

Using Neural Networks to Create an Adaptive Character Recognition System

Alexander J. Faaborg
Cornell University, Ithaca NY

(May 14, 2002)

Abstract — A back-propagation neural network with one hidden layer was used to create an adaptive character recognition system. The system was trained and evaluated with printed text, as well as several different forms of handwriting provided by both male and female participants. Experiments tested (1) the effect of set size on recognition accuracy with printed text, and (2) the effect of handwriting style on recognition accuracy. Results showed reduced accuracy in recognizing printed text when differentiating between more than 12 characters. The handwriting style of the subjects had varying and drastic effects on recognition accuracy which illuminated some of the problems with the systems character encoding.

INTRODUCTION

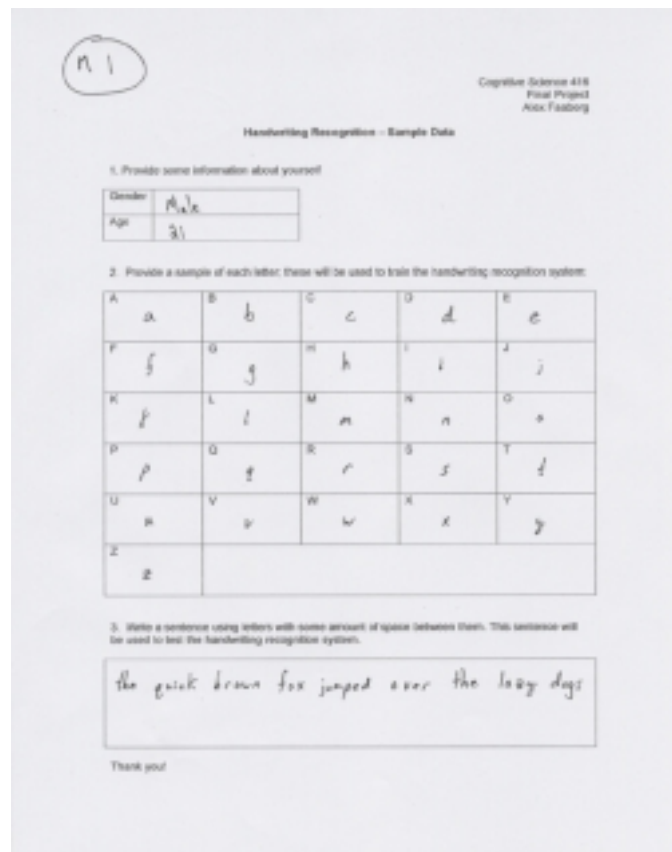
One of the most important effects the field of Cognitive Science can have on the field of Computer Science is the development of technologies that make our tools more human. A very relevant present-day field of natural interface research is hand writing recognition technology. Evidenced by the fact that we are not currently all using Tablet computers, accurate hand writing recognition is clearly a difficult problem to solve. For a system to be considered acceptable it must be extraordinarily good. An analysis of user acceptance of handwriting technology performed by IBM showed that an accuracy rating under 97% was considered unacceptable by most users. (LaLomia). This need for accuracy is so strong that it has even caused millions of people to learn an entirely new way to write, a way that is easier for computers to detect. Unistroke recognition algorithms, like the popular Graphiti used on Palm devices require the user to adapt instead of the device, essentially the antithesis of natural interface design. The technology has proven to be very accurate: "Each character is written with a single stroke. This solves the character level segmentation problem that previously plagued handwriting recognition. The curve drawn between pen down and pen up events can be recognized in isolation. Unistroke recognition algorithms can be relatively simple because there is no need to decide which parts of the curve belong to which character or to wait for more strokes that belong to the same character as is often the case when we try to recognize conventional handwriting." (Isokoski). To develop a handwriting recognition system that is both as reliable as Unistroke, and natural enough to be comfortable, the system must be highly adaptable. Creating software that is as adaptable

as its users are unique is a very challenging problem for conventional computer algorithms. This is why many people in the field of handwriting recognition are turning to neural networks to perform the recognition processing. Adaptable by their very nature, neural networks can bring to the computing world software that molds and conforms in ways algorithms like Unistroke never could. James Pittman notes that already “they have proven to be very robust in the face of the high variability in handwriting.” (Pittman). Pittman describes several ways neural networks can be used to recognize handwritten text. These include analyzing 2D image input, analyzing stroke sequence, and using context to combine results from both approaches into one network. This project will implement neural networks to focus on the image input side of handwriting recognition systems. Experiments will study the effect of set size on recognition accuracy with printed text, and the effect of handwriting style on recognition accuracy. Both experiments aim to show the neural network’s core ability to adapt to variable stimuli.

METHOD

Subjects and Data

Handwriting samples were obtained from 4 subjects, two male and two female. All the subjects were undergraduate students at Cornell University between the ages of 20 and 21. The subjects were given a piece of paper with boxes for each letter of the alphabet, and a large box to write the sentence “The quick brown fox jumped over the lazy dogs.” All the subjects used identical felt tip pens. In addition to the handwriting samples, one printed sample was created using Microsoft word with the font Arial 12pt, in all capital letters.



A handwritten form titled "Handwriting Recognition - Sample Data" is shown. At the top left, the number "11" is circled. The form includes a header with "Cognitive Science 478", "Final Project", and "Alex Falborg". It contains three sections: 1. Personal information (Gender: Male, Age: 21), 2. A 5x5 grid of lowercase letters A-Z with handwritten samples, and 3. A sentence "the quick brown fox jumped over the lazy dogs" written in cursive. The form ends with "Thank you!"

Cognitive Science 478
Final Project
Alex Falborg

Handwriting Recognition - Sample Data

1. Provide some information about yourself

Gender	Male
Age	21

2. Provide a sample of each letter; these will be used to train the handwriting recognition system.

A	a	B	b	C	c	D	d	E	e
F	f	G	g	H	h	I	i	J	j
K	k	L	l	M	m	N	n	O	o
P	p	Q	q	R	r	S	s	T	t
U	u	V	v	W	w	X	x	Y	y
Z	z								

3. Write a sentence using letters with some amount of space between them. This sentence will be used to test the handwriting recognition system.

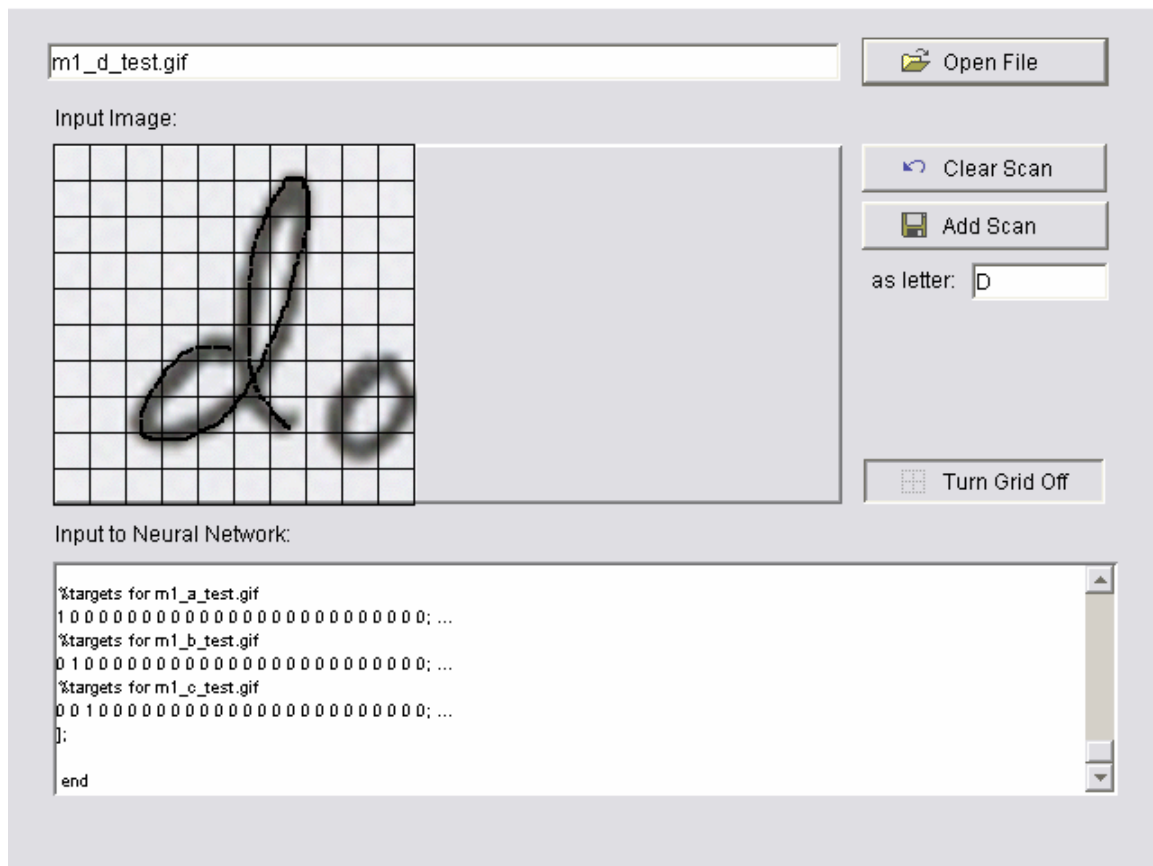
the quick brown fox jumped over the lazy dogs

Thank you!

These pages were then scanned so that the characters could be analyzed by the neural network.

Translating the Input into Matlab Code

Converting the scanned characters to code readable by Matlab was achieved with a Java application. The application allowed the user to load in an image, and then convert that image into Matlab code by painting over it with the mouse. The application was designed to take mouse input as opposed to directly scanning the image so that it could also capture temporal information about the user's strokes. However, there was unfortunately not enough time to implement this feature. While the source images were at a high resolution, they were imported into the Matlab code at a resolution of 10 by 10 (represented as a 100 unit vector) to reduce the processing time of the neural network. The black grid in the interface visually depicts this lower resolution.



Above, the user has loaded a file, and has painted over it to register the active pixels. Pressing the save button will add this image's data to the growing amount of Matlab code in the bottom text box. Each input must also be associated with a letter of the alphabet; this is specified in the interface using the small text box under the “Add Scan” button to specify the appropriate letter. This target letter appears in the Matlab code as a 26 unit vector containing one 1 and twenty-five 0s. Here is an example of a file created using this application, containing information about only one image, c_a.gif.

```
function matrix = faaborgFinal_getData(request)
```

```
if request == 0
```

```
    matrix = [...
```

```
    %%%%%%%%%%%
```

```
    % c_a.gif
```

```
    %%%%%%%%%%%
```

```
    %input 0 for c_a.gif
```

```
    0 0 0 0 0 0 0 0 0 0 ...
```

```
    0 0 0 0 1 0 0 0 0 0 ...
```

```
    0 0 0 1 1 1 0 0 0 0 ...
```

```
    0 0 0 1 0 1 0 0 0 0 ...
```

```
    0 0 1 1 0 1 1 0 0 0 ...
```

```
    0 0 1 1 1 1 1 0 0 0 ...
```

```
    0 0 1 0 0 0 1 0 0 0 ...
```

```
    0 1 1 0 0 0 1 1 0 0 ...
```

```
    0 1 0 0 0 0 0 1 0 0 ...
```

```
    0 0 0 0 0 0 0 0 0 0 ...
```

```
    ;...
```

```
    %input 1 for c_a.gif
```

```
    0 0 0 0 0 0 0 0 0 0 ...
```

```
    0 0 0 0 1 1 0 0 0 0 ...
```

```
    0 0 0 1 1 1 0 0 0 0 ...
```

```
    0 0 0 1 0 1 1 0 0 0 ...
```

```
    0 0 0 1 0 0 1 0 0 0 ...
```

```
    0 0 1 1 1 1 1 0 0 0 ...
```

```
    0 0 1 0 0 0 1 1 0 0 ...
```

```
    0 1 1 0 0 0 0 1 0 0 ...
```

```
    0 1 0 0 0 0 0 1 0 0 ...
```

```
    0 0 0 0 0 0 0 0 0 0 ...
```

```
    ;...
```

```
];
```

```
end
```

```
if request == 1
```

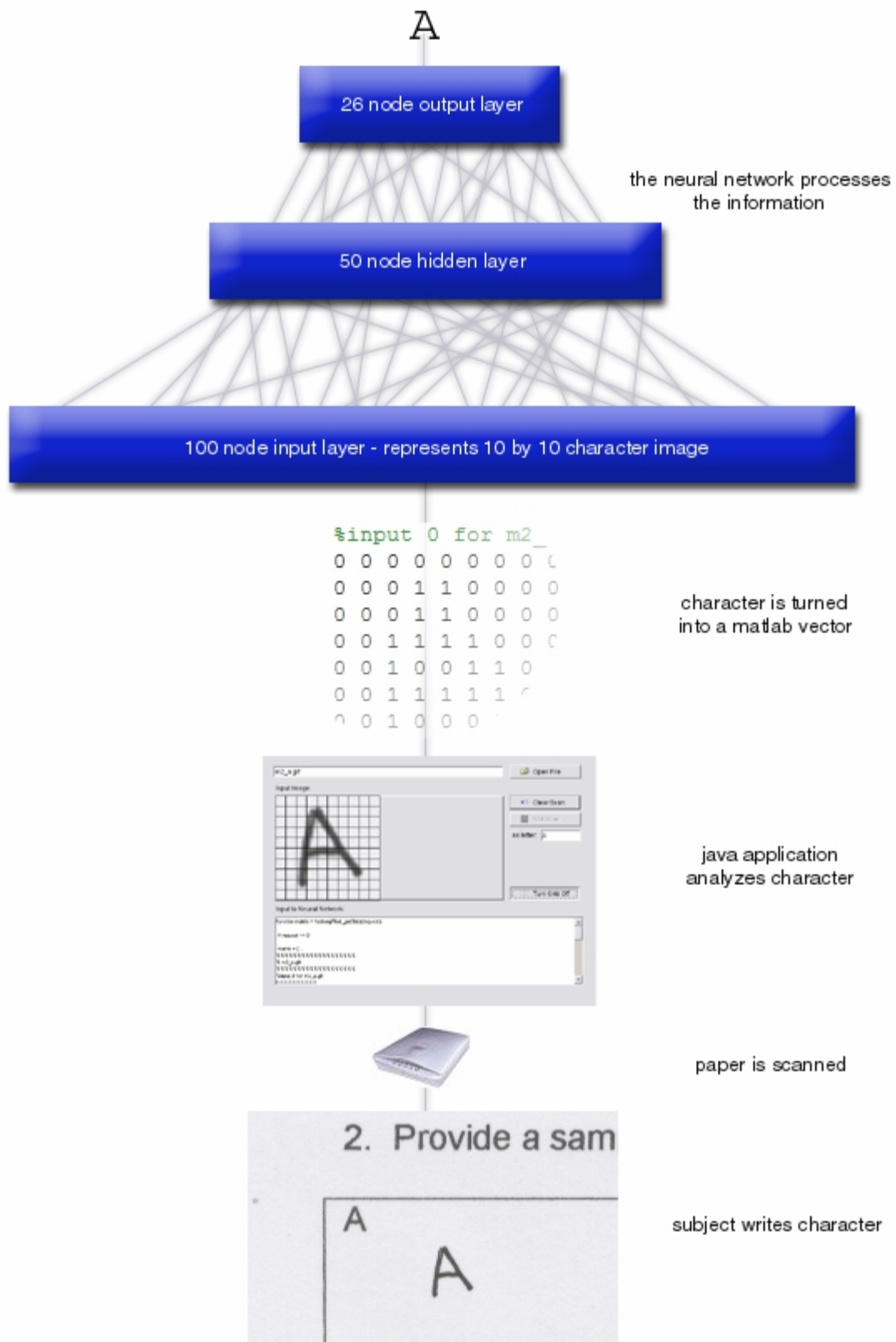
```
matrix = [...  
%targets for c_a.gif  
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0; ...  
1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0; ...  
];  
  
end
```

This code can then be copied out of the text box and into an m file, and the neural network will be able to access a matrix of all the character images in the file by calling `faaborgFinal_getData(0)`, and a matrix of their related target letters, by calling `faaborgFinal_getData(1)`. Copying new generated code into these files allows the network to be quickly modified to test a new set of character information. This easy input switch was important because the experiments involved 5 character sets, containing more than 230 image vectors.

Notice that in the code above, there are two 10 by 10 representations of the same image file. This is done to make the network more adaptable. In addition to directly painting over the character, the image can be used as a guide for painting the same character in a slightly different position or angle. In all of the trials there were three vectors for every image file, each containing a slightly different representation of the character portrayed in the image.

Neural Network Design

The neural network had three layers: an input layer consisting of 100 nodes (for the 10 by 10 character input), a hidden layer consisting of 50 nodes, and an output layer with 26 nodes (one for each letter). The network uses back-propagation in addition to bias weights and momentum.



Explanation of Matlab Files

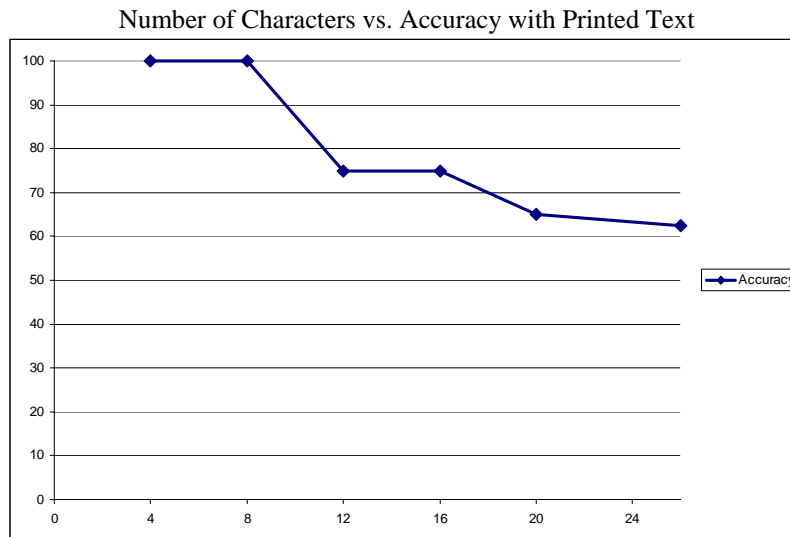
faaborgFinal.m	The neural network. Executing this file from the command line will begin training the network.
faaborgFinal_test.m	Tests the accuracy of the network. Executing this file from the command line will output: actual - recognized a - a b - b c - c d - d for the test set in faaborgFinal_getTestdata.m. Here the test set is the first 4 letters.
faaborgFinal_getData.m	The character set that the network trains on. Replace this file with new character sets to test different types of handwriting. Called by faaborgFinal.m
faaborgFinal_getTestData.m	The character set to test the accuracy of the network. These are characters the network has never seen before. Also replace this file when loading in a new character set. Called by faaborgFinal_test.m
faaborgFinal_squash.m	The squashing function. Called by faaborgFinal.m
faaborgFinal_parse.m	Outputs a letter on the screen for a corresponding output vector. Called by faaborgFinal_test.m
/sets	Contains files that can replace faaborgFinal_getData.m and faaborgFinal_getTestData.m for the 5 character sets the network was exposed to in testing.

RESULTS

[Note: Complete results with lists of exactly which letters were incorrect during each trial can be found in Appendix II]

The Effect of Set Size on Character Recognition

The first test was to determine how many letters the network could be exposed to before it experienced a degradation in recognition accuracy. For this test, the printed text in Arial font was used since these characters represented the simplest input available. Many of the trials took in excess of 30 minutes to train. Accuracy was tested by exposing the network to a set of characters after it had completed training.



The Printed Text:

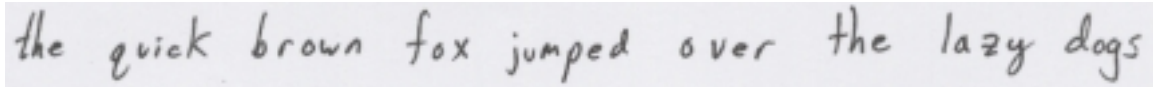
THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS

Trial	Epochs	MSE	Accuracy (percent)
First 4 letters	314	0.0042	100
First 8 letters	1714	0.0094	100
First 12 letters	10000 (max)	0.0279	75
First 16 letters	5465	0.03	75
First 20 letters	2526	0.029	65
All 26 letters	10000 (max)	0.038	62.5

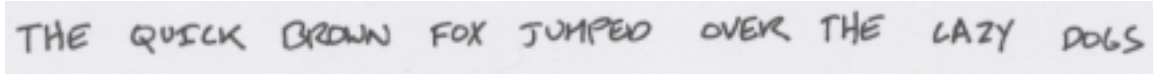
The Effect of Handwriting Style on Character Recognition

To test the effect handwriting style has on character recognition with this system, samples from the four subjects were scanned and converted to a series of Matlab vectors. To test in an environment where 100% accuracy was obtainable, only the first 8 letters of each sample were used. This also reduced the amount of time and processing power needed to run the experiment. Each character image was converted to a Matlab vector three times, each time in a slightly different position. Letters from the sentence in the handwriting sample were used to create the test set to determine accuracy. The neural network was not exposed to the test set during training.

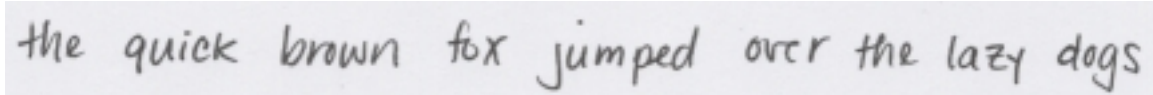
Male 1:

A handwritten sample from Male 1 showing the sentence "the quick brown fox jumped over the lazy dogs" in a cursive script.

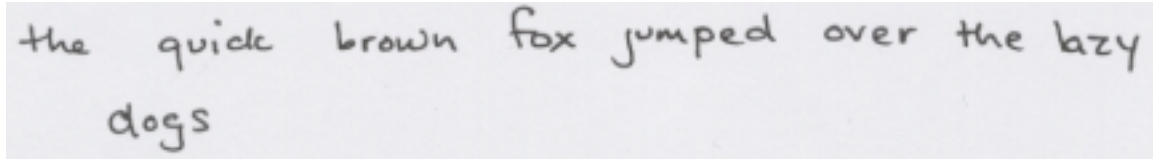
Male 2:

A handwritten sample from Male 2 showing the sentence "THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS" in all caps, using a blocky, uppercase script.

Female 1:

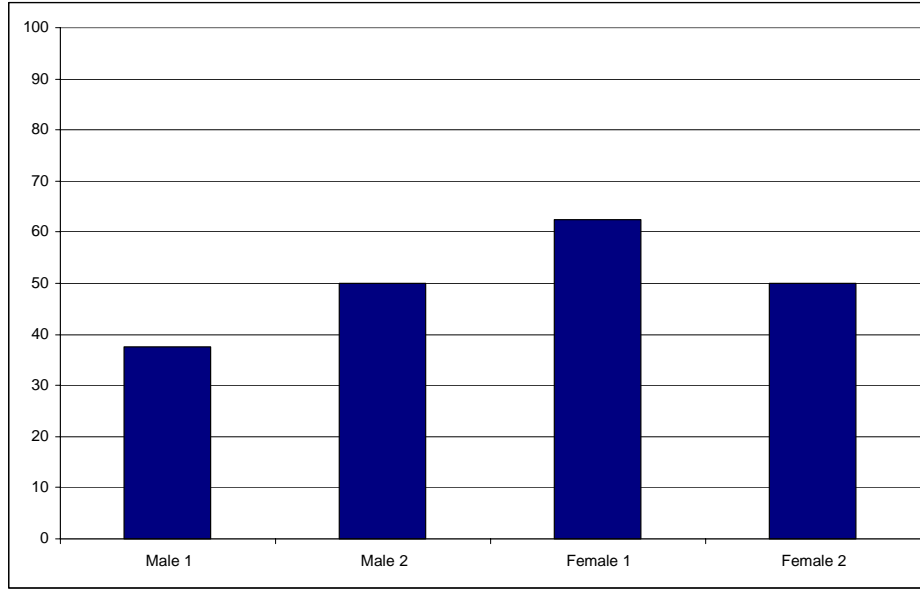
A handwritten sample from Female 1 showing the sentence "the quick brown fox jumped over the lazy dogs" in a cursive script, similar to Male 1.

Female 2:

A handwritten sample from Female 2 showing the sentence "the quick brown fox jumped over the lazy dogs" in a cursive script, with the word "dogs" on a new line.

[Note: These are characters used to test accuracy, images of every character input into the system can be found in Appendix I.]

Handwriting Style vs. Recognition Accuracy



Experiment	Epochs	MSE	Accuracy
Female 1	1225	0.0091	62.5
Female 2	417	0.0099	50
Male 1	10000 (max)	0.0139	37.5
Male 2	3687	0.0097	50

There is a significant drop in accuracy compared to when the system was analyzing printed text: the highest accuracy in this experiment is equivalent with the lowest accuracy in the last experiment. Some of the problems, particularly the reason for Male 1's low score, are discussed in the next section.

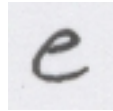
DISCUSSION

Weaknesses and Limitations of Experiment

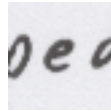
In the first experiment, the test set of images was the same as the training set of images, which is logical since the characters themselves are identical. This was the only time during this project where the two sets were shared; however some variation was presented by encoding the images into Matlab code again with the Java application to create the test set. The rationale for doing this was assuming that the hypothetical software automatically cropping the letters (for this project the letters were cropped manually), would be able to crop consistently.

In the second experiment the network was tested on images that it had never seen before. Looking at the handwriting samples subjectively (see Appendix I), it appears the biggest factor on accuracy was variations in letter size. Male 1's test letters were often much smaller than the training letters he provided:

Train: m1_e.gif



Test: m1_e_test.gif



While the network was able to handle changes in rotation and position by encoding each image as a number of different vectors when generating the Matlab code, the size of the character was fixed. Unfortunately solving this consistent cropping problem algorithmically was outside the bounds of this project.

Conclusion

Despite variations in character size, orientation, and position, the neural network was still able to recognize many of the characters. While 65% accuracy is still far below the 97% users demand, 2D image recognition is only part of the solution neural networks can bring to handwriting recognition. Combined with stroke analysis and temporal information, neural networks look to be a very promising solution. The results of this project are hardly new. In 1996, a 3-layer back-propagation network similar to the one used here was able to recognize characters out of a database of 500 Chinese symbols with a 90% accuracy. (Leung).

While neural networks are a promising solution there are some short term problems. Conducting experiments on this project, it became clear that correctly training a neural network can be a very time and processor intensive activity. This has resulted in some researchers advocating ‘lazy recognition’, not attempting to do character recognition in real time. (Nakagawa). Microsoft has also adopted a strategy with its upcoming Tablet PC to do character recognition silently in the background instead of in real time. But given the power of Moore’s Law, this is only a small hurdle in what may soon become an era of pen-based computing.



An early prototype of Microsoft’s new Tablet PC

REFERENCES

Isokoski, Poika. (2001). Model for Unistroke Writing Time. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. Seattle: ACM Press.

Lalomia, Mary. (1994). User Acceptance of Handwritten Recognition Accuracy. Proceedings of the CHI '94 Conference Companion on Human Factors in Computing Systems. Boston: ACM Press.

Leung, Wing-nin. Cheng, Kam-Shun. (1996). A Stroke-Order Free Chinese Handwriting Input System Based on Relative Stroke Positions and Back-propagation Networks. Proceedings of the 1996 ACM Symposium on Applied Computing. Philadelphia: ACM Press.

Nakagawa, Masaki. Machii, Kimiyoshi. Kata, Naoki. Souya, Toshio. (1993). Lazy Recognition as a Principle of Pen Interfaces. INTERACT '93 and CHI '93 Conference Companion on Human Factors in Computing Systems. Amsterdam: ACM Press.

Pittman, James. (1991). Recognizing Handwritten Text. Human Factors in Computing Systems Conference Proceedings on Reaching Through Technology. New Orleans: ACM Press.

APPENDIX I – CHARACTER INPUT

Computer

1 Provide some information about yourself

Gender	Pentium
Age	0.5 years

2. Provide a sample of each letter; these will be used to train the handwriting recognition system:

A A	B B	C C	D D	E E
F F	G G	H H	I I	J J
K K	L L	M M	N N	O O
P P	Q Q	R R	S S	T T
U U	V V	W W	X X	Y Y
Z Z				

3. Write a sentence using letters with some amount of space between them. This sentence will be used to test the handwriting recognition system.

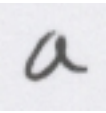
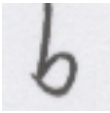
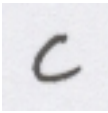
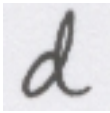
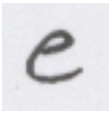
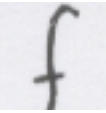
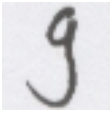
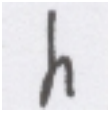
THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS

Male 1

1. Provide some information about yourself

Gender	Male
Age	21

2. Provide a sample of each letter; these will be used to train the handwriting recognition system:

A 	B 	C 	D 	E 
F 	G 	H 	I	J
K	L	M	N	O
P	Q	R	S	T
U	V	W	X	Y
Z				

3. Write a sentence using letters with some amount of space between them. This sentence will be used to test the handwriting recognition system.

the quick brown fox jumped over the lazy dogs				
la:	brick	doed	fo	log: the

Male 2

1. Provide some information about yourself

Gender	MALE
Age	21

2. Provide a sample of each letter; these will be used to train the handwriting recognition system:

A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	Q	R	S	T
U	V	W	X	Y
Z				

3. Write a sentence using letters with some amount of space between them. This sentence will be used to test the handwriting recognition system.

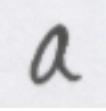
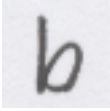
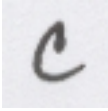
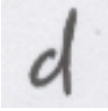
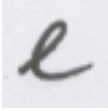
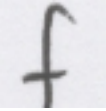
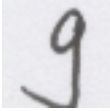
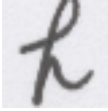
THE QUICK BROWN FOX JUMPED OVER THE LAZY DOGS									
LAZ	BR	CK	ED	HE	FO	GS	THE		

Female 1

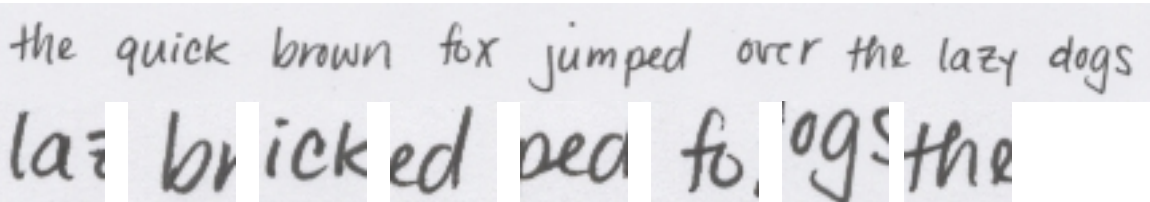
1. Provide some information about yourself

Gender	female
Age	20

2. Provide a sample of each letter; these will be used to train the handwriting recognition system:

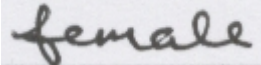
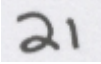
A 	B 	C 	D 	E 
F 	G 	H 	I	J
K	L	M	N	O
P	Q	R	S	T
U	V	W	X	Y
Z				

3. Write a sentence using letters with some amount of space between them. This sentence will be used to test the handwriting recognition system.

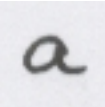

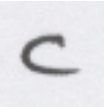
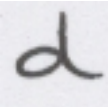
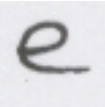


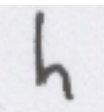

--

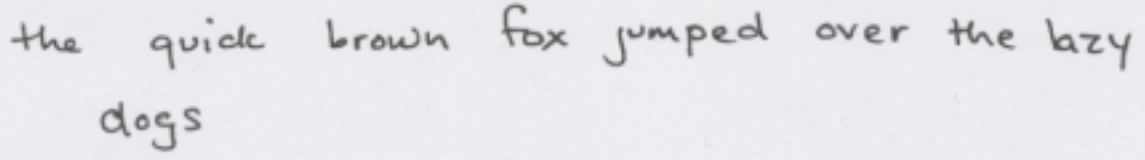
Female 2

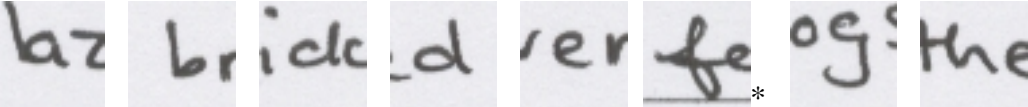
1. Provide some information about yourself

Gender	
Age	

2. Provide a sample of each letter; these will be used to train the handwriting recognition system:

A 	B 	C 	D 	E 
F 	G 	H 	I	J
K	L	M	N	O
P	Q	R	S	T
U	V	W	X	Y
Z				





*this was taken from "female" because the f in her sentence is entirely different

APPENDIX II - COMPLETE EXPERIMENT RESULTS

Experiment (1) The Effect of Set Size on Character Recognition

[All trials used printed text in an Arial font]

Trial	Epochs	MSE	Accuracy	Output
First 4 letters	314	0.0042	100	a - a b - b c - c d - d
First 8 letters	1714	0.0094	100	a - a b - b c - c d - d e - e f - f g - g h - h
First 12 letters	10000 (max)	.0279	75	a - a b - g c - c d - d e - a f - f g - g h - h i - i j - j k - c l - l
First 16 letters	5465	.03	75	a - g b - b c - g d - d e - e f - p g - g h - h i - i j - j k - h l - l m - m n - n o - o p - p
First 20 letters	2526	.029	65	a - a b - w c - c

				d - d e - e f - f g - g h - h i - q j - c k - k l - l m - d n - n o - q p - p q - q r - q s - s t - s
All 26 letters	10000 (max)	.038	62.5	a - v b - k c - c d - d e - f f - f g - g h - k i - i j - z k - k l - l m - m n - n o - q p - p q - q r - p s - p t - x u - g v - v w - w x - x y - y z - z

Experiment (2) The Effect of Handwriting Style on Character Recognition

[All trials are only the first 8 characters of each subjects handwriting]

Trial	Epochs	MSE	Accuracy	Output
Female 1	1225	.0091	62.5	a - a b - b c - c d - d e - a f - f g - h h - a
Female 2	417	.0099	50	a - a b - b c - e d - d e - e f - b g - b h - b
Male 1	10000 (max)	0.0139	37.5	a - c b - f c - b d - d e - e f - f g - b h - e
Male 2	3687	.0097	50	a - h b - b c - b d - d e - f f - f g - h h - h