Conference on Neural Networks*, volume 1, pages 695–700. IEEE and International Neural Network Society, 7 1991.

[8] J. G. Daugman. Complete discrete 2-d Gabor transform by neural networks for image analysis and compression. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 36:1169–1179, 1988.

[9] C. L. Wilson. A new self-organizing neural network architecture for parallel multi-map pattern recignition - FAUST. *Progress in Neural Networks*, 4, 1992. to be published.

[10] A. Rojer and E. Schwatz. Multi-map model for pattern classification. *Neural Computation*, 1:104–115, 1989.

[11] L. D. Jackel, H. P. Graf, W. Hubbard, J. S. Denker, D. Henderson, and Isabelle Guyon. An application of neural net chips: Handwritten digit recognition. In *IEEE International Conference on Neural Networks, volume II*, pages 107–115, San Diego, 1988.

[12] A. Rajavelu, M. T. Musavi, and M. V. Shirvaikar. A neural network approach to character recognition. *Neural Networks*, 2:387–393, 1989.

[13] D. L. Alkon, K. T. Blackwell, G. S. Barbour, A. K. Rigler, and T. P. Vogl. Pattern-recognition by an artificial network derived fron biological neuronal systems. *Biological Cybernetics*, 62:363–376, 1990.

[14] D. O. Hebb. *The Organization of Behavior*. Wiley, New York, 1949.

[15] R. Linsker. Self-organization in a perceptual network. *Computer*, 21:105–117, 1988.

[16] J. Rubner and K. Schulten. Development of feature detectors by self-organization. *Biological Cybernetics*, 62:193–199, 1990.

[17] D. L. Dimmick, M. D. Garris, and C. L. Wilson. Structured forms database. *National Institute of Standards and Technology*, Special Database 2, **SFRS**, December 1, 1991.

[18] C. L. Wilson and M. D. Garris. Handprinted character database. *National Institute of Standards and Technology*, Special Database 1, **HWDB**, April 18, 1990.

[19] *The Science of Fingerprints*. U. S. Department of Justice, Washington, DC, 1984.

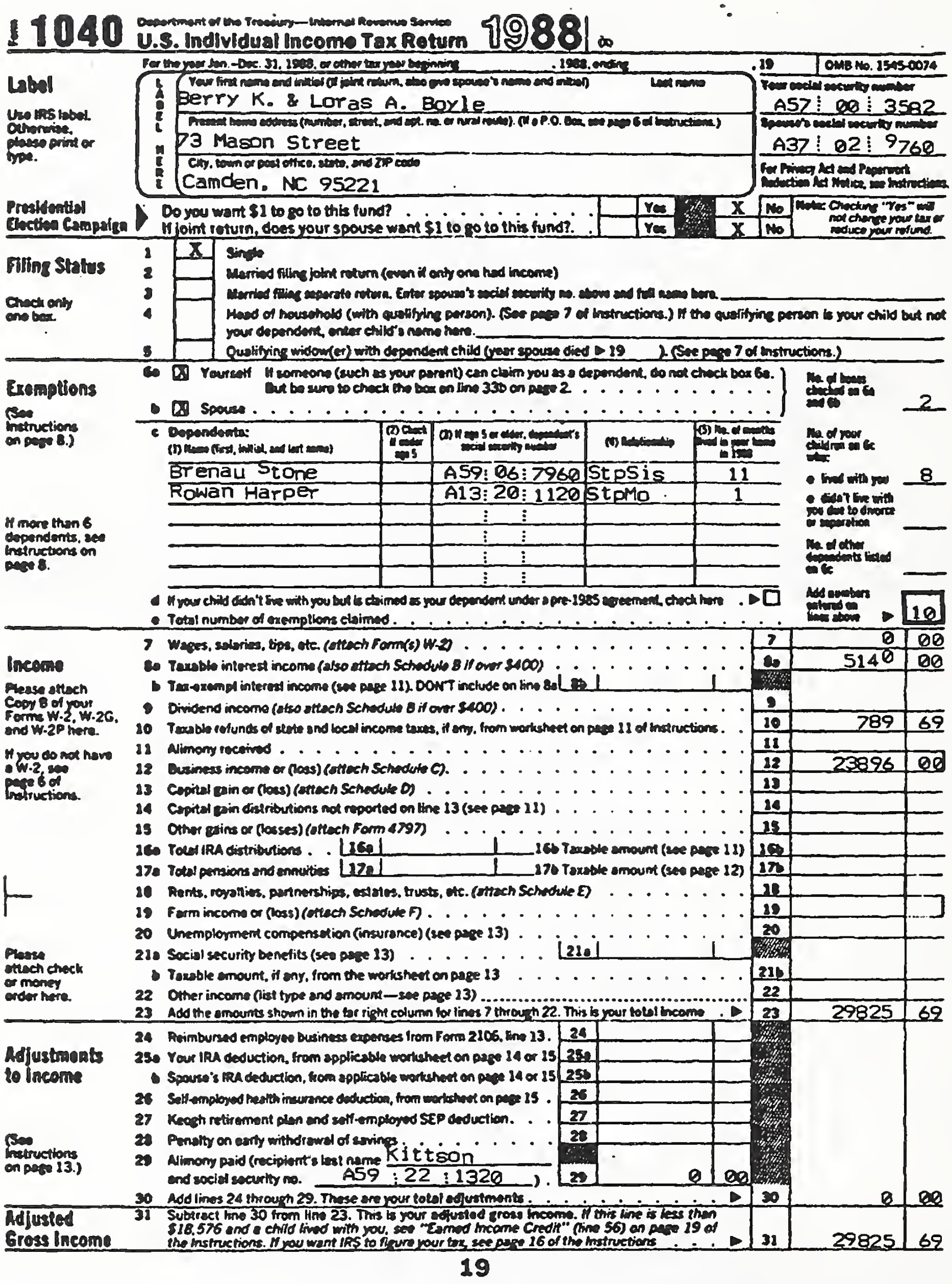
[20] J. H. Wegstein. An automated fingerprint identification system. *National Institute of Standards and Technology*, NBS Special Publication 500-89, Febuary 1982.

Sample	Memory	Filter	ρ	Wrong	Unknown
fl3aa	64	None	.8	83	106
fl3aa	64	S	.8	66	83
fl3aa	64	G	.8	56	80
fl3aa	64	S+G	.8	25	62
fl3aa	64	S+G	.85	25	57
fl3aa	64	S+G	.90	44	66
fl3aa	64	S+G	.75	54	105
fl3ab	64	S+G	.85	30	91
fl3ac	64	S+G	.85	31	72
fl3aa	32	S+G	.85	39	88
fl3ab	32	S+G	.85	44	103
fl3ac	32	S+G	.85	63	104

Table 2: Classification errors in a sample of 512 hand printed characters when several different filters are used in FAUST. Resonance classification and nearest neighbor Hebbian learning are used throughout. S is a shear transform and G is a Gabor filter

Receptor Field	Test Accuracy	Training Accuacy
3	73.32	81.20
5	78.73	79.93
7	77.61	79.37
9	78.03	79.37

Table 3: Classification errors in a sample of 1420 fingerprints when several different receptor fields sizes are used in FAUST learning using equation (11). Resonance classification and correlation based association used throughout.



Form 1040 Department of the Treasury—Internal Revenue Service
U.S. Individual Income Tax Return 1988

For the year Jan.–Dec. 31, 1988, or other tax year beginning , 1988, ending , 19 | OMB No. 1545-0074

Label

Use IRS label. Otherwise, please print or type.

LABEL HERE

Your first name and initial (if joint return, also give spouse's name and initial) Last name
Berry K. & Loras A. Boyle

Present home address (number, street, and apt. no. or rural route). (If a P.O. Box, see page 6 of Instructions.)
73 Mason Street

City, town or post office, state, and ZIP code
Camden, NC 95221

Your social security number
A57 : 00 : 3582

Spouse's social security number
A37 : 02 : 9760

For Privacy Act and Paperwork Reduction Act Notice, see Instructions.

Presidential Election Campaign

Do you want $1 to go to this fund?	Yes		X	No	Note: Checking "Yes" will not change your tax or reduce your refund.
If joint return, does your spouse want $1 to go to this fund?.	Yes		X	No	

Filing Status

Check only one box.

1 ☒ Single
2 □ Married filing joint return (even if only one had income)
3 □ Married filing separate return. Enter spouse's social security no. above and full name here. ____
4 □ Head of household (with qualifying person). (See page 7 of Instructions.) If the qualifying person is your child but not your dependent, enter child's name here. ____
5 □ Qualifying widow(er) with dependent child (year spouse died ▶ 19). (See page 7 of Instructions.)

Exemptions

(See Instructions on page 8.)

6a ☒ Yourself If someone (such as your parent) can claim you as a dependent, do not check box 6a. But be sure to check the box on line 33b on page 2.
b ☒ Spouse

No. of boxes checked on 6a and 6b: 2

c Dependents:

(1) Name (first, initial, and last name)	(2) Check if under age 5	(3) If age 5 or older, dependent's social security number	(4) Relationship	(5) No. of months lived in your home in 1988
Brenau Stone		A59 : 06 : 7960	StpSis	11
Rowan Harper		A13 : 20 : 1120	StpMo	1
		: :		
		: :		
		: :		
		: :		

If more than 6 dependents, see Instructions on page 8.

No. of your children on 6c who:
- lived with you: 8
- didn't live with you due to divorce or separation: ____

No. of other dependents listed on 6c: ____

d If your child didn't live with you but is claimed as your dependent under a pre-1985 agreement, check here ▶ □
e Total number of exemptions claimed

Add numbers entered on lines above ▶ 10

Income

Please attach Copy B of your Forms W-2, W-2G, and W-2P here.

If you do not have a W-2, see page 6 of Instructions.

Please attach check or money order here.

7	Wages, salaries, tips, etc. *(attach Form(s) W-2)*	7	0	00
8a	Taxable interest income *(also attach Schedule B if over $400)*	8a	5140	00
b	Tax-exempt interest income (see page 11). DON'T include on line 8a 8b			
9	Dividend income *(also attach Schedule B if over $400)*	9		
10	Taxable refunds of state and local income taxes, if any, from worksheet on page 11 of Instructions	10	789	69
11	Alimony received	11		
12	Business income or (loss) *(attach Schedule C)*.	12	23896	00
13	Capital gain or (loss) *(attach Schedule D)*	13		
14	Capital gain distributions not reported on line 13 (see page 11)	14		
15	Other gains or (losses) *(attach Form 4797)*	15		
16a	Total IRA distributions 16a 16b Taxable amount (see page 11)	16b		
17a	Total pensions and annuities 17a 17b Taxable amount (see page 12)	17b		
18	Rents, royalties, partnerships, estates, trusts, etc. *(attach Schedule E)*	18		
19	Farm income or (loss) *(attach Schedule F)*	19		
20	Unemployment compensation (insurance) (see page 13)	20		
21a	Social security benefits (see page 13) 21a			
b	Taxable amount, if any, from the worksheet on page 13	21b		
22	Other income (list type and amount—see page 13)	22		
23	Add the amounts shown in the far right column for lines 7 through 22. This is your total income ▶	23	29825	69

Adjustments to Income

(See Instructions on page 13.)

24	Reimbursed employee business expenses from Form 2106, line 13.	24		
25a	Your IRA deduction, from applicable worksheet on page 14 or 15	25a		
b	Spouse's IRA deduction, from applicable worksheet on page 14 or 15	25b		
26	Self-employed health insurance deduction, from worksheet on page 15	26		
27	Keogh retirement plan and self-employed SEP deduction.	27		
28	Penalty on early withdrawal of savings	28		
29	Alimony paid (recipient's last name Kittson and social security no. A59 : 22 : 1320).	29	0	00
30	Add lines 24 through 29. These are your total adjustments ▶	30	0	00

Adjusted Gross Income

31	Subtract line 30 from line 23. This is your adjusted gross income. *If this line is less than $18,576 and a child lived with you, see "Earned Income Credit" (line 56) on page 19 of the Instructions. If you want IRS to figure your tax, see page 16 of the Instructions* ▶	31	29825	69

19

Figure 4: Simulated IRS 1040 page 1; this is the first image from NIST Special Database 2.

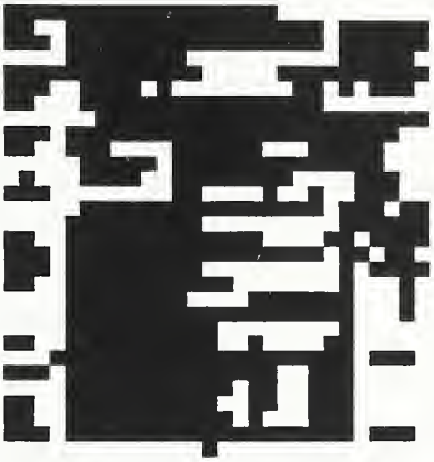


Figure 5: Image of a 1040 page 1 down-sampled to fit a 32 by 32 binary image.

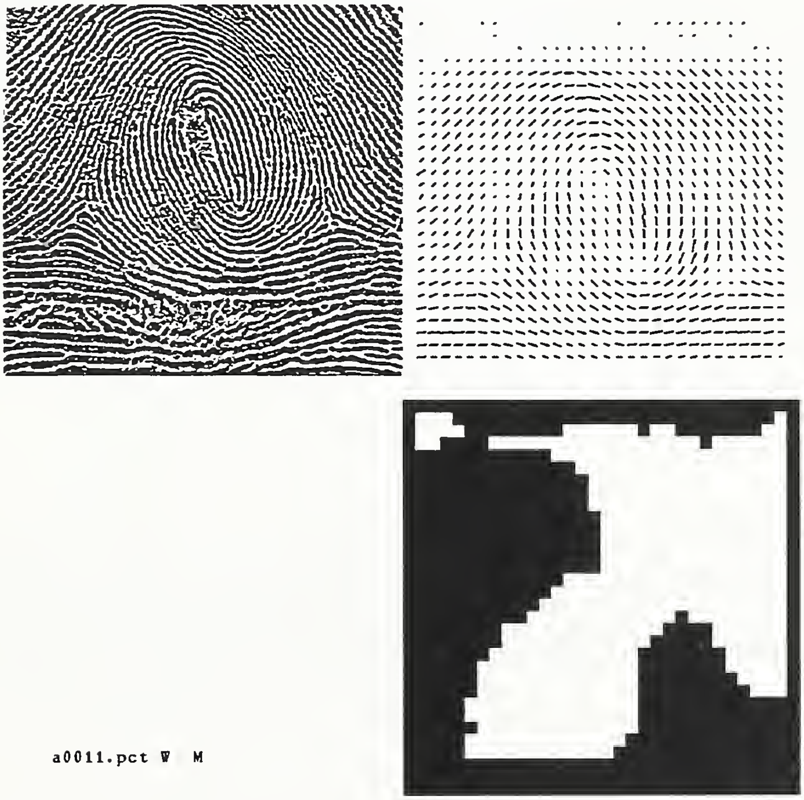


Figure 6: Fingerprint with ridge-valley map and reduced binary map.

IIST-114A
REV. 3-90)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NUMBER
NISTIR 4805

2. PERFORMING ORGANIZATION REPORT NUMBER

3. PUBLICATION DATE
MARCH 1992

TITLE AND SUBTITLE

FAUST: A Vision-Based Neural Network Multi-Map Pattern Recognition Architecture

. AUTHOR(S)

C. L. Wilson

. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
GAITHERSBURG, MD 20899

7. CONTRACT/GRANT NUMBER

8. TYPE OF REPORT AND PERIOD COVERED

. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)

0. SUPPLEMENTARY NOTES

1. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.)

A new architecture is presented for multi-map, self-organizing pattern recognition which allows concurrent massively parallel learning of features using different maps for each feature type. The method used is similar to the multi-map structures known to exist in the vertebrate sensory cortex. The learning used to update memory locations uses a feed-forward mechanism and is self-organizing. The architecture is described by the acronym FAUST (Feed-forward Association Using Symmetrical Triggering). As a demonstration of the effectiveness of FAUST, a character recognition program has been constructed on a massively parallel computer which can perform 99% accurate character recognition on a medium-quality machine printed digits at a speed of 2.4ms/digit, and 88% recognition on multiple-writer hand print with a 2.3% substitutional error rate. All of the calculations were performed on an Active Memory Technology DAP 510[1].

[1] Certain commercial equipment may be identified in order to adequately specify or describe the subject matter of this work. In no case does such identification imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the equipment identified is necessarily the best available for the purpose.

2. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)

neural network; self-organizing; character recognition; fingerprint classification; parallel computing.

3. AVAILABILITY

☒ UNLIMITED

□ FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).

□ ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.

☒ ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.

14. NUMBER OF PRINTED PAGES
18

15. PRICE
A02

LECTRONIC FORM