Using Self-Organizing Recognition as a Mechanism For Rejecting Segmentation Errors

R. Allen Wilkinson, Michael D. Garris and Charles L. Wilson,
National Institute of Standards and Technology,
Gaithersburg, MD 20899

Abstract

We have developed a self-organized neural network based method that concurrently detects segmentation errors and performs character recognition. This method utilizes a two-pass classification scheme. A page of machine printed text is segmented, and a pre-trained self-organizing classifier is used to recognize the images produced by the segmenter. Images that are recognized with a sufficiently high confidence are used to retrain the classifier, adapting the neural network to the current font type being segmented. All the segmented images are then reclassified by the adapted network. The assigned classes of those images which are confidently recognized are accepted, whereas the images which are not confidently recognized are rejected. The pages of text used to develop this method were randomly generated so that no context can be used to correct segmentation or recognition errors. In one experiment, the first classification pass rejected 6.6% of the total number of images segmented. Of these rejected images, only 3.6% were truly segmentation errors. The other 96.4% were correctly segmented. A traditional single-pass classification scheme such as this results in the rejection of an unnecessarily high number of correctly segmented characters, reducing effective system throughput. By retraining the network on only those images accepted by the first pass, the second classification pass rejected only 3.5% of the segmented images, of which 6.7% were segmentation errors. The second pass achieved an accuracy of 99.3% on those images accepted. This clearly demonstrates the network's ability to adapt on the second pass, increasing system throughput by 3.1%. In all the cases studied, greater than 99% of all segmentation errors were detected without any human intervention.

1 Introduction

A histogram-based segmentation algorithm has been developed at NIST[1]. Developed for use on a parallel machine[2], this algorithm proved to be fast and accurate for machine print. This histogram-based segmenter can be used on full page images of text to allow large volumes of machine printed font data to be segmented. This page-level segmenter is able to segment characters of differing font sizes and styles. In order to ensure a high level of accuracy and to reduce the time needed to segment large amounts of data, a scheme was developed to detect segmentation errors while concurrently classifying character images without human intervention. Section 2 discusses the scheme that was developed. An on-line "real time" learning algorithm was used in conjunction with the page segmenter to detect segmentation and classification errors for different font sizes and styles. The results of this study are given in section 3, followed by the conclusions in section 4

2 Segmentation Error Detection and Rejection

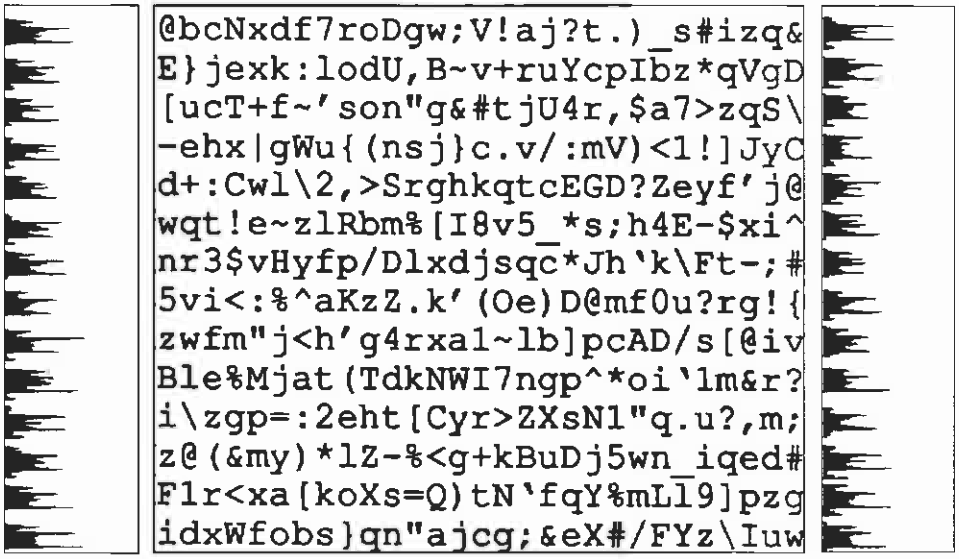
The segmenter used in this study handles multiple lines of data, breaking each isolated line into separate character images, one character per image. After segmentation, the segmented images are sent through a two-pass classification system. Both passes use a self-organizing pattern recognizer, FAUST[3], described in section 2.3. The first pass of classification uses a known set of characters that has been verified to be correct and is known to train FAUST accurately. This primed version of FAUST is used to classify the segmented images. Those images that FAUST classifies with a high confidence are then used to train a second FAUST network to be used in the second pass of classification. The second training session is used to adapt the recognizer to accurately recognize the current font being segmented. Once again, those images recognized with a high enough confidence are considered correctly segmented and classified. All other images are considered potential errors and are rejected. The number of segmented images rejected in the second pass is controlled by the confidence (vigilance) of classification computed by the self-organizing recognition process.

2.1 Segmenter

The segmenter is really two different segmenters. The first is a page segmenter that segments a line or row of text from a page of machine print. The second segmenter is a line segmenter that separates a row of text into individual character images. The page segmenter described in [1] was used for this work. Figure 1 shows a subimage of a page of text with horizontal histograms of the subimage's margins displayed on the left and right. These histograms, which are thresholded spatial binary histograms[1], are used to find lines of text by locating gaps at points of local minima. Since this method uses a column of pixels on the left and right sides it is possible to identify slanted lines.

Once the page segmenter locates a line of text, an image of the line is extracted and sent to the line segmenter. The line segmenter separates the image into subimages of characters by finding vertical voids in the line image. This is accomplished with vertical spatial binary histograms. Figure 2 shows the vertical spatial histogram of a line image. The valleys in the histogram locate the cut points for the segmenter.

There are two distinct limitations with this histogram method. First, a character containing a vertical void will be separated and assumed to be two characters. An example of a character that has a vertical void is a double quote as shown in figure 3 (A). Second, connected characters, as in figure 3 (B), will not be separated because a vertical void cannot be found between them. Figure 4 shows an example of a line with both of these limitations.



```
@bcNxdf7roDgw;V!aj?t.)_s#izq&
E}jexk:lodU,B~v+ruYcpIbz*qVgD
[ucT+f~'son"g&#tjU4r,$a7>zqS\
-ehx|gWu{(nsj}c.v/:mV)<1!]JyC
d+:Cwl\2,>SrghkqtcEGD?Zeyf'j@
wqt!e~zlRbm%[I8v5_*s;h4E-$xi^
nr3$vHyfp/Dlxdjsqc*Jh`k\Ft-;#
5vi<:%^aKzZ.k'(Oe)D@mf0u?rg!{
zwfm"j<h'g4rxal~lb]pcAD/s[@iv
Ble%Mjat(TdkNWI7ngp^*oi`lm&r?
i\zgp=:2eht[Cyr>ZXsN1"q.u?,m;
z@(&my)*1Z-%<g+kBuDj5wn_iqed#
F1r<xa[koXs=Q)tN`fqY%mL19]pzg
idxWfobs}qn"ajcg;&eX#/FYz\Iuw
```

Figure 1: Left and right histogram of a partial page

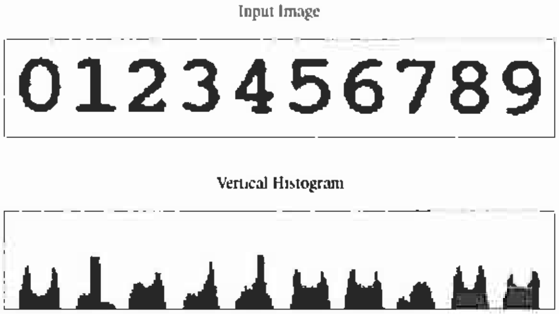


Figure 2: Histogram of an image

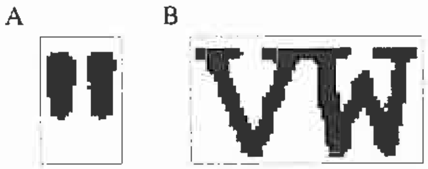


Figure 3: Examples of (A) a vertical void and (B) a connected character

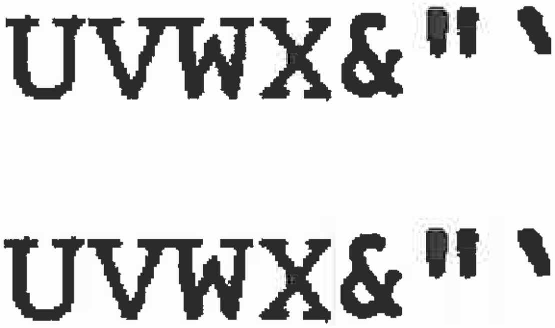


Figure 4: Example of segmentation cuts found using line segmenter

2.2 Neural Network

Previous work has demonstrated that it is possible to use adaptive resonance methods [4, 5] such as ART-1 [6] for feature detection in image recognition problems if the images involved have been appropriately preprocessed. In the CORT-X method [7] these filters are formed to approximate known neural sensitivity patterns; in the neocognitron method[8] the image is segmented into regional features; and in [9, 10] Gabor filters [11] 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 can decrease system flexibility. The self-organized segmentation and classification system presented in this paper does not use any *a priori* assumptions and therefore does not suffer from limitations due to these assumptions.

2.3 FAUST Architecture

The FAUST architecture provides a self-organizing method of feature extraction and classification [3] that avoids *a priori* assumptions but allows on-line "real time" learning. The FAUST architecture is one of several neural networks that provide self-organizing multi-map capabilities [12, 7, 6, 13, 14, 15, 16, 17]. This is achieved using a feed-forward architecture that 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 5. This method allows features of different data types, 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 5 and by the existence of separate input modules for each path.

In FAUST, a pattern comparison method is used to form a centralized learning control which is contained in the symmetric trigger learning control block. The triggering block gates data into the learning blocks on the right of figure 5. 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 5 for graphical clarity but the architecture is not restricted to any number or type of feature classes.

The vigilance parameter in FAUST is a measure of the confidence of recognition in the network. This is generated, as in other resonance methods, by forming a cross correlation between two association strengths. In FAUST these association are with the pattern and the relevance. The associations are formed in each association block using various similarity measures. See [3] for details.

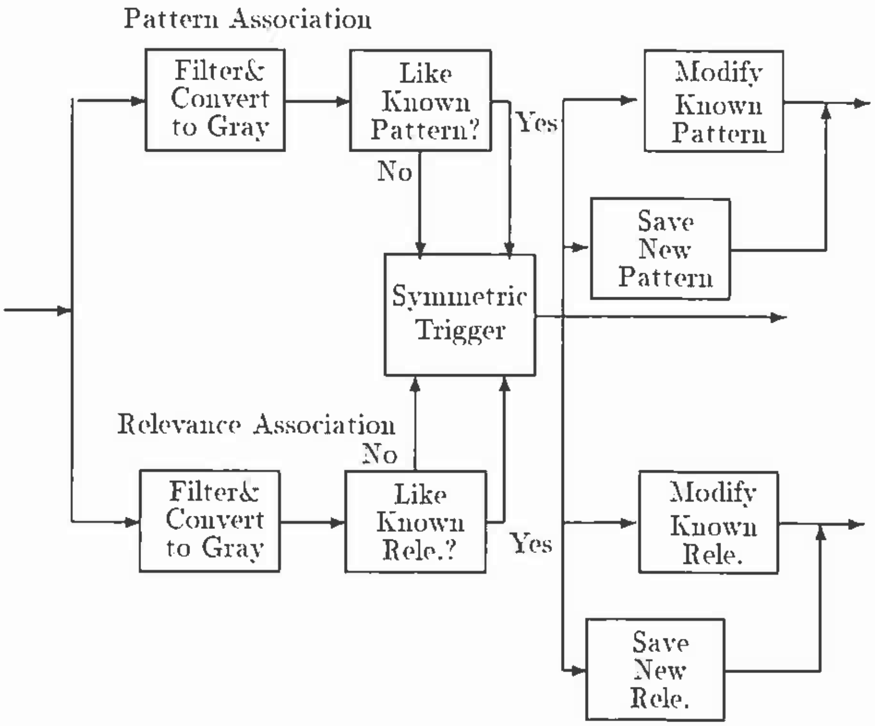


Figure 5: FAUST architecture diagram. Relevance is abbreviated to Rele.

2.4 FAUST Results

The basic structures of a character recognition system and a segmentation error detection system are similar. Both systems have a loading phase, 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 in parallel with the classification. A raster scanned image of characters is input to the system and ASCII classifications are returned. For the segmentation error detection system, the input images are scale-normalized 32x32 pixel images which are loaded directly from the final pass of the line segmenter.

For machine print data with an optimal set of FAUST parameters, it is possible to achieve 99.7% recognition on test samples of 10000 characters [17]. The association rules in FAUST affect the sensitivity of learning and the confidence levels in the triggering process. The maximum recognition accuracy rate is achieved using the inverse square distance association. Using this association rule, the resonance classification requires 2.4ms/character on the parallel computer.

3 Experiments

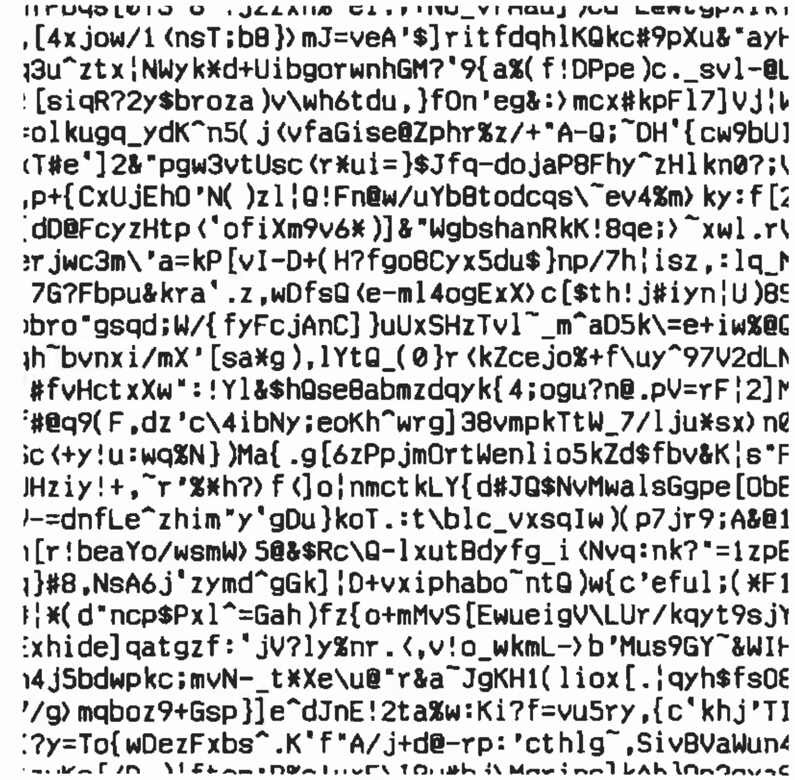
3.1 Input

The input to this experiment was a page of text with a uniformly distributed random sequence of characters. Since the page was randomly generated, there was no context available to correct segmentation or recognition errors. The page consisted of 59 lines and 78 characters per line for a total of 4602 characters per page. The characters used for the page were homogeneously mixed examples of the 26 lower case alphas, 26 upper case alphas, 10 numerics, and 32 special characters. The three fonts used are a Courier 10 point font printed by a laser printer, a dot matrix 10 point font, and a dot matrix 12 point font. These are called Courier, Tex10, and Tex12, respectively, in the following. Figure 6 is a subimage of the Tex12 page.

3.2 Results

Figures 7 through 9 plot the number of segmented images rejected (solid curves) and the number of segmentation errors left undetected (dashed curves) by the two-pass classification system. These numbers are plotted with respect to the system's confidence. This confidence acts as the trigger level in both the learning and classification phases of the FAUST network. As the confidence increases, the chance of detecting all segmentation error increases but at the expense of rejecting more and more correctly segmented characters.

Figure 7 shows the results of the two-pass classification system for the Courier page. The Courier page was segmented into a total of 4602 images. These images contained segmentation errors which were a combination of split and merged characters. The fact that the



```
,[4xjow/1(nsT;b8})mJ=veA'$]ritfdqhlKQkc#9pXu&"ayH
q3u^ztx¦NWyk*d+UibgorwnhGM?'9{a%(f!DPpe)c._svl-@L
:[siqR?2y$broza)v\wh6tdu,}fOn'eg&:)mcx#kpFl7]Vj¦W
=olkugq_ydK^n5(j(vfaGise@Zphr%z/+"A-Q;~DH'{cw9bU]
(T#e']2&"pgw3vtUsc(r*ui=}$Jfq-dojaP8Fhy^zHlkn0?;\
,p+{CxUjEhO'N()zl¦Q!Fn@w/uYb8todcqs\~ev4%m)ky:f[2
.dD@FcyzHtp('ofiXm9v6*)]&"WgbshanRkK!8qe;)~xwl.r\
er jwc3m\'a=kP[vI-D+(H?fgo8Cyx5du$}np/7h¦isz,:lq_M
 7G?Fbpu&kra'.z,wDfsQ(e-ml4ogExX)c[$th!j#iyn¦U)89
obro"gsqd;W/{fyFcjAnC]}uUxSHzTvl~_m^aD5k\=e+iw%@G
qh~bvnxi/mX'[sa*g),lYtQ_(0}r(kZcejo%+f\uy^97V2dLN
 #fvHctxXw":!Yl&$hQse8abmzdqyk{4;ogu?n@.pV=rF¦2]M
"#@q9(F,dz'c\4ibNy;eoKh^wrg]38vmpkTtW_7/lju*sx)n@
Sc(+y!u:wq%N})Ma{.g[6zPpjmOrtWenlio5kZd$fbv&K¦s"F
JHziy!+,~r'%*h?)f(]o¦nmctkLY{d#JQ$NvMwalsGgpe[ObE
)-=dnfLe^zhim"y'gDu}koT.:t\blc_vxsqIw)(p7jr9;A&@1
n[r!beaYo/wsmW)5@&$Rc\Q-lxut8dyfg_i(Nvq:nk?"=1zpE
q}#8,NsA6j'zymd^gGk]¦D+vxiphabo~ntQ)w{c'eful;(*F1
F¦*(d"ncp$Pxl^=Gah)fz{o+mMvS[EwueigV\LUr/kqyt9sjY
:xhide]qatgzf:'jV?ly%nr.(,v!o_wkmL->b'Mus9GY~&WIH
n4jSbdwpkc;mvN-_t*Xe\u@"r&a~JgKH1(liox[.¦qyh$fsOE
'/g)mqboz9+Gsp}]e^dJnE!2ta%w:Ki?f=vu5ry,{c'khj'TI
:?y=To{wDezFxbs^.K'f"A/j+d@-rp:'cthlg~,SivBVaWun4
```

Figure 6: Subimage of the Tex12 image

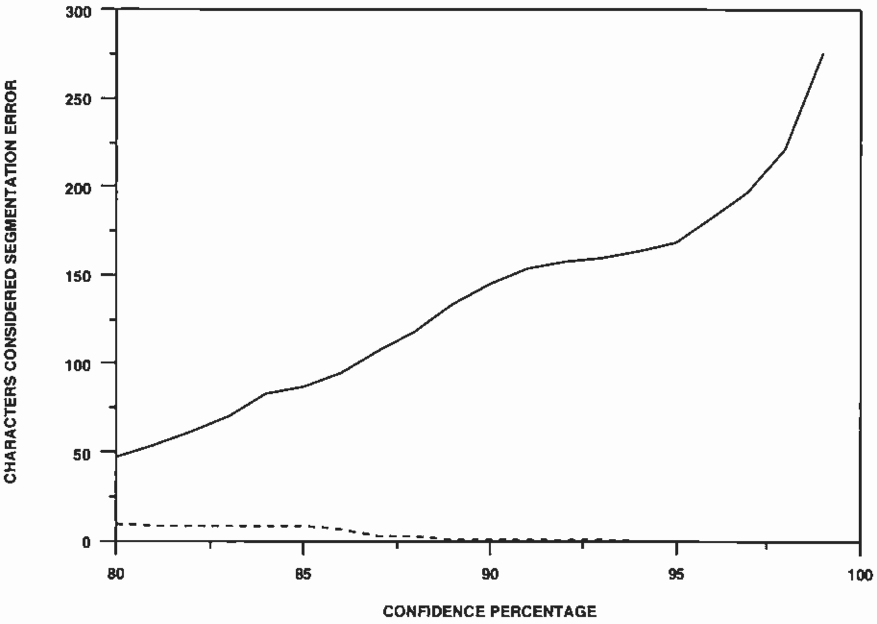


Figure 7: Courier page errors versus confidence and residual segmentation error

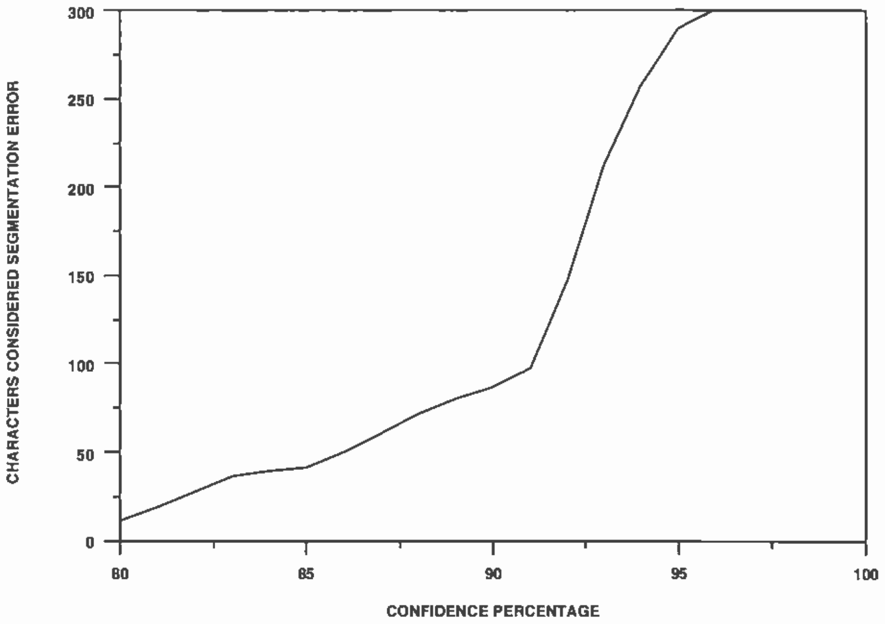


Figure 8: Tex10 page errors versus confidence

number of segmented images equals the number of characters printed on the page is purely coincidental. At a confidence of 80%, nine segmentation errors remain undetected while 45 images of correctly segmented characters have been rejected. At a confidence of 90%, all but one of the segmentation errors has been rejected at the expense of rejecting 135 images of correctly segmented characters. This supports the observation that as the confidence of the system increases, the chance of detecting segmentation errors increases at the expense of rejecting an increasing number of correctly segmented images. Of the images accepted by the system, 99.3% of them contained correctly segmented and classified characters.

Figure 8 shows the results for the Tex10 page. For this page, the segmenter was 100% accurate, so that there were no real segmentation errors to be detected by the system. At a confidence of 80%, the system rejected only 11 correctly segmented characters which represent 0.2% of the characters on the page. At a confidence of 90%, the system rejected 87 correctly segmented images. This represents less than 2% of the characters on the page. This example demonstrates how the system degrades gracefully when no segmentation errors exist.

Figure 9 shows the results for the Tex12 page. At a confidence of 80%, the system rejected one correctly segmented image while rejecting only 487 of a possible 825 segmentation errors. At a confidence of 90%, the system detected and rejected 817 segmentation errors at the expense of rejecting 43 correctly segmented images. These results once again demonstrate that, to detect more errors, the system has to reject an increasing number of correctly segmented characters.

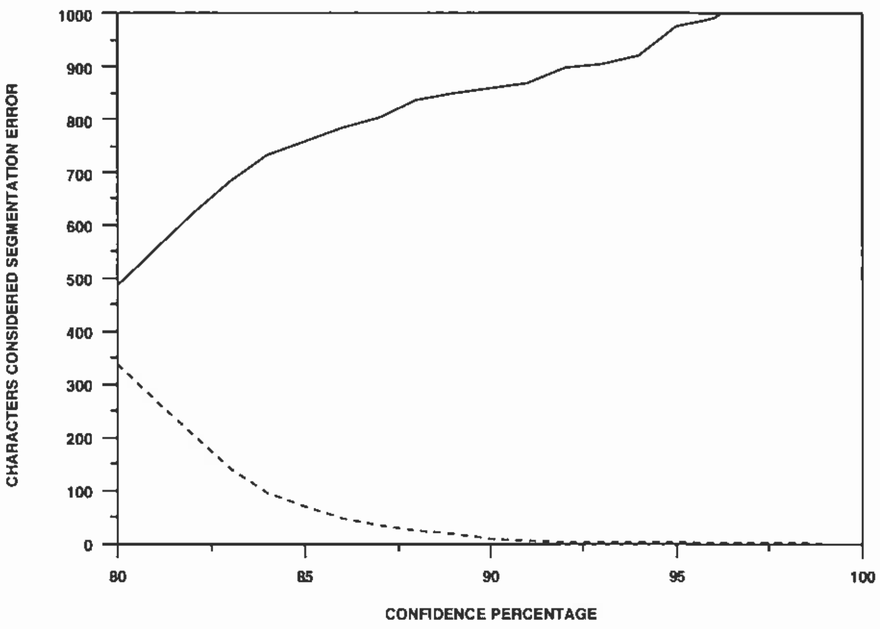


Figure 9: Tex12 page errors versus confidence and residual segmentation error

4 Conclusions

Since developers of multi-font recognition systems require large sets of segmented character images for training and testing, it is necessary to automate the collection of these large samples and thereby minimize the amount of human intervention required. A self-organized neural network based segmentation and classification method has been presented which uses a two-pass classification scheme. This method has been successfully used to generate databases of segmented machine printed character images along with their associated classifications without the use of context or human intervention. The success of this system hinges on its ability to effectively remove segmentation errors while concurrently maintaining a high accuracy on accepted classifications. This capability is primarily attributed to the system's ability, using FAUST, to adapt to the font type currently being processed.

Using this technique, accepted classification accuracies of 99.3% were achieved. In all cases studied, greater than 99% of the segmentation errors were detected. The experiments in this paper demonstrate how the confidence of the two-pass classification system can be tuned to achieve this high level of performance. It was also shown that, as the confidence is increased, the number of segmentation errors detected increases at the expense of rejecting a greater number of correctly segmented characters. Therefore, as the accuracy of the system is increased, effective system throughput is reduced, resulting in smaller data sets being automatically generated. In conclusion, accuracy has its cost.

References

[1] R. A. Wilkinson. Segmenting of Text Images with Massively Parallel Machines. In D. P. Casasent, editor, *Intelligent Robots and Computer Vision*, volume 1607, pages 312–323. SPIE, Boston, 1991.

[2] P. M. Flanders, R. L. Hellier, H. D. Jenkins, C. J. Pavelin, and S. Van Den Berghe. Efficient high-level programming on the AMT DAP. *IEEE Proceedings: Special Issue on Massively Parallel Computers*, 79(4):524–536, 4 1991.

[3] C. L. Wilson. A New Self-Organizing Neural Network Architecture for Parallel Multi-Map Pattern Recognition - FAUST. *Progress in Neural Networks*, 4, 1992. to be published.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

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

[9] M. D. Garris, R. A. Wilkinson, and C. L. Wilson. Analysis of a biologically motivated neural network for character recognition. In *Proceedings: Analysis of Neural Network Applications*, pages 160–175. ACM Press, May 1991.

[10] 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.

[11] 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.

[12] 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.

[13] T. Kohonen. *Self-Organization and Associative Memory*. Springer-Verlag, Berlin, second edition, 1988.

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

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

[16] J. Rubner and P. Tavan. A self organizing network for principal component analysis. *Europhysics Letters*, 10:693–698, 1989.

[17] C. L. Wilson. FAUST: a vision based neural network multi-map pattern recognition architecture. In *Proceedings: Applications of Artificial Neural Networks III*, Orlando, SPIE, April 1992.