Neural-Net Applications in Character Recognition and Document Analysis

L. D. Jackel, M.Y. Battista, J. Ben J. Bromley, C. J. C. Burges, H. S. Baird, E. Cosatto, J. S. Denker, H. P. Graf, H. P. Katseff, Y. LeCun, C. R. Nohl, E. Sackinger, J. H. Shamilian, T. Shoemaker, C. E. Stenard, B. I. Strom, R. Ting, T. Wood, and C. R. Zuraw

AT&T Bell Laboratories, Holmdel NJ 07733 USA

ABSTRACT

A proven strength of neural-network methods is their application to character recognition and document analysis. In this paper we describe a neural-net Optical Character Recognizer (OCR), neural-net preprocessing, and neural-net hardware accelerators that together comprise a high-performance character recognition system. We also describe applications in network-based fax and bit-mapped text processing.

1 Introduction

Character recognition has served as one of the principal proving grounds for neural-net methods and has emerged as one of the most successful applications of this technology. This chapter outlines optical character recognition / document analysis systems developed at AT&T Bell Labs that combine the strengths of machine-learning algorithms with high-speed, fine-grained parallel hardware. From our point of view, the most significant aspect of this work has been the efficient integration of diverse methods into end-to end systems. In this paper we use the task of locating and reading ZIP codes on US mail pieces as an illustration of the character recognition / document

analysis process. We will also describe other applications of the technology, including interpretation of faxed forms and bit-mapped text to ASCII conversion.

2 The Character Recognition Process

Figure 1. shows the "typical" character recognition process, which starts with an optical image and ultimately produces a symbolic interpretation. The process is divided into a series of tasks that are usually executed independently. It begins with image capture in which an optical image is converted to a bit-map. Next the region of interest, in this example the address block, is located. Then the desired field, the ZIP code, is found. This field is then usually size-normalized and sometimes (not shown here) de-slanted. Finally, the characters are segmented and recognized. Note that the recognition phase is only one step in a long process. In our systems we modify this model, using feedback from down-stream stages to influence up-stream decisions. We also apply neural-net hardware and algorithms where they are advantageous.

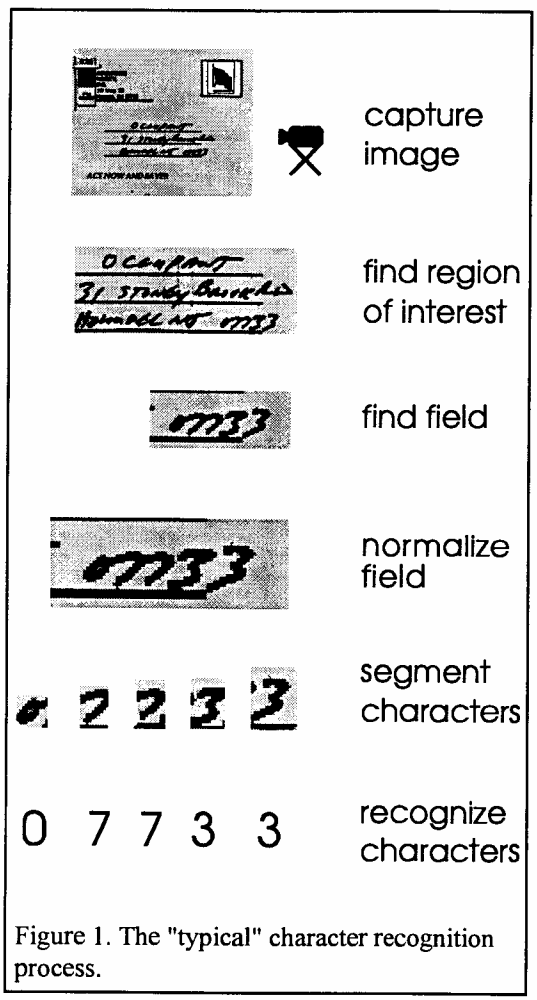

Figure 1. The "typical" character recognition process.

For a user, the end-to-end system performance is what counts. It matters little if the recognizer module is fast or accurate if most of the time budget and most of the errors arise from other modules. It has been our goal to design a system that has no "weak links" in either accuracy or speed. The technology driver for us has been a program sponsored by the US Postal Service whose goal is to produce an automatic address reading system that starts with bit-mapped images of envelopes and produces interpreted ZIP codes. Both

speed and accuracy are key issues. Most of this paper describes technology developed for this application. This kind of problem has also been addressed by other workers [1].

3 The Basic Recognizer: LeNet

At the core of our recognition system is an isolated character recognizer that we now call LeNet [2]. The LeNet architecture is shown in Figure 2. LeNet takes a 20 x 20 pixel field as input and returns a rank-ordered list of possible single-character interpretations of the input image, along with confidence scores for each interpretation.

LeNet is an example of a highly structured neural-net in which the structure seeks to incorporate *a priori* knowledge about the task domain. For the OCR task, this knowledge includes the local two-dimensional geometric relationships that exist in images. LeNet is designed to extract local geometric features from the input field in a way that preserves the approximate relative locations of these features. This is done by creating feature maps that are formed by convolving the image with local feature-extraction kernels. (An important feature of LeNet is that the feature extraction kernels are learned as opposed to being hand-crafted.) These maps are then spatially smoothed and sub-sampled; this latter step builds in invariance to small distortions of the input image [3,4]. In the same way, higher-level feature maps are extracted from the sub-sampled first-level maps. The higher level maps then provide input to a linear classification layer. Although the network has over 100,000 connections, the network structure imposes constraints so that only about 3,000 different weight values have to be learned.

LeNet has several advantages that make it attractive for recognizing characters when high variability (like we see in images of mailed envelopes) is expected. First, LeNet has state-of-the-art accuracy as illustrated by its strong performance in a competition sponsored by NIST , the US. National Institute of Standards and Technology [5]. Second, LeNet runs at reasonable speeds on standard hardware (~10 characters/sec on a workstation) and high speed (~1000 characters/sec) on specialized hardware. LeNet can also be readily trained to recognize new character styles and fonts. It works well for both handwritten and machine printed characters.

In general, in machine-learning tasks, best performance on test data is obtained by controlling the capacity of the learning machine to match the available training data. With this idea in mind, we have modified the architecture of LeNet, hence controlling capacity, depending on the amount of training

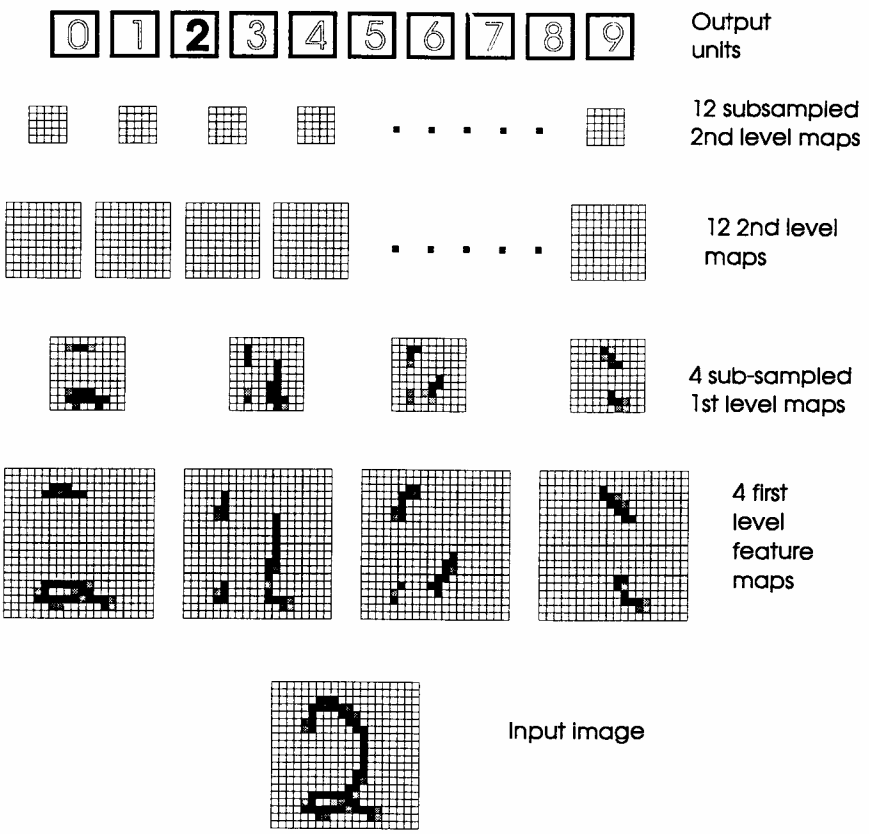

Figure 2. Architecture of LeNet. Each small box represents a neural-net "unit" or "neuron". The weighted connections between units (not shown in this figure) are highly structured. The maps are generated by convolutions with feature extraction kernels. Input images are sized to fit in a 20 x 20 pixel field, but enough blank pixels are added around the border of this field to avoid edge effects in the convolution calculations. In this figure, the activation of the input and the first two layers of the network are indicted: darker shading indicates greater activity.

data available. The version shown in Figure 2 was optimized for a training set of 7000 ZIP code digits. We found that a version with more hidden units and about twice as many weight values provided better results when we switched to a database of 50,000 digits. Even larger nets will be effective for bigger training data bases.

We note that there are other methods for OCR that may be more appropriate than LeNet for some applications. In particular, when our task is to read cleanly printed text with a limited range of fonts, a much simpler network

may give adequate performance. In this case, acceptable accuracy at very high recognition speeds can often be attained by simple template matching. The problems we address in this paper are those in which recognition accuracy is most important and where the quality of the input images or characters may be poor. It is in this regime that LeNet excels. We also note that recognition accuracy equaling or exceeding LeNet has been obtained with a sophisticated pattern matching technique that uses a special metric, known as "tangent distance", for comparing patterns [6]. Currently, this method lags LeNet in speed, but this situation may change as this new method evolves.

4 Segmentation

LeNet recognizes one character at time. If our objective is to recognize a string of characters, the string has to be cut up into individual characters, a process called segmentation. The difficulty of the segmentation task strongly depends on the quality and type of the string. If we have fixed-pitch machine-printed fonts `like this`, and if we can detect this condition, the task is straightforward. For cleanly-printed, variable-pitch machine fonts, the task is more difficult, although a connected components analysis can almost always identify individual characters. For machine printing of poor quality or for handwriting, where different characters might touch and single characters might be broken into several pieces, the task is very difficult. For these difficult cases, segmentation is like a "chicken and egg" problem: in order to recognize we need to segment, but in order to segment we may have to recognize [7,8,9,10,11].

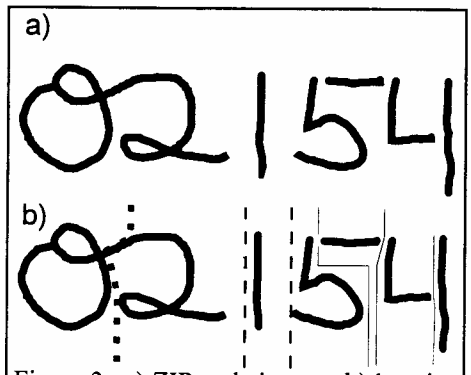

Figure 3. a) ZIP code image. b) location of definite cuts (dashed lines) and tentative cuts (dotted line and thin solid line).

Our current systems [7] take a hierarchical approach to a combined segmentation / recognition process. Our strategy is to find probable segmentation points or "cuts", snip out the "inked" regions between these points, and then see if LeNet can recognize these segments with high confidence, either in isolation, or in combination with neighboring segments. We then choose the set of segment combinations that gives highest overall confidence while accounting for all the ink in the image.

We proceed in the following way: given a string, we first find the "definite" cuts between characters. These are places where there are substantial hori-

zontal gaps in the "inked" image. For the ZIP code image shown in Figure 3, definite cuts occur between the "2" and the "1" and between the "1" and the "5". We denote them by the vertical dashed lines shown in Figure 3b. Then we consider "tentative cuts" where a connected components analysis locates gaps between blobs of ink. In Figure 3, such cuts are within the "5" and the "4" and between the "5" and "4". These tentative cuts are shown as thin, solid lines in Figure 3b. Finally, using heuristic rules, we identify additional tentative cuts at places where characters are likely to be joined or touching. Such a tentative cut is shown as the dotted line in Figure 3b.

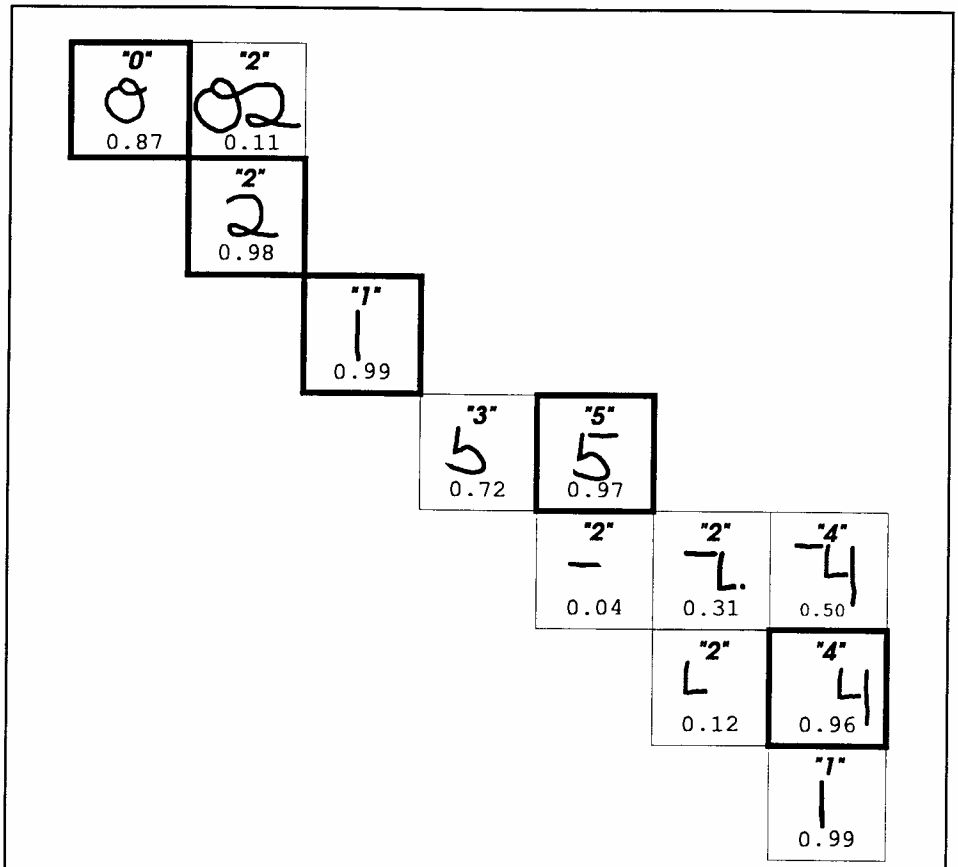

Figure 4. Blocks of segments from Figure 3 that are passed to LeNet. The number near the top of each box is the most likely classification of the "ink" in the box. Blocks in the same row are bounded on the left by the same cut; blocks in the same column are bounded on the right by the same cut. The number near the bottom is LeNet's relative confidence in the classification. In order to choose a consistent segmentation of the image we must ensure that all the "ink" is accounted for, and that it is only used once. Using this rule, the highest overall confidence classification / segmentation is indicated by the boxes with the heavy borders.

The next step in our segmentation process is to pass segments and possible segment combinations to LeNet for scoring as possible characters. Figure 4. shows these segments and segment combinations along with their top scoring classification and confidence level. The number near the top of each box is the most likely classification of the "ink" in the box. Blocks in the same row are bounded on the left by the same cut; blocks in the same column are bounded on the right by the same cut. The number near the bottom is LeNet's relative confidence in the classification. In order to choose a consistent segmentation of the image we must ensure that all the "ink" is accounted for, and that it is only used once. Using this rule, the highest overall confidence classification / segmentation is indicted by the boxes with the heavy borders.

Notice that for the example in Figure 3, in order to recognize and segment a 5 digit ZIP code we had to make 12 calls to LeNet. For 5 digit ZIP code images with no large blobs of extraneous ink, we have to make an average of 7.5 calls. In actual mail streams we expect more calls will be necessary. This places additional demands on the required speed of the recognizer engine and further motivates the use of special purpose hardware to implement LeNet.

A hardware system that has been effective in speeding the recognition / segmentation process is one based on the ANNA neural-net chip [12]. This chip, which mixes analog and digital processing, was specifically designed to speed evaluations of networks like LeNet. A key feature of ANNA's design is the provision for parallel evaluation of non-linear 2-dimensional convolutions, which represent the bulk of the computing required by LeNet. Because LeNet is large (over 100,000 connections), it cannot be evaluated entirely in parallel by ANNA. Instead, sections of the input image are sequentially evaluated, with the corresponding sections of the feature maps being evaluated in parallel. In order for ANNA, (or most any neural-net chip) to run efficiently, heavy system demands are placed in the management of data on and off the chip. The current board-level ANNA system has special hardware to speed I/O operations along with a custom sequencer to control ANNA's instruction execution. The latest versions of this system can evaluate LeNet in about 1 msec, so that, accounting for multiple calls to LeNet during the recognition/segmentation process, a throughput of over 400 character recognitions/sec can be sustained. This speed is about 25 times faster than a state-of-the-art workstation.

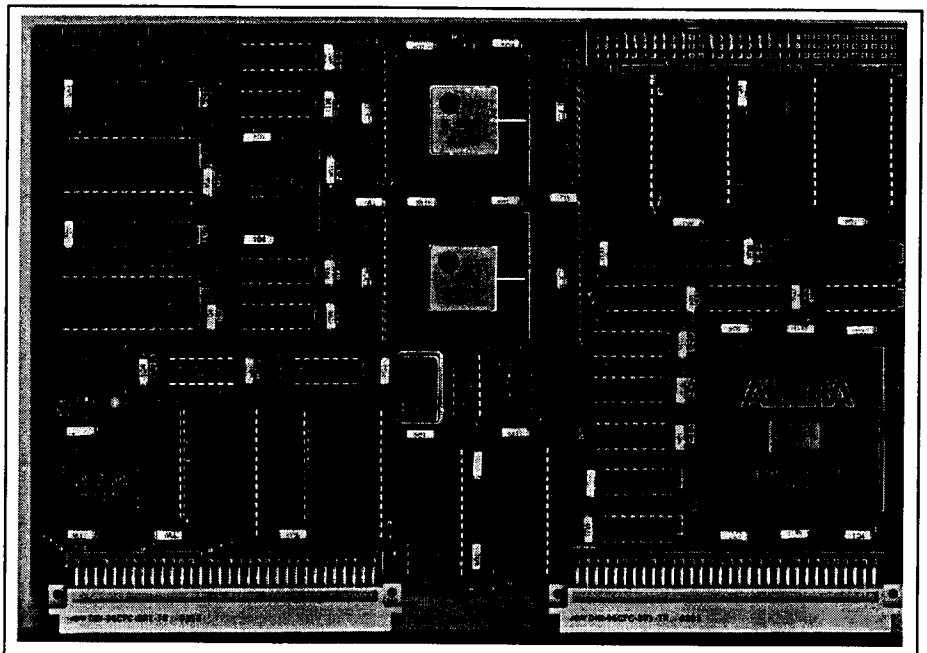

Figure 5. NET32K board-level system. The board contains two NET32K chips, along with a sequencer and circuitry to provide high-speed data input/output paths to the NET32K chips. In image processing applications this board sustains over 100 billion multiply/add operations per second.

5 Normalization

Recognition accuracy can be increased if we know that characters presented to the recognizer engine are limited in their range of size and orientation. This can be accomplished by normalizing the character strings with respect to size and slant angle. We found that while size normalization could be done quickly on a standard work station, slant normalization could not. In this de-slanting process, the overall slant in a string must be detected. Then the image bit-map of the string is modified so that the overall slant is set to zero. Here the most computationally intensive step is the measurement of the string slant angle. We have found that this potential speed bottleneck can be eliminated by using a second neural-net board-level system. This system is based on Hans Peter Graf's NET32K chip [13], which like ANNA mixes analog and digital processing, but unlike ANNA, NET32K has more stored weights (up to 32K vs. 4K for ANNA) at the expense of decreased accuracy (1 bit vs. 6 bits). NET32K excels at scanning relatively large images with large kernels (up to 16 x 16). A working board-level NET32K system, now in use, contains two NET32K chips, as well as custom sequencers and on-

board memory. This system, which is shown in Figure 5, is designed to support a high I/O rate for the NET32K chips. In the applications described below, this system achieves a sustained rate of 100 billion multiply-adds per second at 1.5-bit precision. To our knowledge this is the highest processing rate yet attained in any single-board image processing system.

In order to measure average image slant, the NET32K system scans the image field with a set of oriented edge detector kernels, with each kernel tuned for a particular edge orientation. Examples of these kernels and detected feature maps are shown in Figure 6. After the scan is completed, we count how many places each kernel matched in a section of the image. The kernel that scored the most matches indicates the dominant slant angle in the image. With NET32K, we can find the slant angle of a ZIP code string in about 20 msecs.

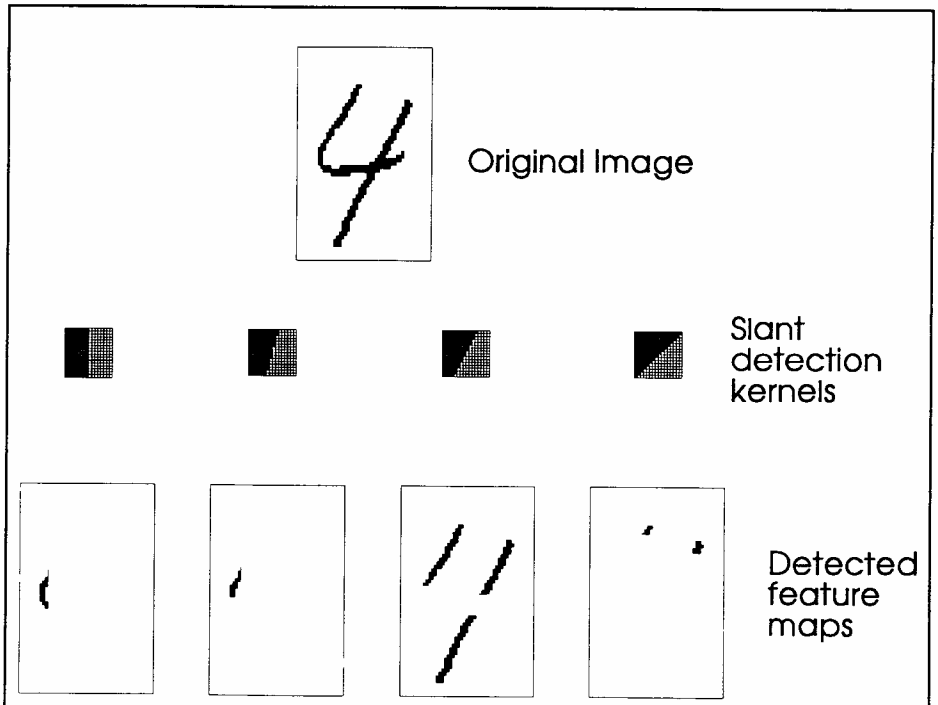

Figure 6. Detection of slant angle using a set of slant detection kernels. A set a 16 kernels (only 4 are shown in this figure) are scanned across the image. For the kernels shown here, a dark shading indicates a region where the required match is white space, and a lighter shading indicates a region where the required match is inked. Feature maps for each kernel record where matches are found in the image. The kernel that scores the most matches across the image indicates the predominant slant.

6 Finding the Region of Interest

Locating the region in an image that contains the desired text can be very challenging, especially if the image is cluttered with extraneous text, graphics, and/or background noise, e.g. Figure 7. Because the images typically contain millions of pixels and because techniques for image analysis usually require many operations per pixel, field location is a computationally intensive task. We have used custom hardware to find address blocks at a rate of more than 10 images/sec. Here again, NET32K shows its effectiveness. The system scans the input image with feature detection kernels that are tuned to the characteristics of text lines. Using the resultant feature maps and our *a priori* knowledge about where address blocks are likely to occur on an envelope, we can find likely candidate fields at the required rate. All the processes described above are now being integrated into a complete end-to-end system.

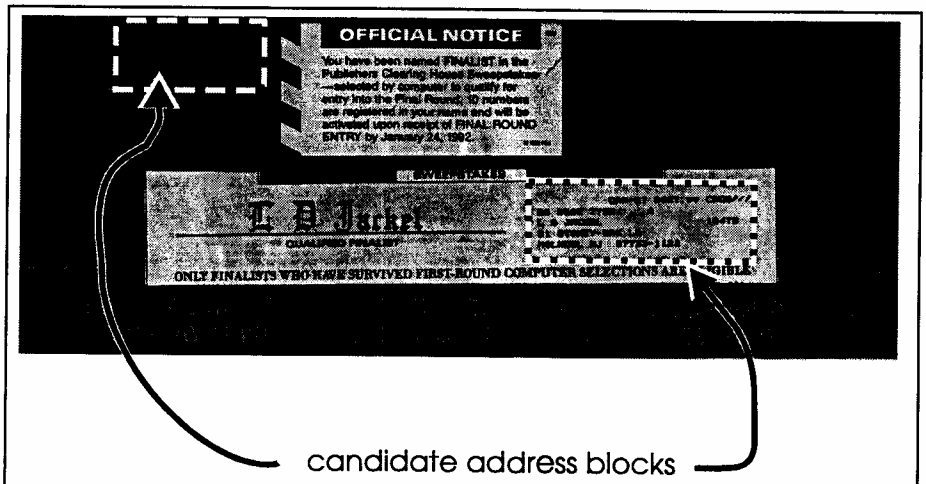

Figure 7. An example of address block location performed by the NET32K system on a particularly challenging mail piece. Two candidate address blocks were found; rules about the likely location of address blocks makes the candidate on the lower right the first choice. The NET32K system can sustain a processing rate of more than 10 images/sec.

7 Additional Applications

The technology developed to solve postal tasks has been successfully built upon and used in telecommunication applications. In this section several of these applications are discussed.

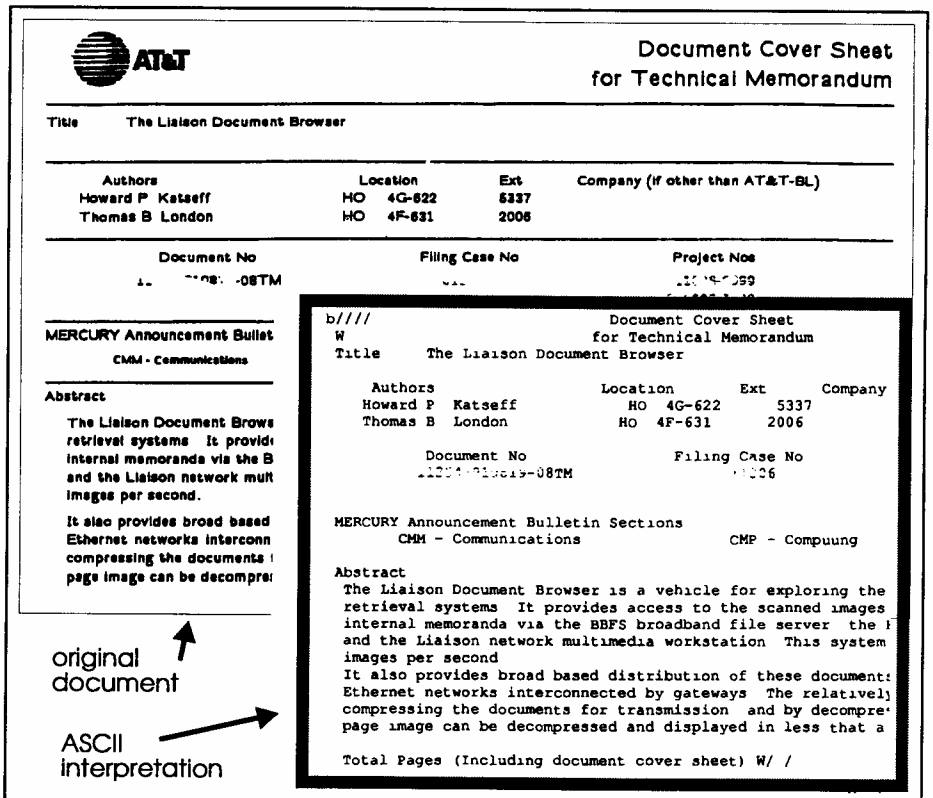

Figure 8. Conversion of bit-mapped documents to ASCII. Using a combination of page-layout analysis and LeNet, images from an internal AT&T document retrieval system have been converted to ASCII text. Part of a bit-mapped document is shown in the background above. The overlay is part of that document's ASCII interpretation. The ASCII text allows word or phrase searching as well as lifting of the text for further word processing.

Bit-Map to ASCII Conversion for Document Retrieval

A document retrieval system that allows users to browse internal technical publications has been in service at AT&T Bell Laboratories for more than a year [14]. The system displays bit-map images of document pages on users' workstation screens. It is now being upgraded to provide an ASCII version of the text as a companion of the bit-map. This allows users to search for words or phrases and to lift sections of text for inclusion in other documents. The bit-map images include many fonts and vary in their image quality.

To obtain the ASCII version of a document, page-layout analysis is first performed using a software package developed by H. S. Baird [15]. This package finds text blocks, and then segments out individual characters. These clipped character images are then recognized using a version of LeNet that was trained on numerous printed fonts and sizes, including character examples that were corrupted with synthetic noise with characteristics similar to those encountered in actual documents [16]. Overall OCR accuracy for this system typically exceeds 99%. An example of an original document and its ASCII version are shown in Figure 8.

Processing of Faxed Forms

As a further example of an application of recognition technology, we describe a system, deployed internally in AT&T, that automates processing of new service orders for parts of the AT&T network. A block diagram of the system is shown in Figure 9. In this system, a client requesting a new service faxes a form to a central AT&T facility. There, the bit-map is first used to identify the type of form. Next, registration marks on the image are located and the image is adjusted to compensate for distortions generated by the fax-scanning process. The image fields that specify the order requisitioner and the ordering information are then clipped out, normalized and passed to LeNet. The forms are designed so that the characters to be processed are written in boxes, eliminating the need for segmentation.

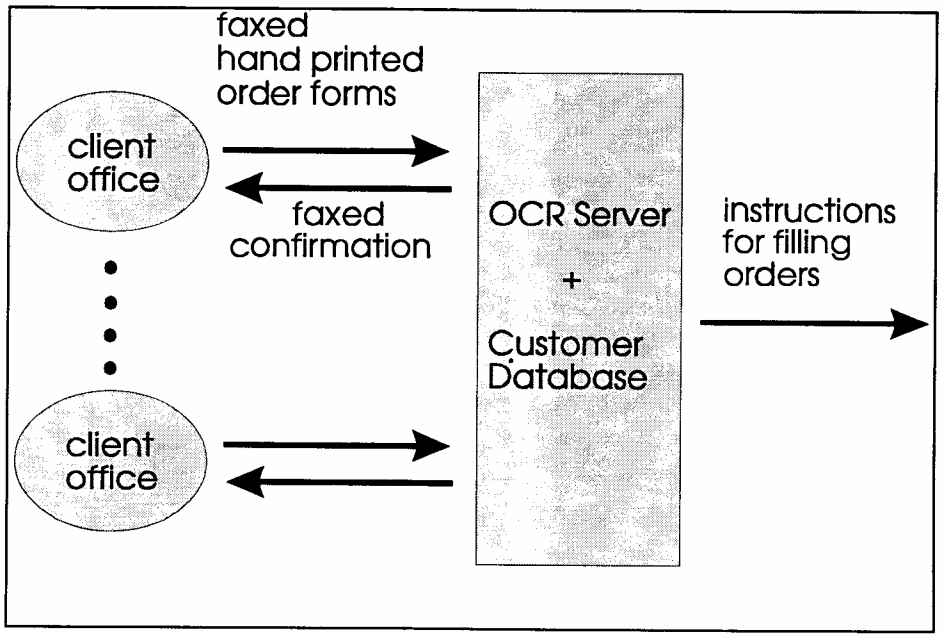

Figure 9. Block diagram of automatic order processing system. Clients fax order forms to a central facility where OCR is performed and the instructions for filling the order are then generated. A confirmation notice is faxed back to the client.

For this task the considerable contextual information available is used to maintain high recognition accuracy. As examples, the requisitioner's name and organization can be cross-checked against a database and the service order can be matched against allowable service codes After this cross-checking, if LeNet still has low confidence in a particular field, that field is passed to a human correction station operator who makes the final decision. This system was successfully deployed in 1992 and is now in everyday use.

Processing of Tabular Text

Another application of the document processing technology is being used by an AT&T Data Center in Kansas City. For this application very high recognition accuracy is essential. First introduced in 1992, this application translates large volumes of densely printed tabular text from scanned documents into structured ASCII format. Typically, the printed text is one of a large variety of small machine printed fonts, and may be poorly registered on the document. The document pages (either original or copy) are scanned in by document type and passed to the AT&T document analysis system, which uses Baird's page-layout analysis. The system locates text strings according to a user-defined template that describes fields within text lines while ignoring other text. Then using fast and accurate algorithms invented by David Itner [17] for fixed-pitch printed text , such as from impact printers, each line is parsed into fields according to the user-defined template. The fields are then passed on to the neural-net character recognizer. After performing a contextual analysis, 99.9% accuracy is achieved. Any low-confidence characters are marked for review by human operators.

8 Conclusions

In this paper we have described some examples of applications of neural-net character recognition and document analysis that have been developed at AT&T Bell Labs. We have concentrated on a system designed to find and read ZIP codes on envelopes for the US Postal Service. In order to meet real-time requirements, this system includes special purpose hardware with neural-net chips. The system also uses a combined approach to the interdependent problems of recognition and segmentation. We have also sketched applications to document retrieval and to automatic processing of faxed forms.

References

1. For example, see G. Martin, M. Rashid, D. Chapman, J. Pittman, "Learning to See Where and What: Training a Net to Make Saccades and Recognize Handwritten Characters," in *Advances in Neural Information Processing* **5**, (NIPS 92) S.J. Hanson, J.D. Cowan, and C.L. Giles eds., pages 441-447, Morgan Kaufmann (1993).

2. Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, "Handwritten Digit Recognition with a Back-propagation Network," in *Advances in Neural Information Processing* **2**, (NIPS 89) D. S. Touretzky ed., pages 396-404, Morgan Kaufmann (1990).

3 . K. Fukushima "Neocognitron: A Self-Organizing Neural Network Model For a Mechanism of Pattern Recognition Unaffected by a Shift in Position," *Biol. Cybernetics* **36**, 193-202 (1980).

4. M. C. Mozer, "Early Parallel Processing in Reading: A Connectionist Approach," in *Attention and Performance, XII: The Psychology of Reading*, M. Coltheart, ed., **Vol. XII**, pages 83-104, Erlbaum, Hillsdale, NY (1987).

5. See Proc. of the First Census Optical Character Recognition System Conf., **NISTIR 4912,** August 1992.

6. P. Simard, Y. LeCun, and J. S. Denker, "Efficient Pattern Recognition Using a New Transformation Distance," in *Advances in Neural Information Processing* **5**, (NIPS 92) S.J. Hanson, J.D. Cowan, and C.L. Giles eds., pages 50-58, Morgan Kaufmann (1993).

7. C. J. C. Burges, O. Matan, Y. LeCun, J. S. Denker, L. D. Jackel, C. E. Stenard, C. R. Nohl, J. I. Ben, "Shortest Path Segmentation: A Method for Training a Neural Network to Recognize Character Strings." *in Proc. of IJCNN, International Joint Conference on Neural Networks*, Baltimore MD, **Vol. III**, pages 165- 170 (1992).

8. J. Keeler, D. Rumelhart, and W. K. Leow," Integrated Segmentation and Recognition of Handprinted Numerals", in *Advances in Neural Information Processing* **3**, (NIPS 90) R. P. Lippmann, J. E. Moody, and D. Touretzky eds., pages 557-563, Morgan Kaufmann (1991).

9. O. Matan, C.J.C. Burges, Y. LeCun, and J. S. Denker ," Multi-Digit Recognition Using a Space Displacement Neural Network," in *Advances in Neural Information Processing* **4**, (NIPS 91) J. E. Moody, S.J. Hanson, and R. P. Lippmann eds., pages 488-495, Morgan Kaufmann (1992).

10. G. L. Martin and M. Rashid, "Recognizing Overlapping Hand-Printed Characters by Centered-Object Integrated Segmentation and Recognition," in *Advances in Neural Information Processing* **4**, (NIPS 91) J. E. Moody, S.J. Hanson, and R. P. Lippmann eds., pages 504-511, Morgan Kaufmann (1992).

11. J. Keeler and D.E. Rumelhart, "A Self-Organizing Integrated Segmentation and Recognition Neural-Net," in *Advances in Neural Information Processing* **4**, (NIPS 91) J. E. Moody, S.J. Hanson, and R. P. Lippmann eds., pages 496-503, Morgan Kaufmann (1992).

12. E. Sackinger, B. E. Boser, and L. D. Jackel, "A Neurocomputer Board Based on the ANNA Neural Network Chip," in *Advances in Neural Information Processing* **4**, (NIPS 91) J. E. Moody, S.J. Hanson, and R. P. Lippmann eds., pages 773-780, Morgan Kaufmann (1992).

13. H. P. Graf, C. R. Nohl, and J. Ben, "Image Recognition with an Analog Neural Net Chip," *in Proc. 11th IAPR Int. Conf. Pattern Recognition,* **4**, pages 11-15 (1992).

14. H. P. Katseff and T. B London, "The Ferret Document Browser", to appear in *Proc. of USENIX Summer 1993 Tech Conf.* Cincinnati, June 1993. The Ferret browser is now being combined with the "Right Pages " electronic library system; see G. A. Story, L. O'Gorman, D. Fox, L. Schaper, and H. V. Jagadish, "The Right Pages Image-Based Electronic Library for Alerting and Browsing," IEEE Computer, pages 17-26, Sept. 1992.

15. H. S. Baird, "Global-to-Local Layout Analysis," *in Proc. IAPR Workshop on Syntactic and Structural Pattern Recognition,* Pont-a-Mousson, France 12-14 September 1988.

16. H. S. Baird, "Document Image Defect Models" *in Structured Document Image Analysis,* H. S. Baird, H. Bunke, & K. Yamamoto eds. Springer-Verlag, New York (1992).

17. D.J. Itner and H. S. Baird, "Language-Free Layout Analysis," i*n Proc. of 1993 International Conf. on Document Analysis and Recognition,* Tsukuba Science City, Japan, October 1993.