Approved for Public Release; Distribution Unlimited
Case # 10-0493

© The MITRE Corporation. All rights reserved.

Online Medical Symbol Recognition Using a Tablet PC

Amlan Kundu, Qian Hu, Stanley Boykin, Cheryl Clark, Randy Fish, Stephen Jones and Stephen Moore

MITRE Corporation
202 Burlington Road
Bedford, MA 01730-1420

Abstract

In this paper we describe a scheme to enhance the usability of Tablet PC handwriting recognition systems by creating a software module for recognizing symbols that are not a part of the Tablet PC symbol library. The goal of this work is to make handwriting recognition more useful for medical professionals accustomed to using medical symbols in medical records. To demonstrate that this new symbol recognition module is robust and expandable, we report results on both a medical symbol set and an expanded symbol test set which includes selected mathematical symbols.

I. Introduction

A Tablet PC allows users to input data with a digital pen as well as via standard keyboard, mouse, or speech. Tablet PCs have special screens that use an active digitizer to enable users to write directly on the screen to control their PCs and to input information such as handwriting or a drawing. This process, called “inking,” enables users to add "digital ink" which appears as natural-looking handwriting on the screen. The digitized handwriting can be converted to standard text through handwriting recognition, or it can remain as handwritten text.

When the input interface is easy and intuitive, handwriting can be a more natural and convenient input method for data entry into applications. When a doctor is standing beside an exam table and moving from room to room, manipulating a keyboard can be quite cumbersome. Thus, inking makes EMR (electronic medical record) creation more natural and convenient for medical professionals than typing on a keyboard. Inking also allows users to insert a sketch or drawing, take free-hand notes, and annotate an existing document such as an X-ray. Consequently, using a pen to input information into the Tablet PC can be a more natural and productive way to work in many situations, and is often quicker and easier than using a keyboard.

On Tablet PCs, a digitizer overlaid on the LCD screen creates an electromagnetic field. When the pen comes in contact with the screen’s electromagnetic field, its motion is reflected on the screen as a series of data points. As the pen continues to move across the screen, the digitizer

© The MITRE Corporation. All rights reserved.

collects information from the pen movement in a process called "sampling." The Tablet PC digitizer is capable of sampling 130 "pen events" (units of motion that correspond to data points) per second [11]. These electromagnetic pen events are then represented visually on the screen as pen strokes. Because of its high sampling rate, the Tablet PC is able to create the effect of "real-time inking;" that is, as the user writes on the LCD screen, digital ink appears to flow from the moving pen. The high sampling rate also enables written ink to be displayed and stored with very high graphical resolution. Not only is this important for visual legibility on the screen, it is necessary for maximizing accuracy during the process of handwriting recognition.

Despite its versatility, the Tablet PC recognition engine does not recognize some common medical symbols, which limits its usability for medical records applications. While symbols composed of standard keyboard characters such as *mg% (milligrams per 100ml)* can be added to the default dictionary, others such as ♂ (*male*) remain unavailable since the addition of non-standard keyboard characters is not supported by the dictionary expansion utility. This paper describes our research effort to recognize a set of 15 of these handwritten medical symbols (Table 1) allowing their use by medical professionals using a Tablet PC. To demonstrate that the symbol recognition module is robust and expandable, we have also included 20 mathematical symbols for an expanded set with which to test our symbol recognition algorithm.

The primary contribution of the work described in this paper is that it deals with online medical symbol recognition in an application which includes both symbols and non-symbols. This combination of “medical”, “on-line” and “symbol/non-symbol” has not been addressed by previous work in the field of handwritten symbol recognition (HSR). While most of the papers related to HSR deal with offline HSR, online HSR, online mathematical symbol recognition in particular, is drawing increased attention [20]. In [21], online mathematical symbol recognition is addressed using a multi-classifier approach. In [22], online chemical symbol recognition is addressed using hidden Markov model based classifiers. In [23], online music symbol recognition is addressed using a complex scheme that recognizes individual strokes of the handwritten music symbol and uses that information for the final recognition. As has been noted, in this study we address online medical symbol recognition. However, our recognition approach does not deal with the handwritten medical symbol in isolation. In our work, handwritten medical symbols need to first be identified and separated from non-symbols as the writer writes. Only after a symbol is identified as a symbol is the final classification performed. Thus, the problem is one of identifying and classifying symbols embedded in a stream of non-symbols. This significantly increases the complexity of the problem.

We begin in Section II with a description of our approach to interface with a standard Tablet PC handwriting application and our handwriting symbol classification system. In Section III we describe the development of our test corpus and present experimental results. In Section IV we provide a description of a research prototype which integrates the medical/mathematical symbol recognition module described here with a Tablet PC as well as our plans for tighter integration with the Tablet PC’s handwritten recognition engine. Finally, Section V presents our conclusions and future plans.

II. Handwritten Symbol Recognition System

© The MITRE Corporation. All rights reserved.

Our handwritten symbol recognition system consists of three stages, 1 - Data Collection and Feature Extraction, 2 - Symbol Identification and 3 - Symbol Classification. This system augments the existing Tablet PC handwritten recognition system without requiring any changes to the existing system. The description of our Handwritten Symbol Recognition system in this section includes our approach to accessing information from the existing Tablet PC.

A. Data collection and Feature Extraction

The first step in symbol classification is the conversion of a handwritten bit mapped symbol image into the features used for classification. The classifier described in this paper uses 35 features extracted from a binary bit mapped image. The basis of our feature set are 30 features which have been successfully used in many handwriting applications [4,5]. These include three moment features that capture global shape information and 27 geometric and topological features including the number of loops, an X-joint feature, horizontal and vertical zero crossing features, a T-joint feature, the number of end points in upper, middle and lower zones of the character image, the number of segments in upper, middle and lower zones, and pixel distribution features.

Based upon our previous research with Arabic character recognition, we added five features to this standard feature set. Arabic characters share characteristics with symbols not normally found in the Roman character set such as abrupt changes in stroke direction and vertical stacking. These additional features were found to improve performance during Arabic character recognition and were included in our symbol classifier [5]. Two of these new features pertain to the character aspect ratio. The maximum vertical extent and maximum horizontal extent of the character are used to compute the normalized horizontal to vertical aspect ratio f_{hv} and the vertical to horizontal aspect ratio f_{vh}. Three chain code features, which assist with abrupt changes in stroke direction, were also included. For the given character image, 8-directional chain codes are computed. All chains with a length greater than a threshold are considered good chains. The feature f_{ch} is assigned a value from 0 (no good chain) to 1 (>3 good chains) based upon the number of good chains. The feature f_{rough} is a chain code based roughness measure and f_{con} captures chain code sharp turn information.

While it appears that some features of this empirically selected feature set are correlated, it has been shown [5] that feature reduction techniques such as Principal Component Analysis (PCA) result in a degradation in performance. Our experiments and analysis showed that we could reduce our feature set from 35 to 28 but at a cost of a 5% reduction in performance. Since feature computation is very fast, we elected to stay with our set of 35 features.

Extraction of these features requires a binary bit mapped image of the user's handwriting. Therefore our first step was the collection of bit mapped images of handwritten samples from our set of target medical symbols. While our main sample collection media was a tablet PC, we also made use of data collected using a digital pen from IOGear. The main advantage of the IOGear device was the relative simplicity (compared to the Tablet PC) of capturing a large number of images for training and testing. IOGear images, when properly captured, are quite similar to those captured with the Tablet PC as they both simulate online handwriting. We begin with a description of the feature extraction process using the Microsoft Handwriting SDK on a tablet PC.

© The MITRE Corporation. All rights reserved.

Capturing an Image with a Tablet PC

The Microsoft handwriting SDK converts Tablet PC writing not into a binary bit mapped image, but into a sequence of strokes defined by Beziér [8] points. A single stroke is the movement of the pen between when it touches the screen and when it is lifted. Beziér points are a compressed representation of these strokes. Given these Beziér points, Beziér functions can be used to generate a bit mapped representation of the stroke as described below.

An N^{th} order Beziér function fits a curve to N+1 Beziér points in such a way that the 2D coordinates of any point along the curve can be extracted. To find a particular point on the curve described by a 3^{rd} order Beziér function we use the equation:

$$p = C_1B_1(t) + C_2B_2(t) + C_3B_3(t) + C_4B_4(t)$$

where:

p is the point in 2D space
C_i are the Beziér points
B_i are the Beziér functions described below
t is the fraction of the distance along the curve (between 0 and 1)

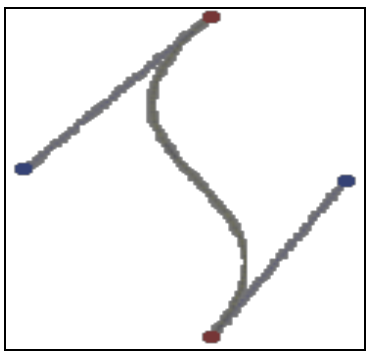
The Beziér functions calculate the weight to be given to each of the four Beziér points in determining the position of the selected point p which is a function of its distance along the curve connecting C_1 and C_4. We used the Bernstein Basis Function, $1 = t + (1 - t)$ as a parametric method for generating these curves. For a cubic (3rd order) curve, we cube both sides:

$$1^3 = (t + (1 - t))^3 \quad \Rightarrow \quad 1 = t^3 + 3t^2(1 - t) + 3t(1 - t)^2 + (1 - t)^3$$

While it is possible to simplify this equation further, for the purpose of this analysis we wanted to express it in terms of *t* and *(1 - t)*. The cubic equation gives us the four Beziér functions:

$B_1(t) = t^3$
$B_2(t) = 3t^2(1 - t)$
$B_3(t) = 3t(1 - t)^2$
$B_4(t) = (1 - t)^3$

Given a sequence of Beziér points, the first four define the first Beziér curve of the stroke to be recreated. The last Beziér point of this group and the next three Beziér points define the next (Figure 1). Since these two curves share a common point, they are connected. The Beziér point computation continues until all the points in the set are used up.



© The MITRE Corporation. All rights reserved.

Figure 1: Creation of a Beziér curve based on four control points given by dots

In our implementation, we used at least 100 points (p) in each Beziér curve to recreate a smooth image followed by filtering and data reduction to create offline images adequate for training and testing. Figure 2 below shows a reconstructed image from 110 Beziér points.

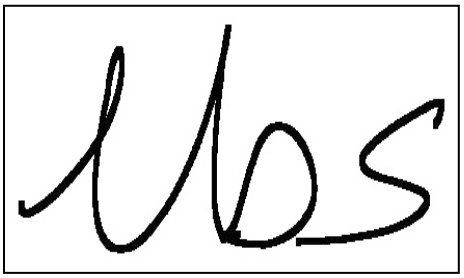


Figure 2: A typical reconstructed image from Beziér points.

The inking in Figure 2 consists of two strokes. The first stroke has 76 Beziér points, and the second has 34. Using the algorithm just described we convert the first 76 Beziér points into a stroke consisting of 25 small connected Beziér curves; each curve consisting of at least 100 points. The resulting offline image is shown in Figure 2.

To capture Beziér points for training and test from a Tablet PC, the application shown in Figure 3 was developed. The interface contains a handwriting area on the left-hand side, from which the input is digitized by the Tablet PC and separated into a series of ink strokes represented by Beziér points. The Tablet PC SDK is used to retrieve the Beziér points and surface them to the application, via a text box, as a series of coordinates, where they can be collected and analyzed. Using this data collection interface, each sample of a handwritten symbol is saved in its own file.

© The MITRE Corporation. All rights reserved.

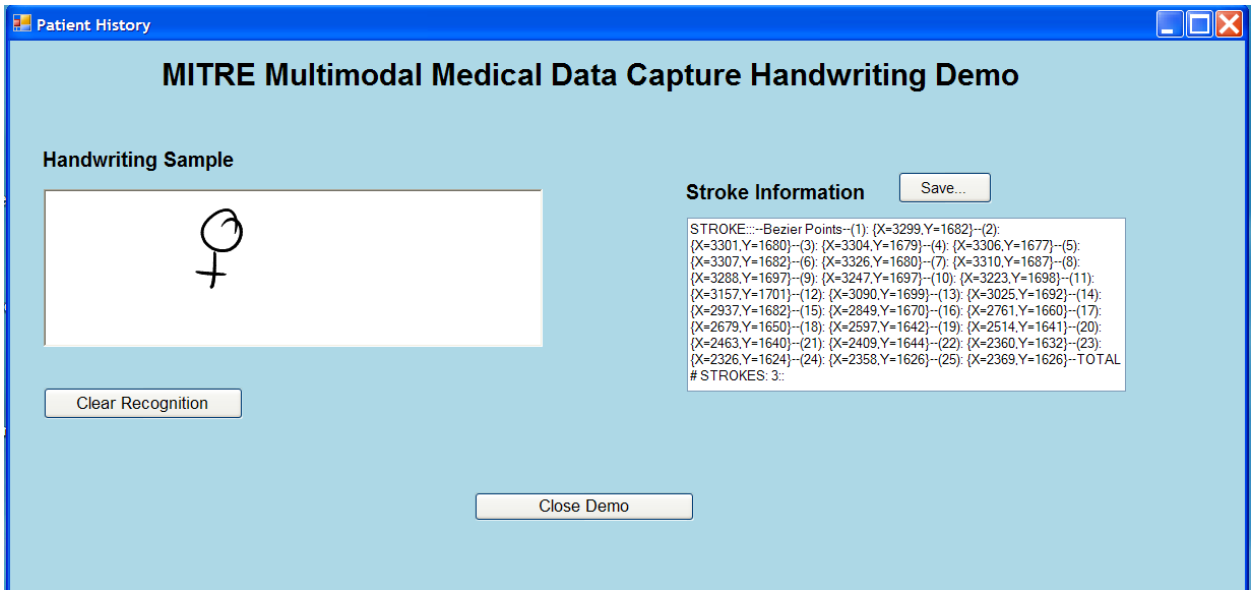


Figure 3: MITRE Handwriting Capture Demo Application

Capturing an Image with the IOGear Digital Pen

The IOGear Mobile Digital Scribe is a device for capturing handwriting strokes while writing on normal paper. The device consists of a receiver which clips onto the paper and a ballpoint pen whose motion is tracked by the receiver. Data stored in the receiver is downloaded via a USB connection and the device software produces a full page image. No special paper is required to use the IOGear digital pen.

We used IOGear to capture handwriting samples and export several pages of high-quality JPG images, which we then used for training and testing. Unlike the Tablet PC interface, a single saved page can contain multiple examples of the handwritten symbols. A segmentation algorithm was used to segment the pages into individual symbol samples for use in the training/testing of the system. Some contrast and brightness adjustments were made to convert the IOGear images from gray scale to black and white (binary) images matching the reconstructed Tablet PC image. Although IOGear images were not created using a Tablet PC stylus, there was significant similarity in the final image output as both simulate online cursive writing.

Some typical sample images of medical symbols are shown in Figure 8 of Appendix-I.

B. Symbol Identification

Typical Tablet PC handwriting systems support multiple languages. It has been shown that performance would be dramatically degraded if all languages shared a single library and recognition engine. Therefore, each of the supported languages -- English, German, French, Korean, Japanese and Simplified and Traditional Chinese -- has its own recognition engine with

© The MITRE Corporation. All rights reserved.

a bank of language-specific handwriting samples. As we have already described, the ability to add non-character symbols to the existing system's English library is not available. Based upon this and the acknowledged benefit of multiple libraries, we decided to treat our new symbol recognition as if it were a new language. Therefore, as shown in Figure 4, the first stage in our new recognizer is a binary classifier that discriminates between a symbol and a non-symbol. Henceforth, we will refer to this step as symbol identification.

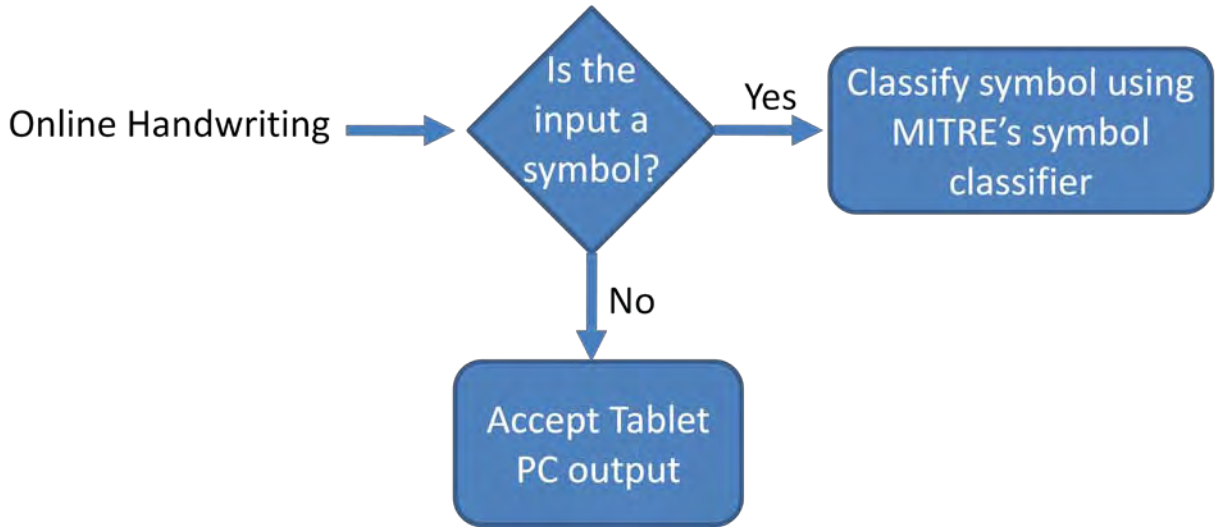


Figure 4: Two stage symbol classification

The binary symbol/non-symbol classification problem is deceptively complex. Many non-symbols have shape characteristics similar to those of symbols. To address this issue our feature set included features for classification that capture the following characteristics of symbols vs. non-symbols.

- Symbols are usually short relative to non-symbol handwriting.
- Vertical stacking of strokes appears in some symbols.
- Basic shapes of the symbols differ from the shapes of the non-symbols

Preliminary testing on the Symbol Identification task was performed using the following standard classifier types [7]:

- Bayes
- Neural Net (MLP)
- Logistic
- Linear Regression
- Radial Basis Function
- Support Vector Machine
- Several Meta-classifiers
- Nearest Neighbor 1 and 5
- Several Tree classifiers (including J48)

© The MITRE Corporation. All rights reserved.

- RandomForest classifiers (essentially a multi-classification based on many trees)
- Several rule based classifiers

We found that three classifiers – a Multilayer Perceptron Neural Net, a RandomForest classifier and a Nearest Neighbor classifier— consistently outperformed all other classifiers. Based upon these preliminary results, we applied a multi-classifier framework to the binary symbol/non-symbol problem using the following classifiers as components:

- Multilayer Perceptron Neural Net [2,3,7,9]
- RandomForest classifier (an advanced tree classifier) [1,7,10]
- Nearest Neighbor classifier [2,3,7]

As shown in section III, this multi-classifier approach was more robust than any individual classifier.

Of the three component classifiers, the multi-layer Perceptron Neural Net and Nearest Neighbor classifiers are well-known [2,3,9]. A RandomForest classifier is an advanced tree classifier. In a RandomForest classifier, subsets of the feature set are randomly selected and trees are constructed based on these subsets. While using multiple decision trees, the focus is on the splitting criterion and the tree sizes to achieve a balance between over-fitting and achieving maximum accuracy. It has been shown [1] that RandomForest classifiers outperform other tree classifiers.

We implemented all three classifiers using the Weka software package [7] choosing the IB1 (Instance-Based Classifier using 1 nearest neighbor(s) for classification) option for the Nearest Neighbor classifier. Default parameters were used for all three classifiers. For the Multilayer Perceptron Neural Net, the number of iterations for training was set to 400. For the RandomForest classifier, the number of trees was set to 10. More details regarding these classifiers can be found in the references [1,2,3,7,9,10]. The individual outputs of the three classifiers were merged using a simple majority voting scheme.

C. Symbol Classification

Once it has been determined that the image to be classified is a symbol, the input then passes through another module that identifies the appropriate symbol from the symbol library. The same classifiers investigated for the Symbol Identification problem were evaluated for Symbol Classification. Interestingly, we found that the same classifiers which gave optimum performance for Symbol Identification were the leading candidates for Symbol Classification. We again made use of the multi-classifier approach described above. In the symbol classification stage, the outputs of the three individual classifiers were merged using a simple majority voting scheme. In cases where the output of each of the classifiers was different, the output of the classifier with the highest confidence was used.

III. Experiments

© The MITRE Corporation. All rights reserved.

This section describes the results of our experiments to both identify when the input handwriting was a symbol and to classify the new symbol set. These experiments begin with the creation of a lexicon for training and test.

A. Lexicon Creation

Since non-symbols include all words, numbers, etc. excluding the symbols, creating a representative corpus for all non-symbols would be a huge task which fortunately turned out to be unnecessary. We observed that symbols are limited to inputs which are one or two characters in length. Therefore our non-symbol lexicon could be limited to one and two character inputs. We developed a lexicon of 443 non-symbol words. This included letters of the alphabet, Arabic numerals 1 through 99, Roman numerals, state abbreviations (e.g., *MA*) and other two-letter geographical abbreviations, temporal abbreviations (e.g., *pm*), and high-frequency one and two letter words such as *it*, *of*, and *go*. In addition to these, medical practice uses many abbreviations which should be considered as non-symbols. We researched several web sites [13-19] to find medical and scientific abbreviations and acronyms, and selected those that are used in clinical reports for inclusion in our non-symbol lexicon. These include abbreviations for units of measure (e.g., *cm*), vitamins (e.g., *B1*), chemical elements and compounds (e.g., *Zn*, *O2*), anatomical terms (e.g., *GB* for *gallbladder*), disease names (e.g., *TB* for *tuberculosis*), procedures (e.g., *bx* for *biopsy*), bodily substances (e.g., *TG* for *triglycerides*), vital signs and lab assay descriptors (e.g., *RR* for *respiratory rate*), report section headings (e.g., *FH* for *FAMILY HISTORY*), the names of medical specialties (e.g., *OB* for *obstetrics)*, hospital locations (e.g., *OR* for *operating room*), the Latin expressions used in drug prescriptions (e.g., *qd* for *every day*), and frequently used expressions such as *RO* (*rule out)* and *WN* (*well nourished)*.

Handwritten sample images of these 443 non-symbol words were collected from different writers. These non-symbol images along with symbol images were used to develop the symbol vs. non-symbol binary classifier. Some typical non-symbol sample images are shown in Figure 9 of Appendix-I.

© The MITRE Corporation. All rights reserved.

For our symbol recognition lexicon we chose 15 common medical symbols (Table 1) whose addition to the standard Tablet PC dictionary is not supported.

Symbol	Meaning	Symbol	Meaning
(lb.)	Pound	≈	Approximately equal
Δ	Change; heat	☐	Male
µg	Microgram	♂	Male
p	After	○	Female
s	Without	♀	Female
µ	Micron	↑	Increase
±	Plus or minus; either positive or negative; indefinite	↓	Decrease
×	Multiplied by; magnification		

Table 1: List of medical symbols for the experiment

In addition, we added 20 common mathematical symbols (Table 2) to create an extended set of symbols. This extended set was created to test the robustness and extendibility of the symbol recognition module

α	β	λ	τ	η	σ	δ	ψ	γ	π
ρ	Π	Σ	Φ	Ω	∫	∓	≅	≤	≥

Table 2: List of mathematical symbols for the experiment

Using both the Tablet PC and IOGear digital pen as input devices, 4102 non-symbols, 2525 medical symbols and 1196 mathematical symbols were collected and allocated for training or test as shown in Table 3. The number of writers used for data collection is indicated in the table. None of the data used for training was used for test. However, some of the data used for test was created by writers of training data.

		Tablet PC	Writers	IOGear	Writers	Total
Medical Symbols	Training	1797	7	510	4	2307
Mathematical Symbols	Training	1080	6	0	0	1080
Non-symbols	Training	989	3	2257	6	3246
Medical Symbols	Test	38	2*	180	4**	218
Mathematical Symbols	Test	120	2	0	0	120
Non-symbols	Test	413	2*	443	4**	856

Table 3: Training and Test Corpus. (*1 new writer, 1 also contributed to training. **1 new writer, 3 also contributed to training)

© The MITRE Corporation. All rights reserved.

B. Symbol Identification

We evaluated the performance of our binary symbol vs. non-symbol classification stage first with a symbol set containing only medical symbols. The classifier was then retrained using the expanded symbol set which includes our selected mathematical symbols to explore the impact of expanding the symbol class.

As shown in Table 4, a total of 1074 samples (218 symbols, 856 non-symbols) were labeled by the symbol identification classifier. The accuracy on this task was 96.7%; a significant improvement over the chance performance of 79.7% achieved by always choosing the majority class of Non-symbol.

	Total	Correct	Incorrect	Accuracy
Medical Symbols	218	218	0	96.7%
Non-symbols	856	821	35	

Table 4: Symbol identification for medical symbols only

Expanding the symbol class to include mathematical symbols reduced the accuracy of the symbol identification task to 92.6% as shown in Table 5. While this is still a significant improvement over chance performance of 71.7%, achieved by always choosing the majority class of Non-symbol, increasing the number of symbols from 15 to 35 caused a 4.2% relative degradation in performance.

	Total	Correct	Incorrect	Accuracy
Medical + Mathematical Symbols	338	332	6	92.6%
Non-symbols	856	774	82	

Table 5: Symbol identification for both medical and mathematical symbols

C. Symbol Classification

The symbol classification stage of our system was evaluated using only the identified medical symbols and then again after retraining with the expanded symbol set.

As shown in Table 6, the accuracy of the symbol identification classifier was 95.4% on the medical symbol only test set and 95.9% on the expanded symbol set. Apparently more than doubling the size of the symbol set does not cause problems during classification even though it produced a minor degradation in the symbol identification stage.

	Total	Correct	Incorrect	Accuracy
Medical Symbols	218	208	10	95.4%
Medical + Mathematical Symbols	338	324	14	95.9%

Table 6: Symbol classification on both medical only and the expanded symbol set

D. Evaluation of Data Collection and Multi-classifier Decisions

© The MITRE Corporation. All rights reserved.

The system described in this paper was heavily influenced by our decisions to collect data from two different sources and to use a multi-classifier architecture rather than a single classifier. In this section we report the empirical performance supporting those decisions.

Training and test data was collected using both the target application Tablet PC and the IOGear digital pen. Table 7 presents performance on both the symbol identification and symbol classification tasks for the extended symbol set. The table indicates that when classification results are separated by the data source, there is very little change in performance. The IOGear digital pen provided 42% of the data used to train the symbol identification classifier but only 15% of the training data for symbol classification.

		Total	Correct	Incorrect	Accuracy
Extended Symbol Identification	Tablet PC	571	521	50	91.2%
	IOGear	623	585	38	93.4%
Extended Symbol Classification	Tablet PC	158	152	6	94.4%
	IOGear	180	172	8	95.6%

Table 7: Performance of the two classifiers on source dependent data

Finally, we evaluated the performance of individual classifiers and compared their accuracy with the accuracy of the combined classifier. Both Symbol Identification and Symbol Classification were performed on subsets drawn from the full expanded symbol test set. Symbol Identification used 856 samples and Symbol Classification used 240 samples. Table 8 indicates that using a multi-classifier approach was significantly better than any of the individual classifiers when applied to the Symbol Identification task. The improvement was less dramatic but still positive when applied to Symbol Classification.

	Symbol Identification	Symbol Classification
Random Forest	82.1%	94.2%
Nearest Neighbor	84.2%	95.0%
Neural Net	80.6%	82.3%
Combined Multi-classifier	90.4%	98.3%

Table 8: Performance of individual classifiers vs. multi-classifier approach on both the Symbol Identification and Symbol Classification tasks.

IV. Application

The handwriting symbol recognition system described in this paper is a part of a larger prototype system being developed at MITRE which provides an interactive multimodal interface for an open-source web application called OpenEMR [23]. OpenEMR is a web application that manages patient medical records. Our interface is designed to allow medical professionals to include speech and handwriting tools when interacting with OpenEMR via a web browser plug-in (Mozilla Firefox), for the purpose of searching, navigating, and inputting medical data in ways that are natural to the user. The purpose of this prototype is to demonstrate the potential for this type of tool to streamline the productivity and workflow of the medical user beyond the traditional keyboard-and-mouse paradigm offered by current electronic medical record systems.

© The MITRE Corporation. All rights reserved.

Software developers can easily adapt the propotype software to work with other electronic health records systems.

In this section we focus on the handwriting modality of the prototype, which connects a Firefox browser with the Tablet handwriting tool. This tool has been customized with medical dictionaries for out-of-vocabulary (OOV) terminology for the purpose of generating better recognition performance.

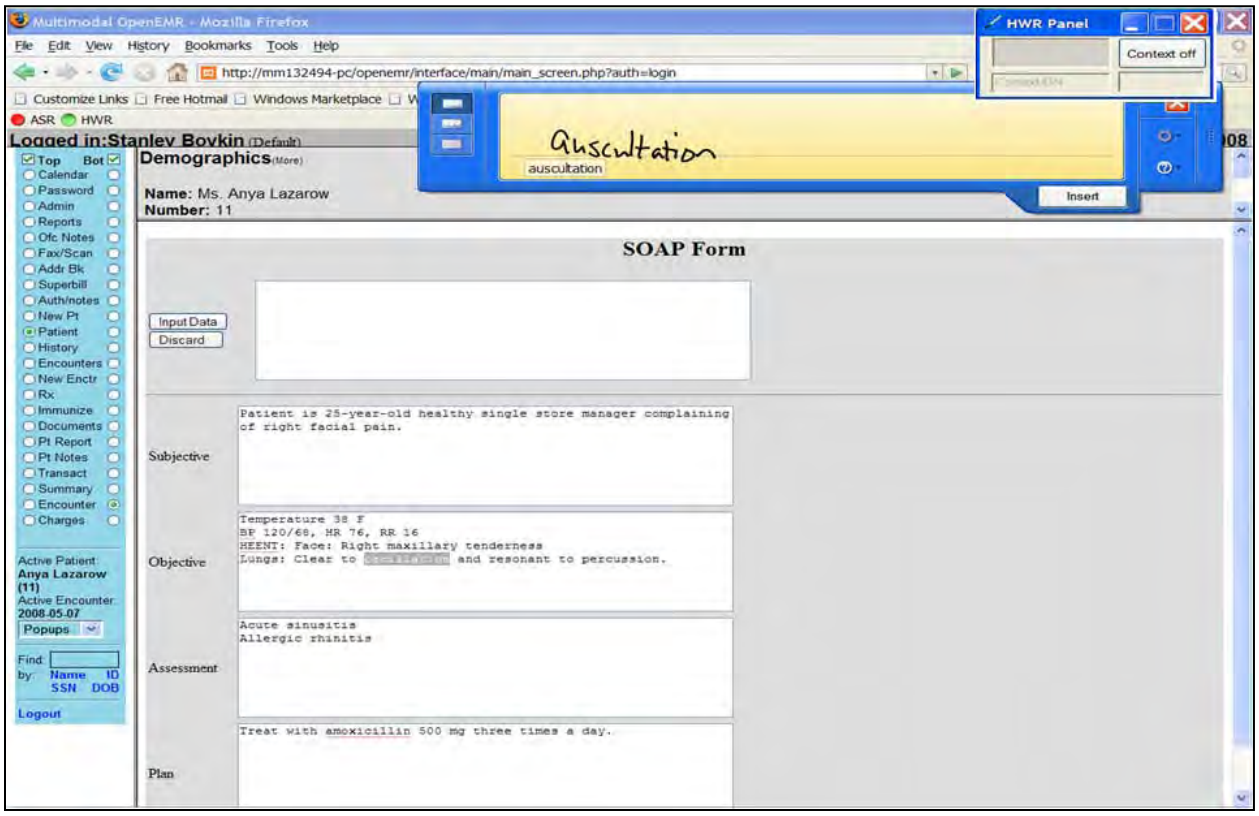


Figure 5: MITRE's handwriting recognition prototype interface instantiated within OpenEMR

Figure 5 shows a screenshot of the prototype interface. The web page being displayed represents the medical practitioner's summary of a patient visit, commonly known as a SOAP Note (Subjective, Objective, Assessment, and Plan). This page of the open-source system has been modified to utilize handwriting and speech, signaled by the green and red buttons in the upper left portion of the screen. A handwriting input panel is situated in the upper right corner. Clicking on this control activates the pop-up handwriting pad shown. In the scenario shown here, the handwriting tool is used to correct text that was input via speech recognition, changing the mis-recognized word 'oscillation' to the correct term 'auscultation'. The handwriting panel is available to a user to enter text or make corrections in text input areas. It is believed that this expanded modality would be useful in environments where keyboarding is not practical – for example, in triage situations where desktop terminals are not readily available.

© The MITRE Corporation. All rights reserved.

Currently, we are working to augment this prototype with the symbol recognition research system described in this paper, so that medical symbols can also be recognized and displayed. A symbol recognition application has been developed to demonstrate the ability to pass ink from the Tablet PC to the symbol recognition algorithm described in this paper (Figure 6).

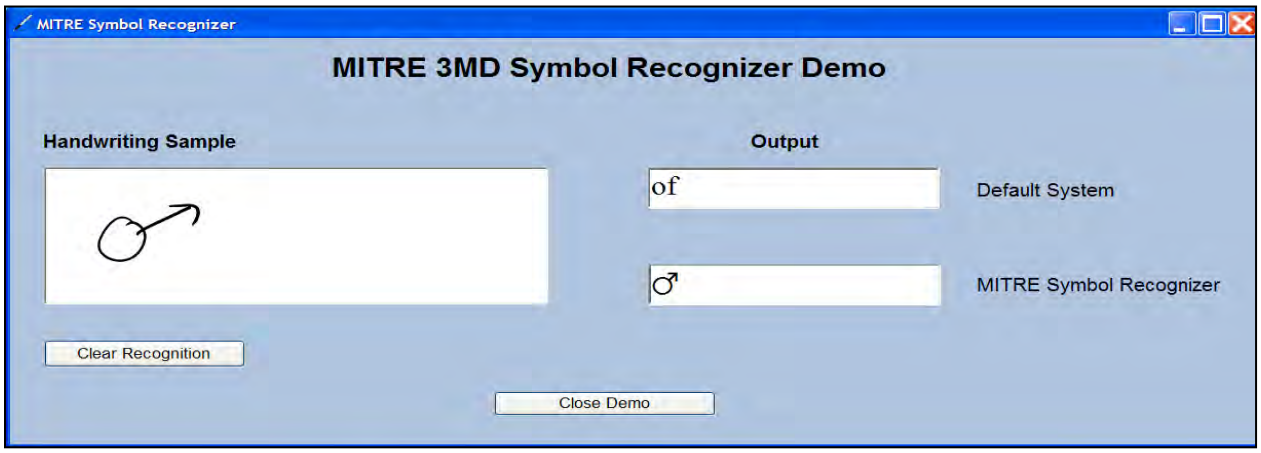


Figure 6: Recognizing symbol input with and without symbol recognizer

This interface contains a handwriting area on the left-hand side, from which the input is digitized and separated into a series of ink strokes. The stroke data is then passed to the symbol recognition algorithm, with the results displayed on the right side. (The Tablet system's result is displayed above to compare the research with default behavior).

We are working to combine the behavior of the two recognition engines -- the native handwritten recognition engine and the symbol recognition engine described in this paper -- so that mixed handwriting, containing both text and symbol input, can be parsed and recognized (Figure 7).

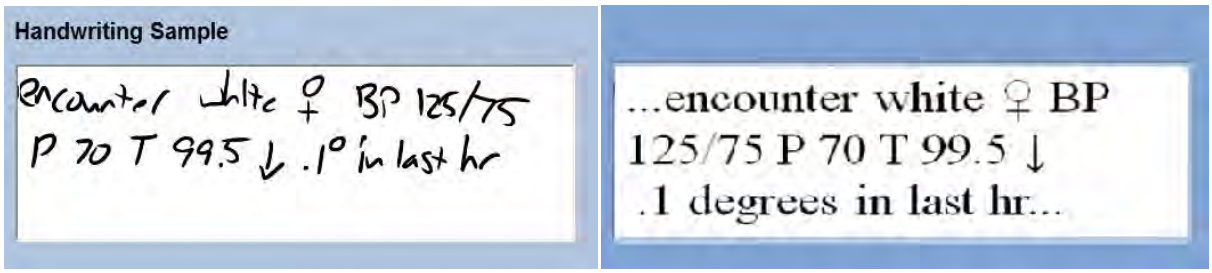


Figure 7: Mixed Text and Symbol Handwriting Recognition (ink and electronic text representations)

V. *Conclusions*

In this paper we have described our method of training a system to automatically recognize handwritten medical symbols that are out of the existing system's lexicon. While our intent is to recognize symbols relevant to a medical application, we have also shown that the symbol

© The MITRE Corporation. All rights reserved.

classification is robust to symbols from other domains. We have shown that using a multi-classifier architecture provides improved performance over an individual classifier.

This study has shown a potential method to augment existing recognition system for out-of-vocabulary symbol recognition.

Future work will address improving the performance of the binary symbol identification stage and final integration of the symbol recognizer with the existing Tablet PC system. Since symbols constitute a small fraction of the handwriting input, misidentification of non-symbol characters by the binary symbol identification stage must be kept very low to reduce inserted errors to the level necessary for adoption of the symbol recognition enhancement. The fact that medical abbreviations look similar to symbols makes this a difficult task. Our future work will involve more data collection and algorithm refinement based on contextual information to improve the performance of binary symbol vs. non-symbol classification.

References

[1] T. C. Ho, "*Random Decision Forest*," Proc. 3rd Intl. conf. on document analysis and recognition, Montreal, CA, pp. 278-282, 1995.

[2] C. M. Bishop, *Pattern recognition and Machine Learning*, ISBN – 10: 0-387-31073-8, Springer, 2006.

[3] J. M. Parker, Algorithm for Image Processing and Computer Vision, ISBN-10: 0-471-14056-2, John Wiley and Son, 1997.

[4] M.-Y. Chen, A. Kundu, and S. N. Srihari, " *Variable duration hidden Markov model and morphological segmentation for handwritten word recognition*." IEEE Trans. on Image Processing , 4(12), 1995.

[5] A. Kundu, J. Phillips, T. Hines, B. Huyck and L. Van Guilder, *"Arabic Handwriting Recognition Using Variable Duration HMM,"* Proceeding of International Conference of Document Analysis and Recognition (ICDAR), pp. 644 – 648, Brazil, Sept. 2007.

[6] J. Shlens, *A tutorial on principal component analysis.* http://www.cs.cmu.edu/ elaw/papers/pca.pdf, pages 1–13, 2007.

[7] Weka Machine Learning Project, The University of Waikato, New Zealand, http://www.scms.waikato.ac.nz/ml/.

[8] P. J. Schneider, **"**An algorithm for automatically fitting digitized curves," ***Graphics Gems, A.*** S. ***Glassner, ed.,*** pp. 612-626. Academic Press, San Diego, CA, 1990.

[9] Lippmann, R. P., "An Introduction to Computing With Neural Net," ASSP Magazine, 2, 4-21, 1987.

© The MITRE Corporation. All rights reserved.

[10] Leo Breiman, “Random Forests”, Machine Learning, pp. 5-32, Vol. 45, No. 1, 2001

[11] “EMR Digital Ink Handwriting Recognition”, http://www.medscribbler.com/handwriting_electronic_medical_records.html, Jan. 12, 2009.

[12] R. Milewski and V. Govindaraju, “Handwriting Analysis of Pre-Hospital Care Reports”, Proc. 17th IEEE Symposium on Computer-Based Medical Systems (CBMS’04), 2004.

[13] “Flash-Med: Medical Acronyms and Abbreviations for Medical Terms”, http://www.flash-med.com/Abbreviation.asp

[14] “Medical Abbreviations”,http://pharmsc.buffalo.edu/courses/phc311/latin.html

[15] “Common Medical Abbreviations”, http://www.globalrph.com/abbrev.htm

[16] “United States Postal Service -- Abbreviations”, http://www.usps.com/ncsc/lookups/usps_abbreviations.html

[17] “List of medical abbreviations – Wikipedia, the free encyclopedia”, http://en.wikipedia.org/wiki/List_of_medical_abbreviations

[18] “Acronyms and Abbreviations Glossary”, http://www.labcorp.com/datasets/labcorp/html/acronyms_abbreviations/newid1.htm

[19] “JD-MD Medical Abbreviations Glossary”, http://www.jdmd.com/glossary/medabbr.pdf

[20] K-F. Chanand D-Y. Yeung, “Mathematical Expression Recognition: A Survey”, International Journal of Document Analysis and Recognition (IJDAR), Vol. 3, No. 1, pp. 3-15, 2000.

[21] U. Garain and B. B. Chaudhuri, “On Machine Understanding of Online Handwritten Mathematical Expressions”, Proceedings of 7th International Conference on Document Analysis and Recognition (ICDAR’ 03), 2003, pp. 349-353.

[22] Y. Zhang, G. Shi and J. yang, “HMM-based Online Recognition of Handwritten Chemical Symbols”, Proceedings of 10th International Conference on Document Analysis and Recognition (ICDAR’ 09), 2009, pp. 1255-1259.

[23] H. Miyao and M. Maruyama, “An Online Handwritten Music Score Recognition System”, Proceedings of the 17th International Conference on Pattern Recognition (ICPR’04), 2004, pp.461-464.

[23] OpenEMR: http://www.oemr.org/

© The MITRE Corporation. All rights reserved.

APPENDIX-I: Sample Images

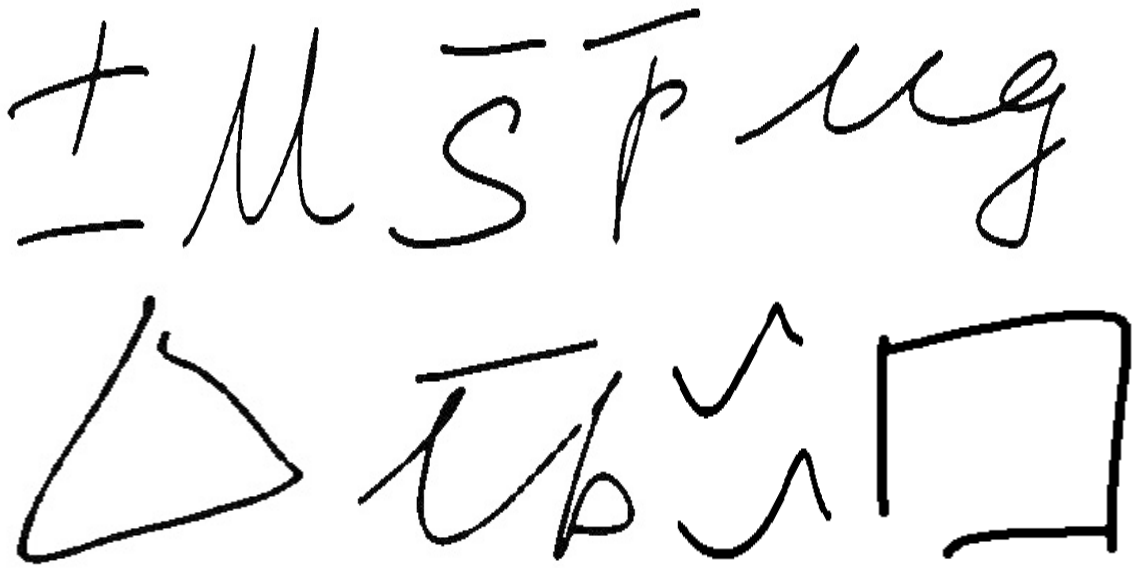


Figure 8: Typical image of symbols 7, 6, 5,4,3, 2, 1, 9 and 10

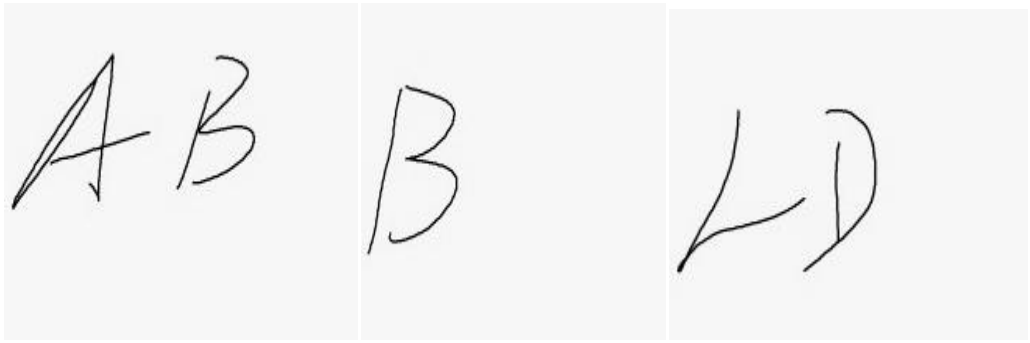


Figure 9: Typical non-symbol images from a lexicon of 443 entries