Robust Dynamic Symbol Recognition: the ClockSketch Classifier

by Kaichen Ma

[B.S., C.S. M.I.T., 2013]

Submitted to the Department of Electrical Engineering and Computer Science

in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Electrical Engineering and Computer Science

at the Massachusetts Institute of Technology

May 2013 [June 2014]

Author: Signature redacted

Department of Electrical Engineering and Computer Science

May 22, 2014

Certified by: Signature redacted

Prof. Randall Davis, Thesis Supervisor

May 22, 2014

Accepted by: Signature redacted

Prof. Albert R. Meyer, Chairman, Masters of Engineering Thesis Committee

Robust Dynamic Symbol Recognition: the ClockSketch Classifier

by Kaichen Ma
Submitted to the Department of Electrical Engineering and Computer Science

May 22, 2014
In Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Electrical Engineering and Computer Science

ABSTRACT

I present an automatic classifier for the digitized clock drawing test, a neurological diagnostic exam used to assess patients' mental acuity by having them draw an analog clock face using a digitizing pen. This classifier assists human examiners in clock drawing interpretation by labeling several basic components of a drawing, including its outline, numerals, hands, and noise, thereby freeing examiners to concentrate on more complex labeling problems. This is a challenging problem despite its specificity, because the average user of the clock drawing test has a high likelihood of cognitive or motor impairment. As a result, mistakes such as crossed-out numerals, messiness, missing components, and noise will be common in drawings, and a well-designed classifier must be capable of handling and correcting for various types of error.

I describe in this thesis the construction of a system that is both accurate and robust enough to handle variable input, laying out its components and the principles behind its design. I demonstrate that this system accurately recognizes and classifies the basic components of a drawing, even when applied to a wide range of clinical input, and that it is able to do so because it relies both on statistical analysis and on common-sense observations about the structure of the problem at hand.

Table of Contents

1. Introduction to ClockSketch

1.1 The Clock-Drawing Test

Originally developed in the 1960s to screen for dementia, the clock-drawing test is now widely used by neurologists to help diagnose a wide range of mental disorders. The test is basic but powerful; a patient is given a piece of paper and asked to draw the face of an analog clock, complete with all 12 digits and with the hands indicating a specific time, often "10 after 11". Judging by the clarity and cleanliness of the resulting drawing, whether components are missing, and other details, clinicians can estimate the patient's level of cognitive impairment and even pinpoint the specific mental faculty that might be impaired, such as memory or language processing. Despite the natural subjectivity of the interpretation process, the clock-drawing test's results have nevertheless been shown to have a high degree of correlation with actual cognitive disorders [1].

Traditionally, the clock drawing test has been carried out with pen and paper, but digitizing pen technology allows the test to be converted into digital format, enabling other methods of analysis, including large-scale data analysis. Digitizing the test captures the data as a sequence of sampled time-stamped points, ordered into strokes based on the rise and fall of the pen itself.

To make it easier for clinicians to interact with the now-digital clock drawing data, the ClockSketch program was created to display clock drawings, allow their components to be labeled and organized, and perform basic analysis on drawings to assist in diagnosis. But while display and analysis are both important for clinical use, it is labeling, and specifically automating the process of assigning labels to each part of a clock drawing, that poses the most interesting challenge from both a computer science and a usability perspective.

1.2 The Labeling Problem

At its most basic, a clock drawing consists of a series of strokes drawn by a patient. Each of these strokes represents some component of the clock drawing, such as a numeral, a hand, or the outline. Making an accurate diagnosis from a clock drawing hinges on the examiner being able to correctly recognize each component. Explicit labeling of each component is unnecessary in the paper case, but when a clock drawing is digitized, it becomes much more important that the strokes corresponding to each clock component are labeled correctly, establishing an interpretation of the drawing that can be used in large-scale or other analysis.

Instead of requiring a human examiner to explicitly label each stroke as they might a paper drawing, we would like an automated process to carry out the bulk of stroke labeling, and only defer to a human evaluator on the most difficult-to-label cases. Automatic labeling will dramatically speed up the processing of most clock drawings, and leave human examiners more time and energy to spend on the ones that require attention the most.

1.3 The Robust Automatic Classifier

What should such an automatic classification system look like? In order to be useful for a large number of clock drawings, an automatic clock-drawing classifier must be able to perform *at least* the following with high accuracy:

- Group together strokes that belong to the same component, such as the numeral "12", and separate strokes that belong to different components, such as the minute and hour hands.
- Identify strokes that make up the clock outline, numerals, and hands, which are the most common components present in a clock drawing.
- Accurately label each of these types of components, assigning the correct numerical label from 1-12 to numeral components, and labeling the hour and minute hands separately.

If an automatic classifier fulfills the above requirements, it will be able to handle the majority of well-drawn clock drawings, consisting of a clock outline, numerals drawn around the outline, and hands (and occasionally a center dot) drawn in the center. However, the performance of the classifier on imperfect drawings must also be considered, as the clock-drawing test is primarily used in a clinical context, with patients who have a higher probability of cognitive problems than ordinary users. Many clock drawings contain small deviations from the expected baseline in the form of increased messiness, crossed-out or repeated numerals, unintentional strokes, or other components that are neither hands or numerals. A smaller percentage of drawings deviate even more strongly, missing components entirely or being so messy that interpretation would be difficult even for a human observer.

As outlined above, an automatic classifier can opt to pass these more difficult recognition problems onto a human observer. However, a *robust* automatic classifier will still be able to first make a strong effort at recognizing the basic components of hands, numerals, and the clock outline, regardless of the messiness of the drawing or presence of extra components.

The automatic classifier I have designed, and describe in more detail in section 3, follows this principle of robustness. Given any unlabeled clock drawing, it will divide up the drawing's strokes into meaningful components based on each stroke's spatial location and time drawn, and then assign the best possible label to each component thus created. It integrates state-of-the-art symbol recognition with human reasoning about the structure of the problem, creating a system that is accurate in labeling, robust to various forms of error, and easily extensible to more diverse components.

2. Definitions

2.1 Terminology

The list below clarifies some of the most basic terms used in labeling. This is not an exhaustive list; less commonly used terms will be defined as they appear in the text.

2.1.1 Clock Drawing Components

- **Clock drawing:** a drawing made by a patient during a single phase of the clock-drawing test. It usually contains one recognizable clock, but sometimes more or less. The goal of this system is to accurately recognize and label all components within a clock drawing.
- **Clock face:** a drawing of a single analog clock, recognized as such by hands within a ring of numerals, often with a circular outline. Very rarely, a clock drawing may not contain any recognizable clock faces.
- **Clock outline:** the stroke(s) drawn encircling the clock face, often a long circular stroke.
- **Numeral:** any of the numbers 1 through 12 drawn around the border of the clock face.
- **Hand:** either the minute or hour hand drawn at the center of a clock face, including possible arrowheads drawn at the tips.
- **Crossed-out numeral:** any numeral, or part of numeral, that has either been scribbled out or overwritten by another different numeral.
- **Noise:** a small, unintentional stroke that does not add meaning to the clock drawing.

Command and Copy: There are actually two different phases of the clock-drawing test. The Command test is one in which a patient is "commanded" to draw a clock from memory, and the Copy test requires the patient to copy a drawing provided of an analog clock face. Each phase produces a single clock drawing, so a full run of the clock-drawing test contains two clock drawings. However, the clock the patient is asked to draw in each test is identical in subject matter, so while these two different types of

drawings add important information in clinical diagnosis, they have not proven measurably different to classify. Unless otherwise specified, I make no further distinction between them.

2.1.2 Technical Components

- **Point:** a time-stamped (x, y) coordinate recorded by the digitizing pen. Due to the sampling rate of the digitizing pen used, no points within the same drawing share a timestamp.
- **Stroke:** a sequence of points in chronological order, with its start and end determined by the pressing down and lifting up of the pen on paper.
- **Group:** a collection of one or more strokes during any phase of the classification process.
- **Symbol**: a collection of one or more strokes intended specifically to represent a single clock component, such as a numeral, a hand, or a clock outline. At the end of classification, all strokes in a drawing will be clustered into labeled symbols.
- **Label:** a classification assigned to a symbol, usually corresponding to a clock component. Examples of labels include the numerals "1" through "12", "minute hand", "crossed-out numeral", and more.

Conventions: For consistency, when calculating angles at any point, this system uses the standard Cartesian coordinate system, where positive y is oriented upwards, as opposed to the coordinate system of digital displays, for which positive y is downwards. Among other things, this means in an analog clock, "12" is located at 90 degrees instead of -90 degrees.

2.2 Data Used

I had access to a large set of clock drawings already labeled by human scorers, and chose several smaller subsets, representing different challenges, for both training and testing. Each and the challenge it poses is listed below. A full list of the drawings in each set is included in appendix A.

- **YDU-51 healthy training set:** 51 tests that have been drawn by healthy patients. The clock faces in this set generally contain all 12 numerals and have no exceptional errors present. The numerals from this data set are used to train the Ouyang nearest-neighbors numeral classifier.
- **YDU-100 healthy test set:** 100 tests similar to the healthy training set above, but a distinct set. They are used in testing to ensure that the system performs accurately on the majority of well-drawn clocks with no obvious errors.
- **VIN-96 clinical test set:** 96 tests drawn effectively at random from our corpus of clock tests, intended to represent a sampling of the average input to the classifier.
- **EGE/ORU-112 clinical test set:** 112 tests also drawn effectively at random from the corpus of clock drawings, intended as a check against specifically tailoring classification to the VIN-96 clinical set.
- **EMD-20 clinical test set:** 20 tests chosen to contain a mixture of easy-to-label and hard-to-label clock faces, used as a sample of some of the more difficult challenges in clock labeling.

The images and diagrams appearing throughout the text contain clocks drawn from each of these sets. They will also be listed individually in appendix B.

3. System Design

3.1 Overview

The ClockSketch classifier was conceived of as a system that emphasizes human-like interpretation over statistical analysis. The reason for this is that clock faces are specifically designed to be consistent and readable to a human viewer: numerals lie near the edges, while hands lie in the center; certain numerals tend to be located at certain angles; and so forth. Knowing this, I believed a system that took advantage of this inherent structure would be the most robust and adaptable to the wide range of input, degraded or not, it might receive.

This resulting classifier is structured, modular, and sequential. Each of its modules performs a classification task using a prominent element in the structure of a clock face, such as dividing up all strokes in the clock drawing into groups of hand-like and numeral-like strokes. Following modules act on these smaller subsets of strokes, breaking them down into more specific groups, until at the end all strokes in the drawing have been divided into groups and assigned labels. The description below provides a step-by-step overview of this classification process.

1. **Initial processing:** find the strokes that make up the (often circular) clock outline. With the outline, we can determine the center of the clock face.
2. **Hand-numeral division:** separate strokes into numeral-like and hand-like categories, based on how far away they are from the clock face center and when they were drawn.
3. **Numeral segmentation:** divide up the numeral-like strokes into individual numeral groups, based on angular proximity.
4. **Numeral labeling:** use digit recognition to give each numeral group a label, and record how likely each label is to be correct.
5. **Hand labeling:** finally, label the hand-like strokes as either minute hand, hour hand, or center dot.

Each of these tasks, its importance, and its implementation is described in more detail in the following sections. They describe as well the challenges faced in each step, the performance of each step, and what potential idiosyncrasies each step introduces in its output, which the following step must take into consideration.

3.2 Initial Processing

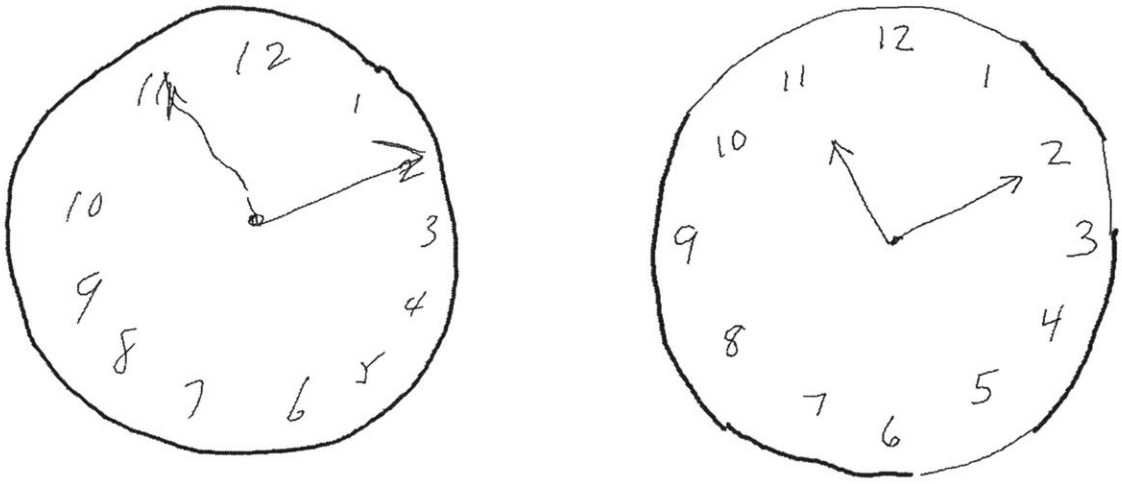
At the start of classification, the classifier has received a set of strokes in chronological order, consisting of the whole clock drawing to be labeled. Several important pieces of information need to be extracted from this set before further classification can be accomplished, and several subsets of strokes are created at the conclusion of this step.

3.2.1 Clock Outline Detection

The clock outline is a very useful stroke for classification. Most clock faces are drawn with one, and knowing a clock's outline gives the classifier several advantages. The clock outline defines the clock face and acts as a boundary beyond which strokes are less likely to contain relevant information (although exceptions, such as numerals slightly outside the clock outline, occur rarely). It can also be used to determine the location of the center of the clock face, itself useful for distinguishing between numerals and hands, determining the angular position of numerals, and other tasks. Therefore, it is one of the first strokes the classifier should either find or determine to not be present.

The majority of clock outlines are circular and consist of a single stroke. Our system thus detects clock outlines by looking for the longest stroke in the drawing, with the caveat that the point-radius of the stroke, defined here as the average distance from all of its points to its center of mass, must be longer than $1/4^{th}$ of the drawing's average radius, defined as the average of the drawing's length and width. This is to ensure that long strokes that cover a wide area, such as a stroke circling the clock face, are chosen rather than scribbles or tightly-wound spirals. If no single long stroke is found that fits these length and area requirements, the system will conclude that no clock outline exists; it cannot detect a clock outline that

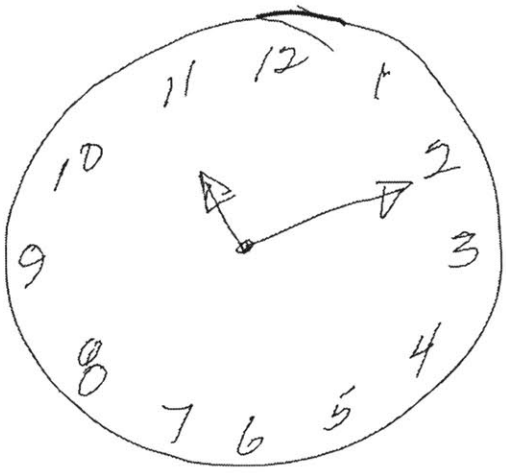
consists of a sequence of small line segments instead of a single long stroke, as the below figure illustrates.



a) b)

[Figure 3.2.1.1 Comparison of a clock whose outline is a single long circular stroke (a), and a clock whose outline is drawn as a sequence of 7 segments (b). Alternating segments are emphasized to make them easier to distinguish.]

The system also accounts for the possibility of a user "patching" their clock outline with a single-stroke addition. If a long clock outline stroke is found, the system will add another stroke to it under the following conditions. The stroke to be added must be the longest stroke in the drawing (aside from the already-found clock outline) that matches two requirements: its closest distance from the clock center must be larger than the average distance of strokes from the center, and its point-radius must be longer than $1/8^{th}$ of the drawing's average radius. These requirements preferentially select long strokes that lie far from the center of the clock and span a wide area, in order to lower the risk of incorrectly selecting a stroke that should actually belong to a numeral. Because the system will only select a single extra stroke for addition, if the patient patches the clock outline with more than one stroke, the second or shorter stroke will be overlooked.



[Figure 3.2.1.2 Clock outline patched with an extra stroke. The extra stroke is bolded.]

Using the above procedure, my system correctly labels 99.0% of clock outline strokes in the YDU-100 healthy clock set, whose clock outlines generally consist of one long circular stroke. Accuracy for the VIN-96 clinical clocks is slightly lower, at 87.8%; this is due to the presence of more complex clock outlines, whether they consist of multiple line segments (fig. 3.2.1.1) or have simply been patched repeatedly (3.2.1.2). Both will be referred to as "many-stroke outlines."

Many-stroke outlines are repairable, but my system does not try to fix them at this time for two reasons. The first, which will come up again in future sections as a general concept, is that the rarity of this case does not justify the increased false positive rate of including checks for it. Of the 416 clock drawings in the VIN-96 and EGE/ORU-112 clinical sets, only 15 drawings contain outlines with more than two strokes, so a method for detecting many-stroke outlines will only apply to at most 4% of clinical clock drawings. On the remaining 96% of drawings containing one- or two-stroke outlines, which the YDU-100 results show are 99% correctly labeled, this method will have no effect at best; at worst, it might falsely mark numeral strokes as outline strokes. Numeral strokes outnumber outline strokes on average 17 to 1; a method for successfully detecting many-stroke outlines would both have to be aggressive enough to recognize one when it occurs, *and* conservative enough to not misidentify numeral strokes as outline strokes.

Secondly, the difficulty of meeting the above requirements outweighs any benefits we might gain by correctly labeling a many-stroke outline, because while the clock outline is useful for classification, its absolute correctness is not necessary. Finding the longest circular stroke of an outline, as long as it is more than 75% complete, is sufficient to accurately determine the center of the clock drawing via elliptical fit; other strokes used to patch the outline generally have little effect on the location of the clock center. Even in the case of a missing or unidentified clock outline, the borders and center of the clock face can be estimated using the average position of all strokes within the drawing, so the information lost by not finding the clock outline can still be found. For these two reasons, this is an acceptable level of accuracy for clock outline detection.

3.2.2 Noise Detection

Another set of strokes the classifier tries to find at the very start is noise, which are small strokes that may be added to the drawing when the patient places the pen on the paper unintentionally, or due to jittering in the patient's hand. Because noise does not appear to add meaning to the clock drawing, and obscures further steps in the classification process if left in the general set of strokes, the most obvious noise strokes are extracted from the general set of strokes at this early stage.

To detect noise, the classifier looks for strokes with very few points (< 30), and whose points are highly concentrated in a small area. This results in the successful recognition of 74.2% of noise-labeled strokes in the VIN-96 clinical set, and 80.0% in the YDU-100 healthy set. Although this is not exceptional, it correctly detects most obvious noise strokes, and more importantly results in minimal false positives with numerals or hands being labeled as noise.

3.2.3 Resulting Output

Initial processing produces three subgroups of strokes: the set of strokes consisting the clock outline (which will provide useful information in future steps), the set of noise strokes (which will be set aside so as to not interfere with classification), and the set of all remaining strokes that are neither clock outline

nor noise. The rest of the classification process will focus on dividing them up into meaningful symbols and labeling them accordingly.

3.3 Hand-Numeral Division

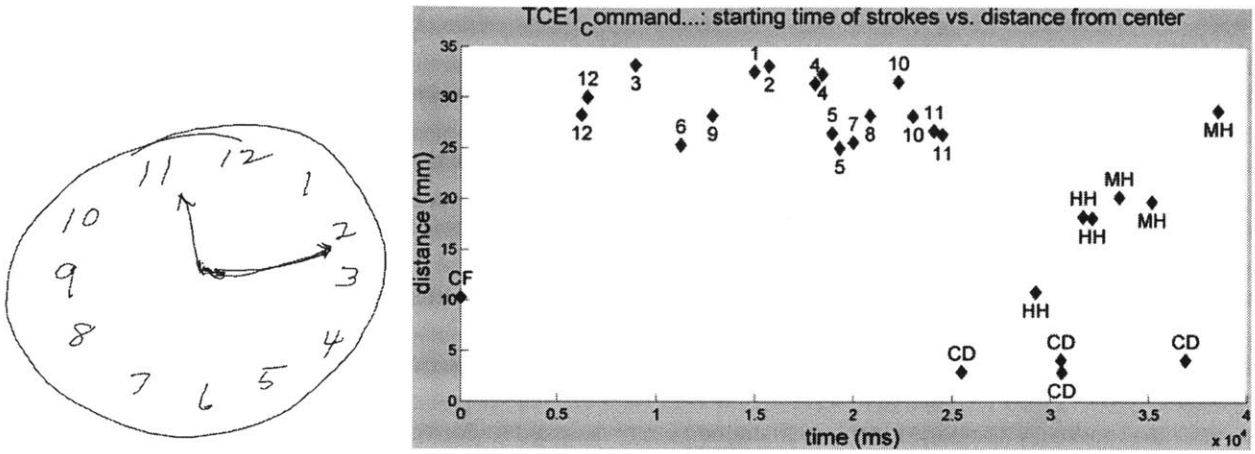
Aside from the outline, the other two essential components of a clock face are hands and numerals. Each is structured differently, and neither measurably depends on the other. Therefore, the goal of this step is to divide the set of remaining strokes into "hand-like" and "numeral-like," so that each group of strokes can be separately classified in following steps.

3.3.1 The Importance of Time

At first glance, it might seem that all that needs to be known to distinguish between hands and numerals is each stroke's distance from the center of the clock face. Strokes closer than the average distance must be hands; strokes farther than the average must be numerals. However, this strategy is confounded by two common occurrences in clock drawings; arrowheads and elliptically-shaped clocks. Many patients draw arrowheads on the tips of hand strokes, and due to their proximity to the outer edge of the clock, arrowheads will be grouped with numerals by a purely distance-based strategy. Second, if the clock is rectangular or elliptical in shape, certain numerals will end up closer to or farther from the center, and closer numeral strokes may well fall below the distance required to be considered numerals. Therefore, a strategy that relies solely on spatial location is not robust enough to handle common variations in clock drawing.

My classifier overcomes this difficulty by using both spatial and temporal data to distinguish hands and numerals. This is supported by two observations. First, the vast majority of patients draw all 12 numerals before drawing the hands of a clock, even if the order of the numerals themselves is not necessarily fixed (some patients begin with 12-6-3-9; others draw all numerals in order). Second, there is usually a noticeable gap in time between patients finishing the numerals and beginning the hands. Thus, on top of

the spatial assumption that numerals are generally farther from the center of the clock face than hands are, we also see that numerals are often temporally distinct from hands.



[Figure 3.3.1.1 A clock drawing with all of its strokes graphed as data points and marked with their ground truth labels. The graph compares the time each stroke was drawn (x-axis) with the stroke's distance from the center of the clock face (y-axis).]

Fig. 3.3.1.1 illustrates the importance of using both the spatial and temporal differences between numerals and hands. Most of the numeral strokes are indeed farther from the center of the clock face than hand strokes are, and most of the hands are drawn much later than the numerals. However, the two outliers – a center dot (CD) drawn immediately after the last numeral, and a minute hand (MH) drawn farther from the center than many numeral strokes – show that using only a single dimension is not enough to ensure numerals are separated form hands.

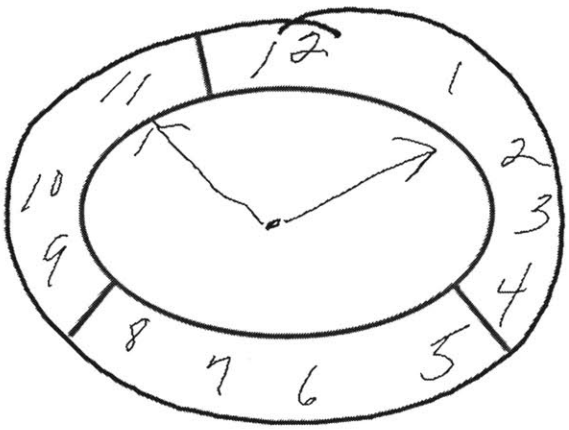
3.3.2 Spatiotemporal Clustering

To separate numeral strokes from hand strokes using both space and time dimensions, I used the k-means median-based clustering algorithm, which iteratively creates clusters of "like" data points and separates unlike data points into different clusters. In k-means, similarity is determined by a user-defined distance metric; data points are added to a cluster if they are closer to the median of that cluster than the median of any other cluster. This allows spatiotemporal clustering to be carried out easily; strokes can be represented as 2-D data points of spatial and temporal information, and their k-means distances from each other calculated using the Euclidean distance metric.

In our implementation, each stroke's data point contained its closest distance from the center of the clock face as spatial information, and its starting time as temporal information. The stroke's closest point to the center was chosen for distance-from-center calculations instead of its geometric mean, because the closest-distance metric distinguishes hand strokes (which tend to have one end very close to the clock center) from numeral strokes (which are far from the center as a whole) more effectively. These data points were then processed to ensure the two dimensions would be scored on similar scales.

- All starting times were shifted so that the first stroke being clustered began at time 0.
- Using the data of all strokes being clustered, I calculated the mean and standard deviation individually for both time and distance. This was used to convert strokes' time and distance into z-scores, i.e. measures of the number of standard deviations each differed from the mean.

Finally, I experimentally found the optimal number of clusters the algorithm should create, as well as the weights for both time and distance dimensions. A maximum of 4 clusters and a weighting ratio of 5:2 time-to-distance resulted in the smallest amount of intermixing between numeral strokes and hand strokes. When tested on the VIN-96 clinical clocks, only 25 out of 760 resulting clusters, or 3.3% of clusters, contained a mixture of numeral and hand strokes; the remaining clusters were homogeneous. Fig. 3.3.2.1 below depicts the resulting groups from one clustering.



[Figure 3.3.2.1 A clock face with its outline bolded, and the four subsets of strokes created by spatiotemporal clustering separated by borders.]

While this step has high accuracy, several common errors do occur and are responsible for the majority of "mixed" clusters. The most common case involves arrowheads, which are spatially closer to numerals than to hands. Because some are drawn immediately after numerals and before hands, they are closer to numerals and are grouped with them. The second case is that of the user "repairing" a numeral after they've finished the hands. Because the repair stroke is temporally far from the rest of the numerals, it is often placed with hands instead. Finally, some users draw a center dot before they begin drawing numerals, and due to its large temporal difference from the rest of the hands, the center dot will be included with numerals. Of these cases, the first and last case are checked for during error correction, while the second case occurs infrequently and is difficult to discover, so it is accepted.

3.3.3 Inner and Outer Groups

Having produced some number of clusters of spatiotemporally similar strokes, the next step is to consolidate them into two groups, one containing numeral-like or "outer" strokes, and the other, hand-like or "inner" strokes. Since the previous step lowered the risk of spatial outliers by grouping similar strokes together, the current step can safely use each cluster's distance from the center of the clock face to classify it as numeral-like or hand-like.

For each cluster of strokes, we calculated its average distance from the center of the clock face using the closest distance between each stroke and the clock center. We also calculate the average closest-distance-from-center of all strokes in the clock face. If the cluster's average distance is higher than .75 of the total average, it is considered part of the outer group; otherwise it belongs to the inner group.

Using this metric to divide clusters into inner and outer, we obtain the following accuracy rates for the VIN-96 clocks. Out of 967 total strokes contained in inner groups, 17 (1.8%) were numeral strokes, 938 (97.0%) were hand strokes, and 12 (1.2%) were noise strokes not caught in the previous noise-detection step. Out of 3412 total strokes contained in outer groups, 3336 (97.8%) were numeral strokes, 64 (1.9%) were hand strokes, and 12 (0.3%) were noise (11) or spokes (1).

If the clock outline was found in the previous step, this step can also create a third subgroup containing outliers, strokes that lie far outside the boundary of the clock outline. Outliers may be any of a number of things, including textual notes the patient took while drawing, scribbles, or noise, but in general they are rare and have little to do with the rest of the clock face. Outliers found here are removed to a separate symbol and labeled "not clock data", and will not be used in future steps in the classification process.

3.3.4 Resulting Output

At the conclusion of this stage, the remaining non-clock-outline and non-noise strokes have been divided into at most three separate groups, each containing spatiotemporally similar strokes. The outer group contains strokes that are likely to belong to numerals, and the inner group contains strokes likely to represent hands. The optional third group, outliers, will not be looked at further during classification.

Much of the rest of classification will be focused on the outer group produced here, as correctly dividing up strokes, labeling the numerals, and recognizing messy and overwritten symbols are the most interesting challenges faced by not only this classifier, but symbol recognition as a whole. The last section will be dedicated to handling the correct labeling of hands.

3.4 Numeral Segmentation

Having produced an "outer" group containing primarily numeral-like strokes, the next step is to divide these strokes into distinct numerals, so that each numeral can be labeled individually. As numeral strokes generally lie in a ring around the outer edge of the clock face, the simplest way to describe this task is to imagine slicing a donut radially, such that each segment of the donut represents a single numeral. Here, we rely on the angular position of strokes' geometric centers relative to the center of the clock in order to divide them into groups.

3.4.1 Angular Differences Between Numerals

The difficulty in this task comes from determining where to cut. I observed from many clocks that the angular difference between strokes from different numerals was often much larger than the angular

difference between strokes within a single numeral; two strokes that lay more than x radians apart from each other were much more likely to belong to different numerals. Therefore, "slicing" between two strokes further apart than some lower bound would be an effective way to separate numerals. I considered dynamically finding this lower bound for each clock drawing during classification, but ultimately concluded that a fixed-size bound would be more consistent and just as effective.

I experimentally determined this optimal lower bound to be .22 radians, or around 12.6 degrees, by performing segmentation on the set of EMD-20 clinical clocks (which contains more complex errors involving >12 numerals, and is a good test of the robustness of this metric). I evaluated this lower bound's accuracy by carrying out segmentation as below, and counting the number of resulting numeral groups that contained either two or more numerals, or only part of a single numeral.

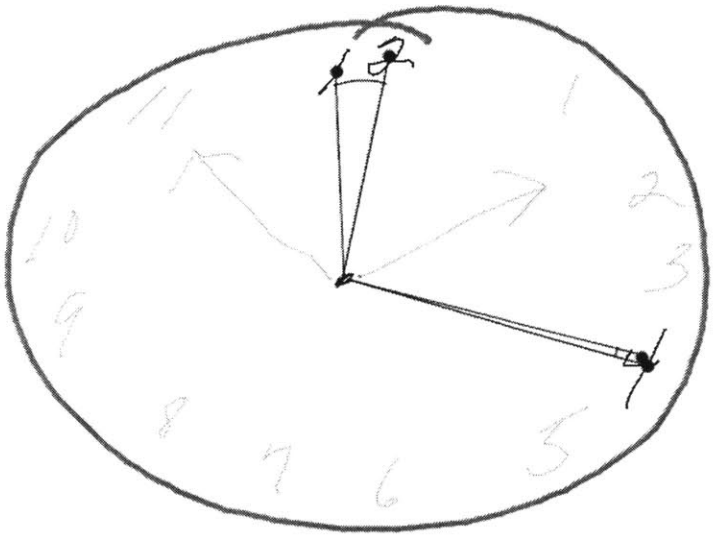
1. Order all numeral-like strokes by angular position.
2. Using the geometric center of the bounding box of each stroke, determine the angular difference between each stroke and the next (and between the last stroke and the first), relative to the center of the clock face.
3. If an angular difference is larger than the lower bound, make a cut; the strokes on either side of that cut will be placed in different numerals.
4. Repeat for all angular differences larger than the lower bound. The resulting sets of strokes represent the fully segmented numerals.

3.4.2 Resulting Output

Using .22 radians as the upper bound, numeral segmentation produced 2326 groups for the full set of 192 clock faces in the VIN-96 set, or approximately 12.11 groups per clock face. Of those 2326 groups, 32 contained only non-numeral strokes, indicating that a few hand-like strokes slipped through in the numeral-hand division step, but the remaining 2294 groups contained at least one numeral stroke. 87 groups (3.8% of 2294) contained split-up parts of numerals, while 51 groups (2.2% of 2294) contained

two or more numerals joined together; the remaining 2156 (94.0% of 2294) contained complete and unique numerals.

While these results are good, some of the most common trends and inaccuracies are worth describing, as they inform future steps in the classification process. First, multi-stroke numerals are more susceptible to being split or joined than other numerals. The 87 groups containing split-up numerals represented 48 different numerals (some of the groups contained strokes from more than one numeral), of which 29 were "12"s and 15 were "5"s. Similarly, the numerals that were most likely to be joined together were 4 and 5, and some combination of 10, 11, and 12. One observation here is that the position and structure of the numerals "12" and "5", with one stroke's geometric center located at an angular distance from the next, makes them more likely to be split up. The below figure provides an example of this for the numeral "12".



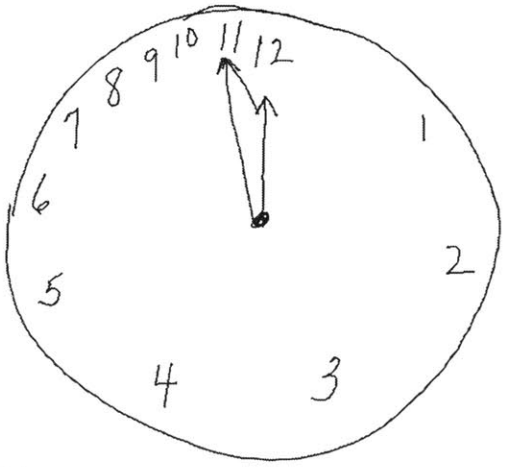
[Figure 3.4.2.1 Illustration of the large angular difference between the geometric centers of strokes in the"12", compared against the small angular difference between the two strokes of the "4".]

Second, this method of segmenting numerals fails in any case where strokes, whether or not they belong to numerals, are placed too close to each other. Three cases of this are described below:

1. Overwriting is the case in which a patient draws one distinct numeral and either crosses it out or writes a second, different one on top of it. The overwritten numeral is clinically important and should be distinguished from the one above, but because the system segments by spatial

difference and not by temporal difference, it cannot distinguish between "layers" of numerals in the same position.

2. Arrowheads or center dots may occasionally be included in the outer group. Arrowheads are often grouped together with the numerals they are closest to (often "11" and "2"), while center dots will be grouped with whichever numeral their angular position relative to the center is most similar to, regardless of the large distance between it and numerals.
3. At times, the patient does not draw numerals evenly spaced throughout the clock, placing many numerals very close to each other. In fig. 3.4.2.2 below, the numerals "9" through "12" will be placed into a single numeral group, as the angular difference between each of them is too small to be considered an inter-numeral division.



[Figure 3.4.2.2 A clock in which the patient allotted too much space to the first five numerals, and had to squeeze the rest of the numerals into a small remaining area. In this case, the numerals "9" through "12" will be combined into a single symbol, owing to the small angular differences between strokes.]

Both split-up numerals and joined numerals, regardless of the reason they were created, will be addressed below at the error correction stage. For now, our system assumes that the group of outer strokes has been correctly divided into some number of individual numerals, usually 12 but sometimes more or fewer. The next step is to assign each of these numerals a fitting label.

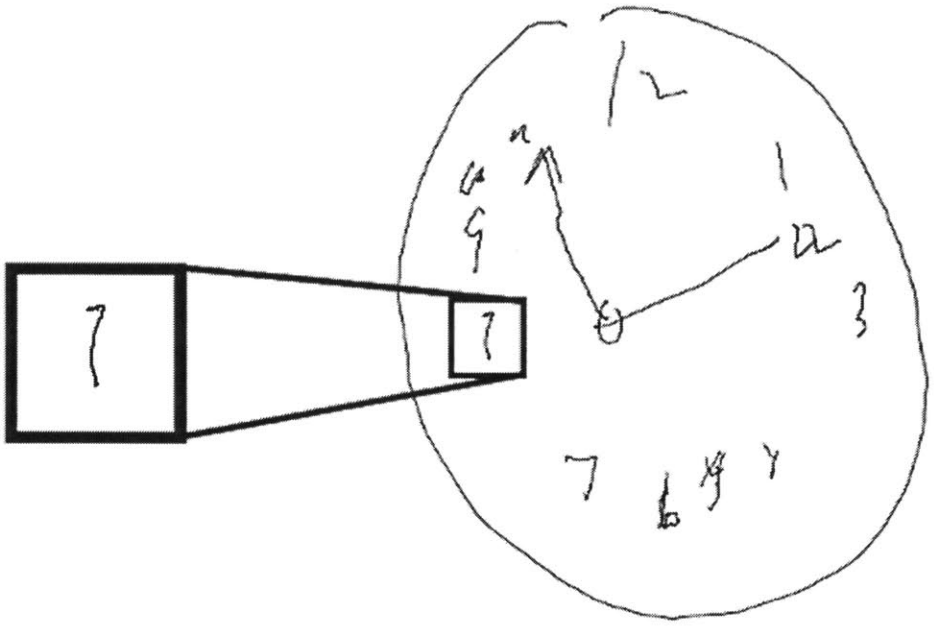
3.5 Numeral Labeling

This section deals with the problem of assigning the most-likely numerical label, such as "1", "6", or "12", to each individual numeral group produced by segmentation. These numeral groups will hereafter be referred to as symbols, as they now represent the finalized collections of strokes that will be output by classification (with some caveats, as seen later in error correction). To determine each symbol's label, I use the nearest-neighbors symbol recognition system developed by Tom Ouyang and Randall Davis (2009) [2], with some important modifications made to improve robustness and make use of the inherent structure of a clock drawing.

3.5.1 Symbol Recognition

While digit recognition is already a well-trod field, and current state-of-the-art classifiers such as Ouyang (2009) perform with upwards of 99% accuracy [2] on data sets of isolated digits, clock numeral labeling remains challenging for three reasons.

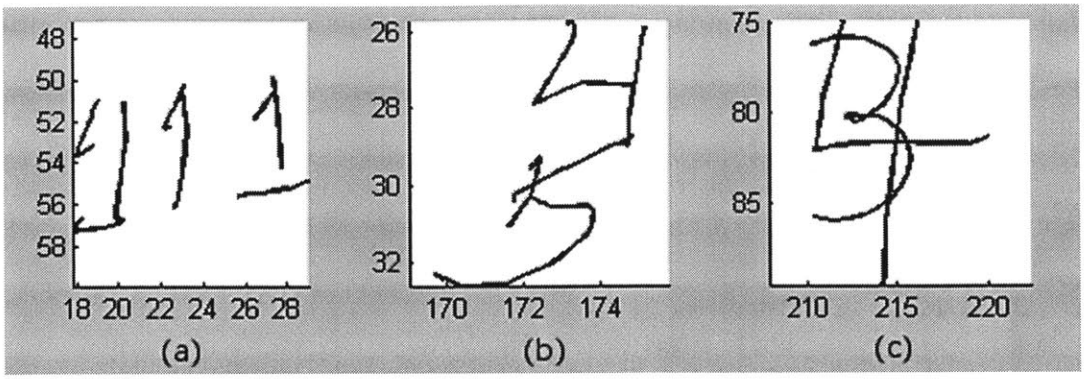
First, the correct numerical label for a symbol depends almost as much on contextual information as it does on the appearance of the symbol. The position and order of the symbol within the clock face are especially important; a symbol located near 0 radians in the Cartesian coordinate system is more likely to be a "3" than a "9". The angular order of symbols also provides valuable information, as the figure below illustrates; a symbol can appear very much like a "7" or "1" when seen in isolation, but when its angular position and the two symbols angularly adjacent to it – a "7" and a "9" – are considered, it becomes clear that the symbol was actually intended to represent an "8".



[Figure 3.5.1.1 Illustrates that a numeral's appearance is not necessarily indicative of its actual label. The "7" seen at left, when viewed in context, is actually a sparsely-drawn "8" located between a "7" and a"9".]

Second, existing digit recognition systems recognize individual digits. However, in the numeral labeling problem, multi-digit numerals such as "10", "11", and "12" are treated as a single symbol instead of two separate digits, due to the difficulty of determining which strokes belong to which digit. As a result, numerals contain more variation than an individual digit does, and the performance of digit recognition systems on them is currently unclear.

Finally, in the case of numeral labeling, there is no guarantee that the symbol being classified actually represents a single, unobscured, complete numeral. In the above section on numeral segmentation, I described the segmentation errors of splitting numerals into different symbols and clustering adjacent numerals (or overwritten ones, or numerals and arrowheads) into a single symbol. Symbols that contain these errors, when passed through recognition, will result in either a low-probability label or a completely inaccurate one. The below figure includes a few examples of these symbols.



[Figure 3.5.1.2 Examples of (a.) a split-numeral group containing an "11" and the 1 from the "12" after it, (b.) a "4" and "5" combined into a single group, and (c.) a "3" overwritten by a "4". The size of each is indicated in mm.]

These three challenges make it clear that basic digit recognition is not a panacea for numeral labeling. However, digit recognition still offers important benefits to classification in spite of – and sometimes because of – these problems. The most obvious is the ability to determine with some certainty the numeral each unknown symbol most resembles. Slightly less obvious, however, is that digit recognition can indicate *how much* a symbol resembles each possible numeral type from "1" through "12". This both allows for a more nuanced comparison of possible labels, and gives the system a robust way to determine when a symbol does not resemble any known numerals and should be scrutinized for segmentation errors.

3.5.2 Training the Recognizer

I chose to use the Ouyang (2009) symbol recognizer in this system for several reasons. It is both accurate and fast, both of which are important for clinical use. It robustly handles a wide range of input symbols, achieving high performance on hand-drawn data sets from digits to polygons to circuit diagrams (99.2%, 98.2%, and 96.2% respectively) [2]. It also provides, for each match, an intuitive score representing the distance between the symbol to be recognized and its nearest neighbor. This "best-match distance" will be essential to future error recognition and correction.

Training

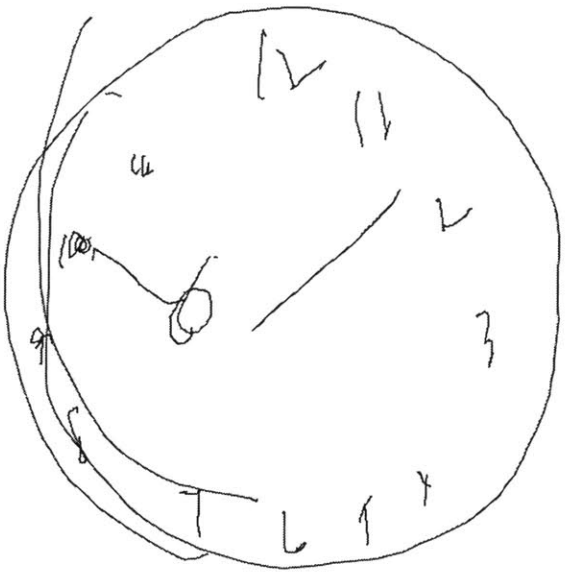
The training data used was the YDU-51 healthy training set, containing ~1212 ground truth labeled numerals from 101 clock faces, 101 per numeral class from "1" to "12". Note that although the set of clock drawings these numerals were taken from was selected to represent primarily healthy patients and

contain all 12 numerals each, the numerals themselves are not necessarily orderly or clean in appearance, and represent a wide range of drawing styles.

These 1212 numerals were not all used to train a single recognizer. Rather, 12 separate "single-numeral" recognizers were trained, each with the set of 100 numerals belonging to one numeral class. The function of these single-numeral recognizers is not to determine which numeral class a symbol most resembles, but how close it is in appearance to the best match within a specific numeral class, creating a best-match distance for that numeral. Knowing the best-match distance for each possible numeral class provides an important advantage, as it allows distances for different classes to be weighted based on symbol-specific context information, such as angular position. This weighting is described in the next subsection.

3.5.3 Angularly Weighted Recognition

One of the problems with basic symbol recognition, as mentioned above, was that it did not take advantage of the position of numerals in a clock face. In the majority of clock drawings, patients draw numerals at consistent angles around the perimeter of the clock face. Even when a numeral is messy or unrecognizable at a glance, its position and relation to potentially less messy numerals are strong indicators of what numeral was originally intended. This can be seen in the numerals drawn in the clock face below, especially the "10" and "11"; it would be difficult to interpret either correctly without knowing the context of their location and surrounding numerals.



[Figure 3.5.3.1 A clock containing numerals that are nearly unrecognizable alone, but clear in context.]

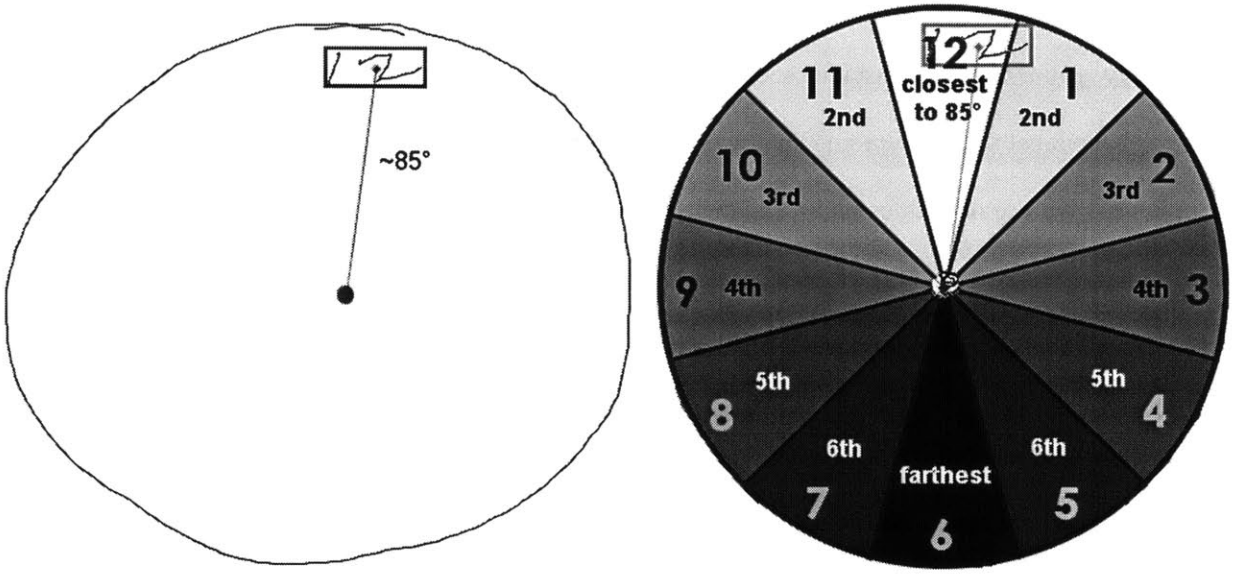
Recognizing the importance of angular position, this classifier incorporates it into the recognition-and-labeling process via weighting, as described below.

First, the symbol's closest-match distance score to each of 12 digit classes is determined. These scores are placed in a vector in numerical order from "1" to "12". Note that in this case a smaller score, indicating a smaller distance between the symbol and a match, is preferable.

Second, the symbol's angular position relative to the center of the clock face is determined, using the geometric center of its bounding box. A weight vector is then created using the symbol's angular position, containing weights for each numeral from "1" to "12". The method for assigning weights as well as the weights themselves is given below.

1. On an ordinary analog clock face, find the numeral closest in angle to the symbol being labeled. Fig. 3.5.3.2a and b show that for a symbol at 85 degrees, this is "12".
2. Starting from the closest numeral, assign a proximity label to each numeral in the clock based on its distance from the closest numeral. For example, "1" and "11" in fig. 3.5.3.2b are immediately adjacent to "12", so they are labeled as 2nd-closest. "2" and "10", separated by 1 numeral from 12, are 3rd-closest, and "6", on the opposite side of the clock face from "12", is the farthest.

3. Once each numeral has been given a proximity label, it is assigned a weight according to the values in table 3.5.3.3; numerals far from the symbol to be labeled are given larger weights.



a) A symbol at 85 degrees. b) Proximity of each numeral type to symbol in a)

[Figure 3.5.3.2 A diagram of how angular weights are calculated for each numeral type based on the angular location of the symbol to be labeled. a) shows the symbol to be labeled with its angle relative to the clock center, and b) shows an ordinary analog clock face with each of its numerals labeled by its angular proximity to the symbol in a).]

Proximity to Symbol	Angular Weights
Closest	1.0
2nd	1.25
3rd	1.5
4th	1.66
5th	1.75
6th	1.8
Farthest	2.0

[Figure 3.5.3.3 Table of angular weights based on the proximity of each numeral to the symbol being labeled. Numerals farther away are assigned larger weights.]

Numeral	1	2	3	4	5	6	7	8	9	10	11	12
Weight	1.25	1.5	1.66	1.75	1.8	2.0	1.8	1.75	1.66	1.5	1.25	1.0

[Figure 3.5.3.4 Vector of angular weights from "1" to "12" for the symbol depicted in fig. 3.5.3.1. "12" is closest to the symbol and thus has the smallest weight, while "6" is farthest from the symbol and has the largest.]

The table in figure 3.5.3.3 shows the set of weights currently used by the classifier, ranging from 1.0 for the closest numeral to 2.0 for the farthest. Because users don't usually draw symbols at the exact angles

they would be on an ordinary clock, the 2nd closest numerals to the symbol are not penalized very heavily, allowing some leeway for a clock drawn at a small angular offset.

Once the angular weight vector is built, as shown in fig. 3.5.3.4, its weights are then multiplied element-by-element with the vector of distance scores, and the numeral with the smallest resulting distance score is chosen as the label for the symbol. That weighted distance score is also recorded as a measurement of how much the symbol resembles its given label.

3.5.4 Resulting Output

Testing this labeling system on the set of VIN-96 clinical clocks, I found that angular weighting provides a measurable improvement in accuracy over un-weighted scores. Out of 2156 correctly segmented numeral symbols (not including symbols containing split or joined numerals), recognition without angular weighting accurately labeled 2067 symbols (95.9%), while angularly weighted recognition accurately labeled 2137 (99.1%), approaching the accuracy levels of the Ouyang system on the Pen Digits dataset [2]. This strongly supports the importance of angular context in clock numeral labeling.

As mentioned previously, correctly segmented numerals make up 94% of all symbols containing numeral strokes. The remaining 6% of numeral symbols, containing split, joined, or overwritten numerals, cannot be meaningfully labeled as-is; they need to be detected and repaired before they can be labeled. Furthermore, 32 symbols produced by segmentation contained non-numeral strokes entirely, such as hands or noise; they should not be labeled as numerals either. Section 4 will describe in more detail the steps that need to be taken to detect and repair such errors. For now, we move to the last stage of basic classification, that of labeling the hand strokes.

3.6 Hand Labeling

To some extent, labeling the hand strokes in a clock face is much less complex than labeling numerals. There are only three common labels for hands: the hour hand and minute hand, which are self-explanatory, and the center dot, which is defined as a small circular stroke at the center of the clock face.

Most other types of strokes, such as crossed-out hands, uncaught noise strokes, and spokes (long strokes that cross the entire clock face), are rare. The format of the clock-drawing test also lowers the ambiguity involved in drawing hands; because clinicians ask their patients to draw the time as "10 after 11", the minute and hour hands should be on opposite sides of the clock center in a correctly drawn clock face, with the hour hand pointing towards 11 and the minute hand pointing at the 2. This allows an individual hand or center dot to be distinguished more easily.

However, hand labeling also runs into some challenges due to having less structure than numerals, particularly when separating one hand from the other. In an ordinary analog clock, the hour hand is distinguishable from the minute hand by being shorter and sometimes thicker. However, many patients do not draw the hour hand noticeably shorter than the minute hand, making length less useful for determining which is which; neither do they "thicken" the hour hand. Thus the best way of distinguishing hands has to rely on the test's structure – that is, assuming each hand correctly points towards opposite edges of the clock – to separate hour from minute hands.

The hand labeling procedure I decided on makes use of the angle of each stroke relative to the center of the clock face to distinguish between hour and minute hands, with a separate check for center dots beforehand. As below:

1. Calculate both the average distance from center and the average length of all hand strokes being labeled, using each stroke's geometric center for the distance from center.
2. For each stroke, first determine whether it is a center dot, defined as the maximum dimension (length or width) of the stroke's bounding box being less than .33 of the average length of all strokes, and the stroke's *farthest* point from the clock center being closer than the average distance from center. Effectively, this ensures tiny strokes near the center of the clock face are marked as center dots.

3. If a stroke does not meet criterion for being a center dot, obtain the angle relative to the clock center of its farthest point from center. If this angle lies between -90 degrees and 90 degrees – that is, the right side of the clock – then the stroke belongs to the minute hand. Otherwise, it belongs to the hour hand.

3.6.1 Resulting Output

On the VIN-96 clinical clocks, this procedure resulted in the correct classification of 392 out of 412 hour hand strokes (95.1%), 333 out of 358 minute hand strokes (93.0%), and 148 out of 158 center dots (93.7%). Of the 20 mislabeled hour hand strokes, 17 were marked as center dots and 3 as minute hands; of the 25 mislabeled minute hand strokes, 19 were marked as hour hands and 6 as center dots; and of the 10 mislabeled center dots, 4 were marked as hour hands and 6 as minute hands.

The above accuracy rates are good, but indicate a tendency to label some hour hand strokes as center dots, and some minute hand strokes as hour hands. This occurs when users draw hands as a set of smaller strokes instead of one or two long strokes, as when these smaller strokes are close to the center of the clock face, they are more likely to be incorrectly identified.

Another thing to note is that this error rate does not take into account non-hand-strokes present in the inner group. 26 strokes belong to crossed-out hands, which are currently not accounted for due to the difficulty of detecting when a hand has been overwritten. 32 are strokes from numerals, noise, or other components entirely. All of these will be mislabeled as minute hands, hour hands, or center dots unless they are otherwise detected.

Finally, this section can only label the hand strokes present in the inner group itself; it has no way of accounting for any hand strokes that may have been incorrectly placed into the outer group during hand-numeral division. Those hand strokes will eventually be labeled as numerals.

3.7 Results

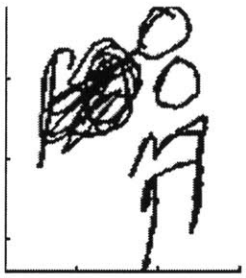
Having described the basic ClockSketch classifier in full, I include a summary of its stroke-labeling accuracy on the VIN-96 clinical set, as well as a breakdown of accuracy rates by stroke type. The table below includes entries for many stroke types besides hands, numerals, noise, and the clock outline for completeness; however, many of them are not yet labeled by the basic classifier.

VIN-96 Clinical Set	Accuracy	# Strokes	# Containing Clocks
All strokes	89.4%	4771	192
Numeral strokes	93.5%	3346	192
Hand strokes	83.5%	1044	192
Crossed-out numerals	0.0%	19	13
Noise strokes	74.2%	89	53
Tick marks	0.0%	1	1
Clock outline strokes	87.8%	230	192
Other	0.0%	41	12

[Figure 3.7.1 The accuracy rates of classification on each type of stroke for clock drawings from the VIN-96 clinical set.]

These results are promising; the system achieves good performance on numeral and clock outline strokes, and a little less on hand and noise strokes. Three categories of strokes - crossed-out numerals, tick marks, and the remainder of strokes represented by "other" - are not yet handled by the classifier.

However, if we want to improve the accuracy of the system further, the next step is to take a closer look at why numeral strokes are being mislabeled. As indicated above, numeral strokes make up the vast majority of strokes within a clock face, and even a small increase in the accuracy rate of labeling numerals would improve the performance of the classifier measurably. Furthermore, while detecting and labeling edge cases like tick marks is a relatively simple process, the errors of numeral labeling tend to be caused by symbols like the one below.



[Figure 3.7.2 An example of a heavily overwritten and re-overwritten symbol containing at least three potentially-recognizable numerals.]

The problem of mislabeled numeral strokes has as much to do with incorrectly grouped or overwritten numerals as it has to do with recognition error. The task of determining how to correctly classify a symbol like figure 3.7.2 above is a more general and more useful problem to solve than accounting for various rare cases. Therefore, the following section will focus on error correction for numeral symbols, and specifically that of unraveling the complexities of numerals that have been poorly grouped or overwritten.

4. Error Correction

The per-stroke labeling accuracy of the basic system outlined above is sound, especially on the most frequent classes of strokes encountered – numerals, digits, and the clock outline. However, to improve the performance of the system, account for crossed-out numerals, and build a more robust and unique classifier, we must now focus on the most common errors faced by numeral symbols, and work to repair them as smartly and as accurately as possible.

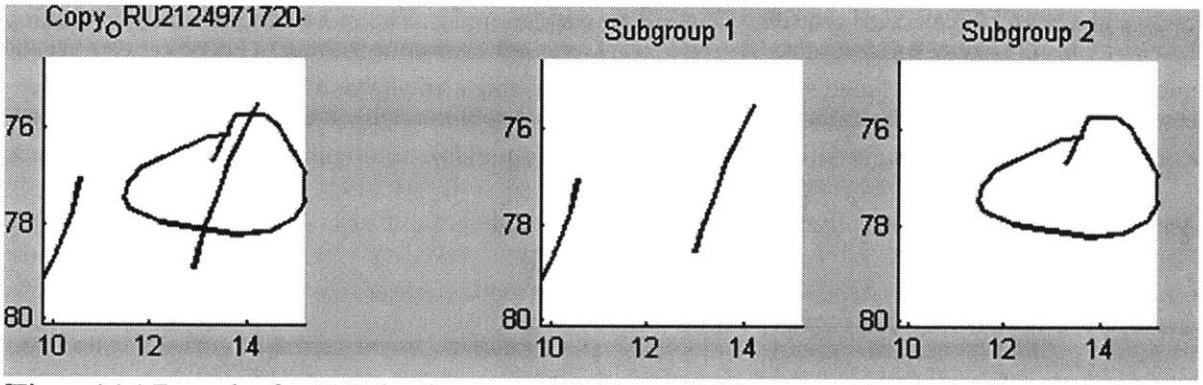
This section will first describe the errors in question, then outline the system's strategy for detecting and repairing them. I also include explanations for where and how this strategy may fail, and what can be done to remedy these failures in the future.

4.1 Common Errors

While some numerals are simply mislabeled by symbol recognition, the majority of numeral-labeling errors stem from incorrect grouping of numeral strokes. The most prominent of these errors are described below.

- **Split numerals:** A multi-stroke numeral may have its strokes divided into two symbols during segmentation. The two halves are unlikely to be correctly labeled during recognition, as neither half resembles the intended numeral. For example, a "12" may have its two digits split into separate symbols, which will be labeled as "1" and "2" instead. Split numerals need to be recombined into a single symbol before they can be correctly recognized; however, the system must be careful not to do this overzealously, such as combining the "1" and "2" present in a normal clock face into a single symbol.
- **Joined numerals:** Strokes from multiple numerals may be combined into a single symbol by segmentation. Joined numerals need to be divided into their constituent numerals before they can be correctly recognized.

- **Overwritten numerals:** This is a sub-category of joined numerals, in which instead of incorrect segmentation leading to two adjacent numerals being grouped together, one numeral is written directly over another and both are included in the same symbol. For example, a user may accidentally write a second "8", then overwrite a "9" on top. In this case, the two (or more) overwritten numerals need to be put in separate symbols, but only the numeral on top should be assigned a numerical label; the rest of the overwritten numerals must be labeled as crossed-out. While this usually means the last-written numeral is the intended one, some cases of overwritten numerals are more complex; the example below shows a "11" that the user only partially overwrote in order to turn it into an "10." In this case, the first "1" should not be considered crossed out and should be included with the last "0."



[Figure 4.1.1 Example of an "11" that the user modified into a "10" by overwriting the 1 with a 0. The leftmost graph shows the combined symbol, while the other two list the two numerals drawn in temporal order.]

- **Arrowheads labeled as numerals:** Drawn at the tips of hands, arrowheads tend to be closer to numerals and are often grouped with numerals during the hand-numeral separation step. Most arrowheads included with numerals are close to the "11" and the "2" (due to the time being drawn as 11:10), and are often included with them in a single symbol. The difficulty of fixing this error comes both from separating arrowheads from the numerals they're included with, and then recognizing that they are arrowheads and not numerals – without labeling legitimate numerals as arrowheads.

- **Center dots labeled as numerals:** This is a separate problem from arrowheads, and a little more specific. Center dots that are drawn before numerals tend to be grouped with numerals during the hand-numeral separation phase, because they are much closer in time to numerals than to the hands. However, because they're much closer to the center of the clock face than arrowheads are, it is easier to pick them out from numerals.
- **Numerals labeled as hands:** Occasionally, the converse of the above problem occurs when numerals are either drawn too close to the clock center, or drawn very shortly before the hands, and end up grouped with hands instead. These numerals should be detected and moved into the set of numerals, but ensuring a low false positive rate is difficult here as well.

The errors of symbols containing split, joined, and overwritten numerals are dealt with in the following sections, as well as the error of grouping center dots with numerals. The more general errors of finding numerals that have been included with hands, or distinguishing arrowheads from numerals, are not accounted for in the current classifier; their difficulties and possible solutions will be described in section 6, which deals with the future direction of this project.

4.2 Error Detection Strategies

Correctly detecting an erroneous symbol is even more important than correctly repairing it, and it is better to overlook an error during detection than to be overzealous and include a false positive. This reasoning stems from the relative rarity of errors in classification compared to the majority of correctly-grouped, correctly-labeled clock faces. For this classifier to be robust to a wide range of input, it must necessarily ignore the rarest cases and focus on the most common and identifiable.

With that said, several of the errors mentioned above are indeed identifiable with a low false positive rate, by making use of information from multiple sources. This information includes the match-distance score produced by the Ouyang symbol recognizer, the number and position of strokes in a symbol, and the time between strokes in the same symbol. Each of these pieces of data tells us something about a different

possibility of error in the symbol, and a combination of them helps the system greatly narrow down the potential error candidates. Each of the subsections below describes one of these error detection metrics and how it is used.

4.2.1 Match-Distances

The primary way my system detects error candidates is by checking whether a symbol's match with its numerical label "looks bad enough," with the assumption that symbols that already resemble their labels don't need to be processed further. During numeral labeling, the Ouyang symbol recognizer outputs both a label and a best-match distance for each symbol, and this match-distance can be compared against some upper-bound distance to determine whether it indicates a poor match.

However, obtaining match-distances that can be compared is a multi-step process. The first step is to convert the raw distance scores output by the recognizer into a more meaningful and consistent standard. The second step is to determine in this new standard, either experimentally or logically, the best upper bound of "messiness" to use. Only then can a match be measured as poor or not.

To create a consistent and comparable standard for match-distances, I provide the following intuition: knowing what a "correct" label looks like allows me to guess at whether an unknown label is correct as well. More specifically: if I know the average match-distance for a correctly labeled symbol in a specific numeral class, I can compare this average correct match-distance to the match-distance of an unknown symbol labeled as that numeral class. For example, if the match-distance of an unknown symbol to "12" is lower than or equal to the average match-distance of ground truth "12"s to "12", the symbol is likely to actually be a "12" as well. If the unknown symbol's match-distance is significantly greater than the average, however, I can conclude that the symbol's match to its assigned label is poor, and likely incorrect.

I implement such a standard by converting each symbol's raw distance from its label into a z-score, defined as the difference between a value and its mean, measured in terms of its standard deviation. I used

the set of ground truth numerals from the YDU-100 healthy testing set (containing 200 examples of each numeral) to calculate the mean and standard deviation of correct match-distances for each numeral class *i*. The z-score for each symbol's match-distance d_i, where i represents the class of numeral that the symbol was labeled as, is $(d_i - \text{mean}_i) / \text{stdev}_i$.

Once z-scores have been computed, the next step is to determine what the best upper bound of messiness is; how many standard deviations above the mean does a match-distance have to fall for it to be considered a poor match? In order to make a more informed decision, I first calculated the average z-scores for each individual "type" of numeral symbol produced during segmentation, with specific attention to how the z-scores of split-numeral or joined-numeral symbols differed from the z-scores of correctly segmented numeral symbols. I also performed this computation using both the VIN-96 and the EGE/ORU-112 clinical sets, to avoid bias towards a single data set.

VIN-96 numeral data	Mean	Stdev	# Groups
Split-numeral	2.868	2.507	87
Joined-numeral	3.832	1.802	51
Non-numeral (hands, noise)	5.238	2.662	32
Correct numeral	0.624	1.405	2156

[Figure 4.2.1.1 Mean and stdev of z-scores for all symbols created during numeral segmentation of the VIN-96 clinical set.]

EGE/ORU-112 numeral data	Mean	Stdev	#Groups
Split-numeral	2.519	3.268	66
Joined-numeral	4.56	2.002	44
Non-numeral (hands, noise)	5.594	3.196	90
Correct numeral	0.459	1.448	2447

[Figure 4.2.1.2 Mean and stdev of z-scores for all symbols created during numeral segmentation of the EGE/ORU-112 clinical set.]

Both fig. 4.2.1.1 and 4.2.1.2 show a clear difference between the average z-scores of correctly segmented numerals, which are low, and the much higher z-scores of split, joined, or non-numeral symbols. From these, we can also conclude that a z-score of 2.0 is around one standard deviation higher than the average z-score of a correctly segmented numeral symbol, and that anything higher than 2.5 is likely to represent one of the incorrectly segmented symbol types.

To detect "problematic" symbols, or symbols that are likely to be erroneously labeled or poorly segmented, I decided on 3.0 as a hard upper limit (above which a symbol is clearly problematic) and 2.5 as a soft limit. Because the soft limit has a higher risk of false positives, it relies on the presence of another indicator of error: the number of strokes in the symbol. The next subsection describes this.

4.2.2 Stroke Count

Most numerals in a clock face are drawn with one or two strokes, including the multi-digit numerals such as "10" and "12". While an ordinary numeral may contain more than two strokes, such as when the user adds another stroke to repair a less-well-drawn part of the numeral, the presence of more than two strokes in a symbol produced by segmentation more often suggests that multiple numerals have been included in the symbol. Because this is not as clear an indicator of error as we would like, it is used only in conjunction with the soft upper-bound on z-scores described above.

4.2.3 Temporal Differences

Large differences in the time strokes in the same symbol were drawn can be a telltale sign of the presence of multiple or overwritten numerals. This is because a single numeral, even a multi-stroke one, is usually drawn entirely as a complete unit before moving on to the next; a large gap in time between one stroke and the next in chronological sequence usually suggests that the lagging stroke belongs to a separate numeral entirely.

Detection of a temporal difference can take two forms: sequence and timing. When a symbol's strokes are not consecutive, this strongly indicates that the non-consecutive strokes belong to different numerals. When all strokes in a symbol are consecutive, determining whether multiple numerals are present becomes less certain. However, experimental evidence using the VIN-96 clinical set shows that a gap in time of at least .3 seconds between one stroke and the next tends to indicate that the two strokes belong to different numerals and can be split into different symbols. Because this method of detection has a higher

false positive rate, it is used only after a symbol has been labeled as problematic by virtue of having a high match distance.

4.2.4 Spatial Differences

One would expect spatial differences to be a strong indicator of whether a symbol contains multiple numerals, since two sets of self-intersecting strokes separated by a large amount of empty space clearly represents multiple numerals to a human observer. However, in the set of VIN-96 clinical clocks, nearly every case involving a spatial difference between two distinct numerals could have been just as easily detected using the temporal difference between the numerals. Furthermore, relying on temporal difference actually results in a lower false positive rate than relying on spatial difference, due to the problem of multi-digit numerals like "10", "11", and "12" containing built-in spatial differences between digits.

As a result, while the current system includes a detector for determining whether a symbol contains spatially separate groups of strokes, it is effectively unused due to the temporal-difference detector being more effective.

4.2.5 Size

Because the size of numerals tends to fluctuate between each numeral class and user, size is only used to detect center dots that have been included within a numeral symbol. As center dots tend to be very close to the clock center while numerals are farther away, the distance spanned by a symbol containing both a center dot and numerals will be greater than the average distance from each stroke to the clock center. Therefore, by comparing the maximum dimension of a symbol against the average stroke-to-center distance, we can detect symbols containing center dots.

4.2.6 Current Implementation

Having described the major error detections methods above, I now summarize how they are used to detect and categorize each type of error. The correction of each error is covered in section 4.3, which deals specifically with repair strategies.

Center Dots

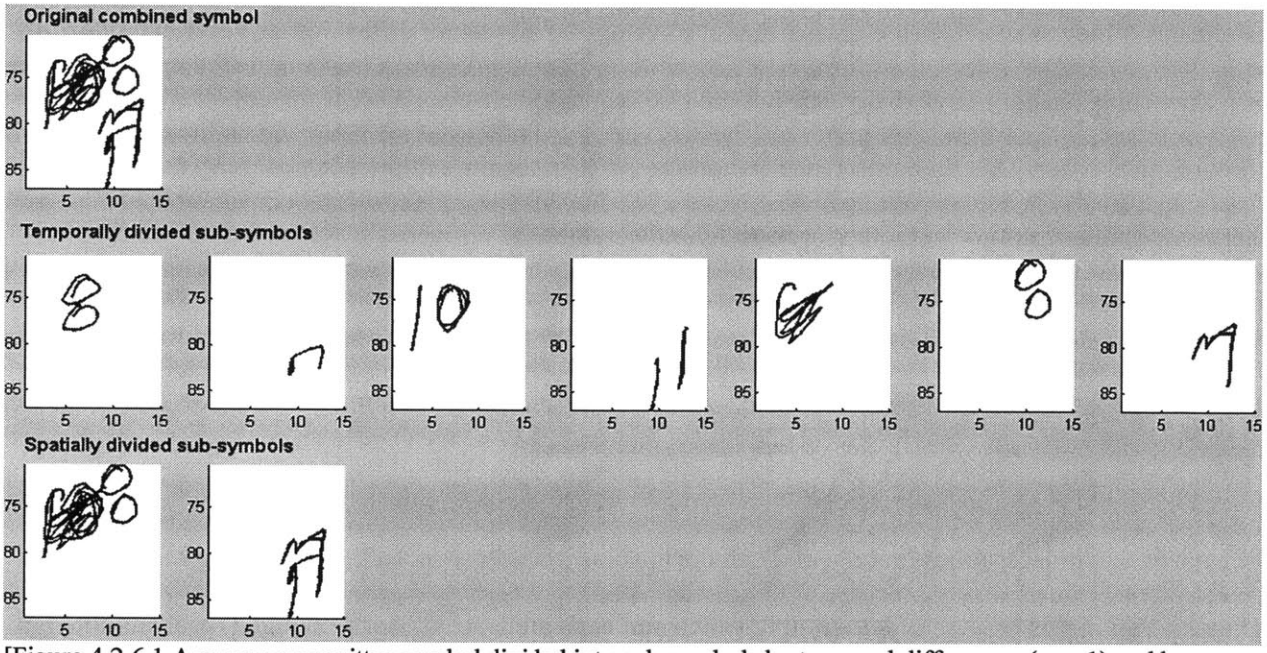
Separating center dots from numerals is done immediately after numerals have been segmented into individual symbols, but before they are labeled. Numeral symbols are marked as center-dot-containing candidates if they span a distance greater than the average stroke-center difference, as described in 4.2.4, or if they contain a very large difference in time (3.0 seconds) between adjacent strokes listed in chronological order. This time difference is much larger than the time difference necessary to conclude that a symbol has multiple numerals, so as to limit the number of erroneously detected multi-numeral symbols.

Joined and Overwritten Numerals

Hunting for joined or overwritten numerals is done only after numerals have been both segmented and labeled, as this process relies more heavily on the match-distance produced by the Ouyang recognizer. The first step in this process is determining which numerals are labeled poorly enough ("problematic") to be checked for errors, in order to avoid false positives. To do this, all numeral symbols' match-distances are converted into z-scores and compared against the upper bounds specified in subsection 4.2.1. Z-scores higher than 3.0 automatically indicate that the symbol is problematic; symbols with a z-score between 2.5 and 3.0 that contain more than two strokes are also labeled as problematic.

Problematic symbols, thus determined, are then further broken down by categories. Symbols that contain non-consecutive strokes are immediately marked as requiring temporal division during the repair step. Otherwise, symbols that contain more than two strokes are first checked for the presence of a temporal difference between consecutive strokes greater than .3 seconds; if one is found, they are also marked for temporal division. Any remaining symbols are checked for the presence of a spatial difference between strokes, defined as whether the symbol's strokes can be divided into at least two separate groups, each of which intersects only strokes within its own group. If such non-mutually-intersecting groups of strokes are found, they will be marked as separate symbols in the repair phase. Finally, symbols without either a

spatial or a temporal difference are simply accepted as-is. The figure below shows the different divisions that can be made for a heavily overwritten and joined symbol, based on either temporal or spatial differences.



[Figure 4.2.6.1 A messy overwritten symbol divided into sub-symbols by temporal differences (row 1) and by spatial differences (row 2).]

Interestingly, while the spatial division depicted is closer to how a human might choose to separate the original symbol, the temporal division separates the combined symbol into more individually meaningful components. We will see in later sections that temporal division is in general more effective at distinguishing individual components, especially overwritten ones, than spatial division.

Problematic symbols with only two strokes are handled a little differently from the above. If such symbols do *not* contain a spatial difference, then they are checked for a temporal difference and either marked for temporal division or returned. The intuition behind this choice is that a two-stroke symbol with the strokes separated by space is more likely to be an ordinary two-digit numeral, such as "12", than to be a true error. This is a more conservative approach that limits the number of errors that can be detected in two-stroke symbols, but it is necessary to avoid a higher risk of false positives. A one-stroke problematic symbol is simply returned as-is at this time.

Split Numerals

I consider split numerals separately from joined/overwritten numerals, because split numerals can't be consistently detected using a z-score messiness cutoff. For example, if a "12" is split into its constituent "1" and "2", those two symbols will likely be recognized as the numerals "1" and "2" with apparently reasonable match-distances, even when they differ in angle from the ordinary locations of "1" and "2." Instead, detecting split numerals is more of a trial and error process which, because it incorporates both detection and repair, will be described in more detail in section 4.3.

4.3 Repair Strategies

In all of the above errors mentioned, mislabeling strokes occurred as a result of the classifier poorly grouping strokes, whether it was by placing strokes from different components into the same symbol, or separating strokes that make up a single component into multiple symbols. Error repair is thus defined as re-grouping strokes such that only the strokes making up a single component are placed in the same symbol.

Each of the three categories of errors focused on during detection requires a different repair strategy. These repair strategies are described below with attention to their advantages and disadvantages.

4.3.1 Center Dot Removal

Once a numeral symbol has been detected as possibly containing a center dot, the system will determine whether one is actually present, by looking at all strokes within the symbol. If a stroke is closer to the clock center than .6 * the average closest distance of all strokes from the clock center, it will be removed from the original numeral symbol and added to the inner group. This does not necessarily mean it will be labeled as a center dot during the hand labeling phase, just that it is a stroke that is far more likely to be a hand than a numeral, and thus should not be in a numeral symbol. All other strokes that do not fit this criterion are left unmodified, and the symbol returned.

During testing on the YDU-96 clinical set, using a value of .6 * average stroke distance-from-center results in 17 strokes being moved from numeral symbols to the inner group, 16 of which were hand strokes. Testing on the EGE/ORU-112 set resulted in 37 strokes being moved from numerals to hands, of which 22 were actual hand strokes. Of the remaining 15 strokes, only 3 were digit strokes. The remaining 12 strokes represented other classes of object often close to the clock center, such as spokes (which cross the entire diameter of the clock face).

Advantages and Disadvantages

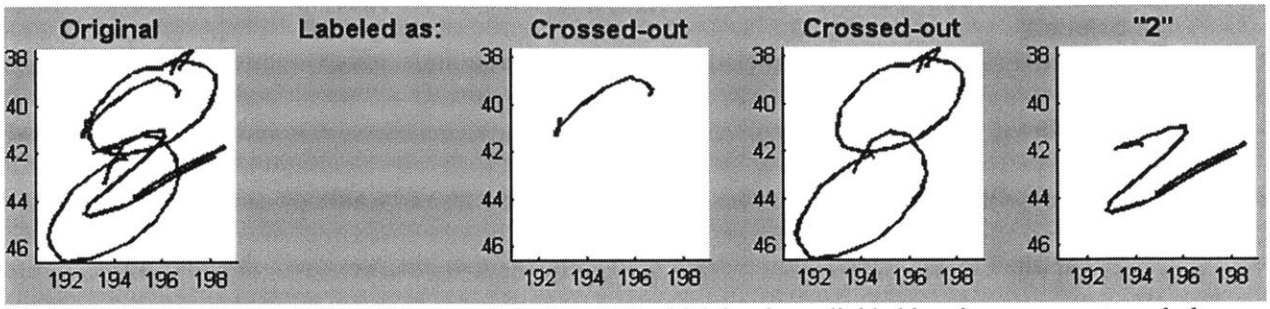
While this process has a reasonable accuracy rate, it is not extensible to removing hand strokes in general from numeral symbols. Any approach that assumes strokes close to center are hand strokes will still overlook arrowheads, which tend to be far closer to numerals. Furthermore, as the EGE/ORU-112 results show, it will also catch other strokes that may be close to the center of the clock face but are not necessarily hands. This is not necessarily an error, but this process should be improved in the future to encompass additional classes of strokes.

4.3.2 Temporal Segmentation

When a symbol contains a large difference in time between its strokes, it indicates that the strokes on either side of the temporal gap should be considered separate symbols. The division process simply orders all strokes in the symbol chronologically, then separates two groups whenever there is a non-consecutive stroke or a time difference of > .3 seconds between two consecutive strokes. The biggest challenge here is correctly labeling each of the sub-symbols resulting from this division process, and determining which sub-symbols should be considered crossed-out numerals instead of ordinary numerals.

A crossed-out numeral is defined as either: a stroke that is recognizably a numeral but has been overwritten by a scribble or another numeral, or the scribble itself that is used to cover a crossed-out numeral. A recognizable numeral that overwrites another is given a numerical label as usual. The image

below of a crossed-out numeral, next to the sub-symbols it was divided into, illustrates which numerals should be considered crossed-out and which should not.



[Figure 4.3.2.1 An example of an "8" overwritten by a "2", which has been divided into its component symbols (including an earlier stroke drawn before the "8"). The component symbols are listed in chronological order from left to right; the "2" is the last to be drawn.]

This definition of crossed-out numerals, and the necessity of distinguishing them from two ordinary numerals that happen to be included in the same symbol, exposes the two difficulties of labeling the sub-symbols that result from temporally segmenting a single symbol:

1. Can this sub-symbol be considered a numeral or a scribble?
2. If this sub-symbol is a numeral, is it covered by a later sub-symbol, which would make it a crossed-out numeral instead? Or is it sufficiently far from other sub-symbols that it should be considered an ordinary numeral?

These two considerations shape my system's method of labeling sub-symbols, described in more detail below, with a brief addendum describing how scribble detection is handled.

Sub-symbol Labeling

Once a symbol has been temporally divided into sub-symbols by either non-consecutive strokes or by time differences of > .3 seconds, these sub-symbols are chronologically ordered. Beginning from the last drawn sub-symbol in the list:

1. Determine whether or not it can be considered a scribble. To be labeled as one, the sub-symbol must both be a poor match as a numeral (with a z-score of >3.0) and be recognized as a scribble by the scribble detector. If it is not a scribble, it will be assigned a numerical label via recognition.
2. Whether the first sub-symbol is labeled as a numeral or a scribble, iterate over each sub-symbol drawn before it.
    a. If the first sub-symbol overlaps at least 20% of the previous sub-symbol's area (as defined by the intersection of their convex hulls), then the previous sub-symbol is considered to be overwritten; it will be labeled as a crossed-out numeral.
    b. Otherwise, leave it unlabeled.
3. After all previous sub-symbols have either been labeled or ignored, repeat steps 1-2 for remaining sub-symbols, moving forwards from the end of the list, until all sub-symbols have been labeled.

In short, this process labels all sub-symbols that are not covered by later strokes as numerals. Any sub-symbols that are scribbles, or overlapped even to a small extent by a later sub-symbol's strokes, will be labeled as crossed-out numerals instead.

Scribble Detection

I define a scribble, or a stroke used to cross out another stroke in the drawing, as a stroke containing many more sharp turns than any stroke from an ordinary clock component such as a hand or numeral. The boundary for "too many sharp turns" is set as 5.

The number of sharp turns in a stroke is calculated using a fixed-width moving window; at each point p(i), the direction of the line segment from p(i) to p(i+1) is compared to the direction of the line segment from p(i +WINDOW_WIDTH) to p(i+WINDOW_WIDTH+1). If the magnitude of the change in direction is greater than 120 degrees, the turn is considered sharp. This means that each sharp turn will likely be recorded up to WINDOW_WIDTH times, so a stroke with more than 5 * WINDOW_WIDTH recorded sharp turns will be considered a scribble. In practice, WINDOW_WIDTH is set to 7, which lowers the

risk of a small jagged edge being labeled as a prominent sharp turn, but is generally not so large as to miss a legitimate sharp turn entirely.

This method of detecting scribbles does not account for a circular scribble or a spiral, in which sharp turns tend to be limited but the change in direction is still present. A more robust future implementation may involve tracing a stroke from start to finish, and recording each time its direction changes by more than PI/2 degrees.

Advantages and Disadvantages

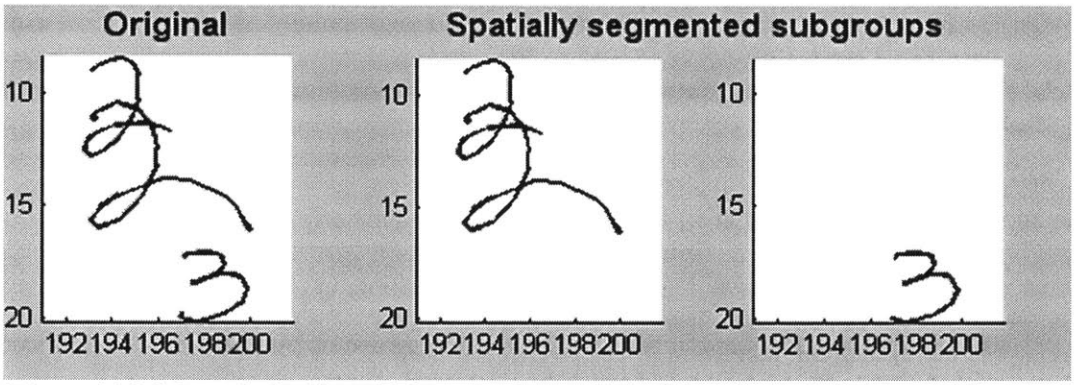
Temporal segmentation and the resulting labeling has been shown to be quite effective at breaking up complex overwritten numerals that even humans would have trouble making out, as fig. 4.3.2.1 illustrates. However, it is vulnerable to a few potential errors.

False positives are a constant concern for temporal segmentation, because the amount of time users take to draw one numeral versus another is highly variable. Symbols may be erroneously segmented if the user either draws two strokes in the same numeral with a long pause between them, or draws a numeral and then returns to it much later to "repair" it, such as adding "cap" strokes to the tips of an "11". These later-added strokes may be unnecessarily split off into a separate symbol of their own, when they should actually be kept with the numerals they add to. A way of determining which sub-symbols actually represent "add-ons" may be useful in the future.

Temporal segmentation also misses a few complex cases of overwritten numerals. In one example, a user drew a second "10" after their first "10", realized their error, crossed out only the 0, and then carefully wrote a "1" over it. In this case, temporal segmentation will fail to connect the fact that the first "1" drawn and the second "1" drawn, one at the very start and one at the very end, should belong to the same numeral. While this error may later be repaired by split digit recombination, cases like this in which numerals are drawn in a complex order will remain difficult to solve.

4.3.3 Spatial Segmentation

The other method of regrouping strokes in a single numeral symbol is by using strokes' positions, specifically by determining which strokes intersect and separating groups of strokes that do not intersect each other. The third row of fig. 4.2.6.1 above provided an illustration of how one symbol can be spatially segmented; the figure below shows another example.



[Figure 4.3.3.1 A symbol with multiple numerals present, and the sub-symbols that spatial segmentation produces.]

Advantages and Disadvantages

In the figure above, the two "2"s overlap each other but do not intersect the "3", so they are placed in one subgroup and the "3" another. This demonstrates one weakness of spatial segmentation; it cannot separate a numeral from one that overwrites it, even if they are clearly distinct numerals. Another weakness is that numerals that naturally contain multiple digits that do not overlap, such as "11" and "12", will be split into multiple subgroups by spatial segmentation. Due to these two weaknesses and the fact that temporal segmentation is much more effective at dividing up joined/overwritten numerals with fewer false positives (see figure 4.2.6.1), the system generally does not use spatial segmentation. In future, it may be revisited for more specific uses, but for now this section is included as general information.

4.3.4 Split-Numeral Recombination

Detecting and repairing split numerals is carried out after all other error correction stages, especially temporal and spatial segmentation, have been completed. This is so that false positives incorrectly

segmented by the previous steps may also be caught and repaired, along with any other numerals split by the basic classifier.

The fundamental basis for deciding which two numeral symbols should be combined is trial and error. Aside from a method focused on joining specific examples, such as finding symbols that resemble "1" and "2" and combining them to create "12"s, there is no real way to *theorize* whether two symbols would be more correct as a single symbol, except by combining them and checking whether the result looks better than its individual components. Therefore, my system performs the following steps over the full set of numeral symbols, not including symbols already marked as crossed-out numerals, to check for symbols that should be combined:

1. Order all symbols by their angular position around the center of the clock face.
2. For each two angularly adjacent numeral symbols: if either contains more than one stroke, move on to the next pair. Otherwise, If both contain a single stroke each, use the Ouyang recognizer to assign a numerical label and a match distance to each.
3. Convert the two match distances into z-scores. If either of these z-scores are less than 0, it is raised to 0; this allows split-digits such as "11" and "12", whose parts may look a lot like numerals and thus have low z-scores, to still have a possibility of being repaired.
4. Average the two starting symbols' z-scores.
5. Combine the two starting symbols into a single symbol and determine its closest numeral label and match-distance. Also convert this distance into a z-score.
6. If the z-score of the combined symbol is both lower than the average z-score of the starting symbols, and lower than a strict messiness threshold (3.0), then the combined symbol can be accepted as a reasonable numeral. Otherwise, the two starting symbols are restored.

Advantages and Disadvantages

This process is generally conservative. It will not attempt to combine symbols with more than one stroke, and requires that the combined symbol looks more like a numeral than its constituent parts. This greatly lowers its risk of erroneously grouping two unrelated symbols together, but also lowers its error repair rate for numerals such as "12" or "11," whose constituent parts tend to look a lot like the numerals "1" and "2." Improving this repair process in the future will likely involve removing some of its current restrictions so as to handle more cases, while using a more robust numeral-recognition strategy to avoid falsely grouping two symbols together that should not be.

4.4 Results

To see the results of error correction, which mostly acts on numeral symbols, we compare the numbers and types of numeral symbols produced by the basic classifier to the numeral symbols produced by error correction. The below tables of results illustrates the shift in the frequency of various types of errors, including split, joined, and mislabeled numerals, as well as symbols that contain no numeral strokes at all.

All VIN-96 Numeral Symbols	Basic Classifier	With Error Correction
Correctly labeled numerals	2117	2226
Split numerals	86	60
Joined/overwritten numerals	78	19
Non-numerals (hands, noise)	29	67
Mislabeled numerals	15	14

[Figure 4.4.1 Comparison of the number of various types of numeral symbols in the VIN-96 data set, and how error correction changed their distributions.]

The data above shows that error correction on the VIN-96 clinical set is effective; it results in a 5.1% increase in the number of correctly labeled numerals, a 76% decrease in the number of joined or overwritten numerals, and a 30% decrease in split numerals. Another interesting result is the dramatic increase in the number of symbols containing only non-numeral strokes, but this can be attributed to the system working as intended. Because the joined-numerals category included symbols that combined both numerals and hands, when those symbols were split up, the hands were placed into their own symbols and considered non-numeral, thereby boosting the number of non-numeral symbols.

For comparison on another unrelated clinical data set, I include as well the results of the EGE/ORU-112 set before and after error correction.

EGE/ORU-112 Numeral Symbols	Basic Classifier	With Error Correction
Correctly labeled numerals	2405	2521
Split numerals	66	67
Joined/overwritten numerals	64	11
Non-numerals (hands, noise)	87	166
Mislabeled numerals	26	32

[Figure 4.4.2 Comparison of the number of various types of numeral symbols in the EGE/ORU-112 data set, and how they changed from the basic classifier to after error correction.]

The EGE/ORU-112 data set contains more examples of rare strokes, such as tick marks and spokes, and as such has far more non-numeral strokes to account for. However, the general trend – a 4.8% increase in correctly labeled numerals and an 82% decrease in joined or overwritten numerals – holds here as well. The overall number of split numerals did not seem to change in this set of clocks, suggesting that either the split-numeral repair strategy is too conservative, or needs to be adapted to more diverse instances of split numerals.

But in general, these results indicate the effectiveness of the current error repair strategy in correctly separating strokes of different types into different symbols, and in labeling numerals. The challenge that remains is to expand the set of symbols the system is capable of labeling, this time paying especially close attention to the different types of non-numeral symbols, such as hands, noise, and tick marks.

5. System Performance and Final Results

With error correction implemented, I now provide a breakdown of the final system's performance on multiple sets of clock drawings, each representing a slightly different challenge. The performance information below includes the percentage of strokes in each category that were correctly labeled, the number of strokes overall, and the number of clocks that contained strokes from each category.

YDU-100 Healthy Clocks

The YDU-100 set was selected to contain relatively well-drawn clock faces from healthy individuals, each of which contains all 12 numerals and hands. This allows us to use it as a gauge of the system's basic performance.

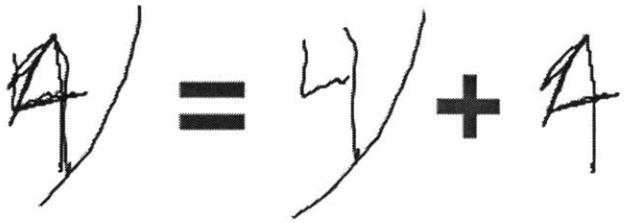
YDU-100 Healthy Set	Accuracy	# Strokes	# Containing Clocks
All strokes	94.8%	4766	200
Numeral strokes	98.1%	3436	200
Hand strokes	91.6%	963	200
Crossed-out numerals	28.0%	25	15
Noise strokes	80.0%	65	44
Tick marks	0.0%	43	4
Clock outline strokes	99.0%	205	199
Other	0.0%	29	12

[Figure 5.1 Breakdown of classification accuracy for each of 8 categories of strokes, for clock drawings in the YDU-100 data set.]

Although my system currently lacks a method of detecting and labeling tick marks, which lowers its accuracy here, the overall performance (94.8%) on this set of clock drawings is good. In particular, the clock outline, numeral strokes, and hand strokes have consistently high performance (over 90%), showing that the system's assumptions about these types of components during the classification process were reasonable.

Of the categories of strokes that are labeled, the one with poorest performance is that of the crossed-out numerals. Here, I make an unintuitive observation; when healthier patients overwrite numerals, they tend to make the result look clean and recognizable as a numeral. But a symbol that does not look messy will

not be detected as problematic and will not be split by error correction, leading to more crossed-out numerals being ignored in healthy clocks. The example below of a "4" overwritten with another "4" shows how this might be the case; the resulting overwritten numeral is too recognizable as a "4" to be detected as a potential error.



[Figure 5.1 An overwritten "4" made up of two "4"s, with the second drawn 40 seconds after the first. The first should be labeled as a crossed-out numeral; however, the combined symbol resembles an ordinary "4" and so the overwriting is not detected.]

Considerations for how to better detect and repair crossed-out digits like these should be the focus of future work on this classifier.

VIN-96 Clinical Clocks

I use the VIN-96 clinical set, a set of randomly-selected clock drawings with a relatively small number of rare strokes, as a sample of the ordinary input to the system. Much of the current system's experimental variables were determined by testing using this data set, so it does not necessarily provide unbiased information; however, its final results are still worth describing below.

VIN-96 Clinical Set	Accuracy	# Strokes	# Containing Clocks
All strokes	91.2%	4771	192
Numeral strokes	95.8%	3346	192
Hand strokes	83.5%	1044	192
Crossed-out numerals	36.8%	19	13
Noise strokes	74.2%	89	53
Tick marks	0.0%	1	1
Clock outline strokes	87.8%	230	192
Other	0.0%	41	12

[Figure 5.2 Breakdown of classification accuracy for each of 8 categories of strokes, for clock drawings in the VIN-96 data set.]

In this set of clock drawings, while the accuracy of numeral labeling remains above 95%, the accuracy of labeling hand strokes and clock outlines begins to drop as users introduce more variation into them. This

can be seen by the higher number of clock outline strokes than actual clocks, suggesting that certain clock outlines were drawn with multiple strokes rather than a single long stroke, which is a case that the system does not account for currently. The accuracy of detecting noise strokes begins to decrease as well, likely due to the increased messiness of these clock drawings, which gives rise to different types of noise.

Of note is that the accuracy rate of crossed-out numerals, while still low, has increased above that of the previous "healthy" clocks' accuracy. This can be explained by the increased presence of scribble-like strokes in this set of clocks, which are easier to detect and recognize as crossing out another stroke.

EGE/ORU-112 Clinical Clocks

The EGE/ORU-112 clock drawings, another set randomly drawn from clinical data, contains many more uncommon strokes such as spokes and text. It is a more challenging set to classify correctly than the VIN-96 set as a result.

EGE/ORU-112 Clinical	Accuracy	# Strokes	# Containing Clocks
All strokes	88.6%	5861	224
Numeral strokes	95.1%	3810	222
Hand strokes	85.7%	1173	224
Crossed-out numerals	58.3%	72	20
Noise strokes	80.8%	349	104
Tick marks	0.0%	62	5
Clock outline strokes	85.5%	275	221
Other	0.058%	120	120

[Figure 5.3 Breakdown of classification accuracy for each of 8 categories of strokes, for clock drawings in the EGE/ORU-112 data set.]

Compared to the VIN-96 clocks, the accuracy of this set is lower in general, due to the much higher numbers of tick marks and "other" strokes present in clock drawings. However, despite these unhandled strokes, classification accuracy remained stable or even increased for the other basic categories of strokes within the set, including hands and crossed-out numerals. This demonstrates that the presence of secondary strokes does not strongly impede correct classification of basic clock components.

Having looked at three different sets of data, each with its own complexities, we can conclude that the ClockSketch classifier is robust enough to accurately label all of the basic components of a clock drawing (clock outlines, numerals, hands, noise), even when operating on un-selected and messy clinical data as in the EGE/ORU-112 set. But as the EGE/ORU-112 set also shows, this classifier can't simply ignore all clock components outside of the basics and still achieve high performance on clinical clock drawings. In the future, our system must be able to handle these secondary clock components.

6. Conclusion and Future Direction

I have described and implemented the new ClockSketch classifier, a robust stroke classification system that accurately groups and labels the basic components of a clock drawing, including numerals, hands, the clock outline, and noise. It is able to divide a set of undifferentiated strokes into meaningful and labeled symbols, even in the presence of messy input data or secondary/extraneous strokes. This ability is derived from a human-logic-centric view of the structure of the input, as well as a willingness to use and combine multiple dimensions of information, especially spatiotemporal information, in diverse ways. This system will both allow clinicians to analyze clock drawings "in the field" more quickly and automatically, and provide useful theoretical information about how clock drawings are made.

As the ClockSketch classifier has provided a firm framework for clock drawing classification, future work should build upon this framework both to improve the classification of basic clock components and to add the capability to label more diverse types of components. I describe below some of the additions that could be made, along with the possible difficulties encountered in making them.

Tick Marks

As seen by the per-data-set results in section 5, tick marks are present in a very small percentage of clock faces, but when they do appear there are usually a large number of them, as they tend to correspond to the numerals in a clock face. Theoretically, deciding whether a stroke is a tick mark is relatively simple; they are often linear strokes drawn intersecting or close to the clock outline, with angles oriented towards the center of the clock face. Both of these pieces of information are easily calculated.

However, because tick marks are so close to numerals, the possibility of falsely marking a legitimate numeral stroke as a tick mark is a primary concern, which is exacerbated by the rarity of tick marks in the majority of clock faces. In light of this, I believe tick mark detection should only be performed after numeral segmentation and even after error correction has divided up joined symbols. At that point, most

strokes that belong together have been segmented into their own individual groups, and it becomes simpler to determine which group may correspond to a tick mark (a group containing only a single stroke, linear, pointing towards the center, intersecting the clock outline).

Improving Hand-Labeling Performance

Among the basic clock components, strokes representing hands and center dots are currently the least accurately labeled, for two reasons. The first is due to the natural difficulty of distinguishing between hour hands, minute hands, and center dots in the hand-labeling process. The second is that a considerable number of hand strokes, particularly arrowheads, are being grouped with numerals during hand-numeral separation. This is supported by figures 4.4.1 and 4.4.2, which provide a breakdown of the actual contents of numeral-labeled symbols, and which show that many symbols actually contain strokes in the "non-numeral" category, which includes noise and hands.

To improve the classifier's accuracy in hand labeling, a way of detecting hand strokes hidden among numeral symbols will be essential, but it must not be overzealous in declaring true numeral symbols to be hands. Using recognition to determine how much a symbol looks like a numeral will likely be useful for avoiding false positives. The time a symbol was drawn may also be telling, as users often draw a full set of numerals before drawing the hands, but more experimentation is needed to determine whether a "numeral-hand time gap" can be consistently found for every clock face.

7. Acknowledgements

I'd like to thank Professor Randall Davis for his patient guidance and direction these past two years. I have learned a great deal from him about professionalism and dedication as well as academic research, without which I could not have succeeded, and which I look forward to applying in future endeavors.

I'd also like to thank Tom Ouyang for providing experience and tips for the problem of symbol recognition, and Yale Song for helping research alternate directions the ClockSketch classifier could have proceeded along.

Finally, I want to thank the rest of my research group, present and former: Ying Yin, Andrew Correa, Jeremy Scott , William Souillard-Mandar, and Danica Chang, for support and encouragement throughout my studies.

This work was supported in part by contract D13AP00008 from the Defense Advanced Research Projects Agency.

8. References

[1] Shulman, K. I. (2000), Clock-drawing: is it the ideal cognitive screening test?. Int. J. Geriat. Psychiatry, 15: 548–561. doi: 10.1002/1099-1166(200006)15:6<548::AID-GPS242>3.0.CO;2-U

[2] Ouyang, Tom Y., and Randall Davis. "A Visual Approach to Sketched Symbol Recognition." *IJCAI*. Vol. 9. 2009.

9. Appendix

A. Clock Drawings in Training and Testing Sets

A.1 YDU-51 Healthy Training Set

YDU0015872230-2012-04-13-09-22-13ScoredV3.7.csk
YDU0024943741-2012-04-12-11-58-18ScoredV3.7.csk
YDU0029629324-2012-04-26-14-55-31ScoredV3.8.csk
YDU0030990860-2012-07-02-15-24-30ScoredV4.0.csk
YDU0053889897-2012-06-08-10-14-47ScoredV3.8.csk
YDU0057362444-2012-12-20-07-56-35ScoredV4.1.csk
YDU0064762880-2012-12-20-12-05-22ScoredV4.1.csk
YDU0065892741-2011-10-13-09-25-34ScoredV4.0.csk
YDU0089151951-2012-11-28-10-39-20ScoredV4.1.csk
YDU0092317699-2013-03-08-09-42-13ScoredV4.4.csk
YDU0093793509-2012-11-27-15-05-48ScoredV4.4.csk
YDU0112592176-2012-07-31-16-43-38ScoredV4.0.csk
YDU0123041202-2011-11-15-09-21-19ScoredV3.8.csk
YDU0129091497-2011-12-19-18-51-20ScoredV3.8.csk
YDU0144341018-2012-04-24-15-12-09ScoredV3.8.csk
YDU0160769184-2012-05-09-12-21-12ScoredV3.8.csk
YDU0169902394-2011-11-08-17-54-10ScoredV3.8.csk
YDU0181639157-2012-11-16-09-34-50ScoredV4.1.csk
YDU0187125954-2013-05-23-10-36-18ScoredV4.4.csk
YDU0189863547-2012-06-13-08-04-00ScoredV3.8.csk
YDU0191480696-2013-04-11-10-09-41ScoredV4.4.csk
YDU0196742056-2012-04-27-12-43-28ScoredV4.0.csk
YDU0211097573-2011-11-28-18-48-15ScoredV4.0.csk
YDU0213031281-2013-02-28-08-05-50ScoredV4.4.csk
YDU0236027528-2011-10-17-18-39-30ScoredV3.7.csk
YDU0240350443-2012-07-12-08-52-55ScoredV4.0.csk
YDU0247161988-2012-07-23-10-55-23ScoredV4.0.csk
YDU0252721858-2011-11-01-15-16-12ScoredV3.7.csk
YDU0257649605-2012-09-24-13-21-16ScoredV4.1.csk
YDU0257793879-2012-02-17-10-00-09ScoredV3.7.csk
YDU0262424844-2012-12-04-09-43-24ScoredV4.1.csk
YDU0283089966-2012-11-13-08-23-52ScoredV4.1.csk
YDU0283670083-2012-11-29-10-12-58ScoredV4.1.csk
YDU0283970673-2012-06-11-15-21-01ScoredV4.0.csk
YDU0286312782-2013-01-18-10-17-23ScoredV4.4.csk
YDU0295441515-2012-07-17-14-54-08ScoredV4.0.csk
YDU0305577176-2012-08-17-10-39-04ScoredV4.1.csk
YDU0308807254-2012-11-08-13-06-00ScoredV4.1.csk
YDU0310771230-2011-11-14-18-10-16ScoredV3.8.csk
YDU0312745603-2012-06-11-19-10-54ScoredV3.8.csk
YDU0320124394-2013-04-29-11-04-18ScoredV4.4.csk
YDU0323716285-2012-10-15-08-14-30ScoredV4.1.csk
YDU0331939947-2012-09-12-08-15-15ScoredV4.1.csk

YDU0342530934-2011-10-13-11-48-27ScoredV3.7.csk
YDU0354521219-2011-10-07-09-38-42ScoredV3.7.csk
YDU0371045035-2012-02-13-15-17-44ScoredV3.7.csk
YDU0372786121-2013-02-11-09-35-09ScoredV4.4.csk
YDU0375115358-2012-07-17-14-37-32ScoredV4.0.csk
YDU0395245091-2012-10-03-13-33-44ScoredV4.1.csk
YDU0408173575-2012-10-23-07-47-51ScoredV4.1.csk
YDU0410561867-2012-01-19-09-24-47ScoredV3.7.csk

A.2 YDU-100 Healthy Testing Set

YDU1458697332-2013-04-26-11-09-13ScoredV4.4.csk
YDU1459723324-2011-12-02-08-16-44ScoredV3.8.csk
YDU1460920966-2011-11-18-07-47-33ScoredV3.8.csk
YDU1469606748-2012-12-11-10-08-26ScoredV4.1.csk
YDU1473040393-2012-04-12-10-33-21ScoredV3.7.csk
YDU1473494971-2011-12-06-15-30-15ScoredV3.8.csk
YDU1476838616-2012-08-21-08-26-06ScoredV4.1.csk
YDU1481304857-2012-01-25-11-24-17ScoredV3.7.csk
YDU1482261301-2012-07-10-09-36-47ScoredV4.0.csk
YDU1483910122-2012-03-08-10-38-01ScoredV3.7.csk
YDU1487544303-2012-09-17-15-44-06ScoredV4.1.csk
YDU1493344152-2012-12-17-11-51-55ScoredV4.1.csk
YDU1494051224-2011-12-06-19-18-57ScoredV3.8.csk
YDU1494484873-2012-05-07-13-12-19ScoredV3.8.csk
YDU1503656639-2013-01-29-10-14-25ScoredV4.4.csk
YDU1515921483-2013-01-10-10-41-40ScoredV4.4.csk
YDU1522045959-2013-05-23-09-15-56ScoredV4.4.csk
YDU1522657394-2012-03-23-11-48-11ScoredV3.7.csk
YDU1523118076-2011-11-22-17-18-56ScoredV3.7.csk
YDU1524067713-2013-02-15-10-01-03ScoredV4.4.csk
YDU1536706745-2012-01-23-15-24-32ScoredV3.7.csk
YDU1546193696-2012-02-27-16-37-29ScoredV3.7.csk
YDU1551141438-2012-08-16-10-27-41ScoredV4.1.csk
YDU1552370799-2013-03-01-10-04-20ScoredV4.4.csk
YDU1558755772-2012-04-02-17-21-13ScoredV3.7.csk
YDU1589894957-2012-07-23-11-24-59ScoredV4.0.csk
YDU1596983648-2012-07-31-09-26-01ScoredV4.0.csk
YDU1608374391-2012-08-22-09-19-22ScoredV4.1.csk
YDU1622808830-2012-01-31-08-21-08ScoredV3.7.csk
YDU1625107784-2012-06-07-09-46-51ScoredV3.8.csk
YDU1637367575-2012-07-13-08-46-50ScoredV4.0.csk
YDU1640468945-2012-02-24-09-36-36ScoredV3.7.csk
YDU1663829053-2012-02-21-17-58-58ScoredV3.7.csk
YDU1664510618-2012-02-23-08-40-34ScoredV4.0.csk
YDU1698831024-2012-02-21-09-41-37ScoredV3.7.csk
YDU1700123170-2012-07-26-08-10-07ScoredV4.0.csk
YDU1726209825-2011-11-08-16-03-35ScoredV3.8.csk
YDU1728485575-2011-11-22-08-32-13ScoredV3.7.csk
YDU1731592614-2013-05-24-12-46-50ScoredV4.4.csk

YDU1737670214-2013-04-08-09-22-25ScoredV4.4.csk
YDU1737982982-2012-06-22-12-55-42ScoredV4.4.csk
YDU1758879357-2011-10-03-18-06-51ScoredV3.7.csk
YDU1758978507-2012-07-05-10-16-53ScoredV4.0.csk
YDU1766128431-2012-07-12-09-10-32ScoredV4.0.csk
YDU1766227533-2012-12-18-09-58-19ScoredV4.1.csk
YDU1771628421-2012-04-24-19-17-14ScoredV3.8.csk
YDU1780078212-2013-01-10-10-13-15ScoredV4.4.csk
YDU1793443167-2012-10-02-10-08-58ScoredV4.1.csk
YDU1796066114-2012-02-10-10-14-59ScoredV4.0.csk
YDU1798309932-2012-04-20-10-39-25ScoredV3.8.csk
YDU1803059071-2011-11-08-09-16-21ScoredV3.8.csk
YDU1824770682-2012-04-10-07-54-02ScoredV3.8.csk
YDU1826778250-2012-12-18-15-37-16ScoredV4.1.csk
YDU1830830601-2012-06-26-17-22-37ScoredV3.8.csk
YDU1839401766-2011-11-22-10-14-37ScoredV3.7.csk
YDU1839553265-2012-07-02-13-10-27ScoredV4.0.csk
YDU1839604771-2011-12-09-10-30-08ScoredV3.8.csk
YDU1857163369-2012-03-13-09-39-37ScoredV4.0.csk
YDU1870990911-2012-01-17-07-57-45ScoredV3.7.csk
YDU1878818512-2011-11-15-16-02-23ScoredV4.0.csk
YDU1886670830-2012-11-16-08-08-23ScoredV4.1.csk
YDU1891448893-2012-04-03-17-46-56ScoredV4.0.csk
YDU1903803704-2012-09-14-11-16-58ScoredV4.1.csk
YDU1905167133-2013-04-18-10-19-48ScoredV4.4.csk
YDU1915587439-2012-11-09-10-35-42ScoredV4.1.csk
YDU1921640868-2012-03-15-09-21-53ScoredV3.7.csk
YDU1922832223-2013-04-23-08-10-45ScoredV4.4.csk
YDU1950620409-2011-10-27-09-30-57ScoredV3.7.csk
YDU1953159052-2011-12-13-10-02-21ScoredV3.8.csk
YDU1956444688-2011-12-13-08-06-06ScoredV3.8.csk
YDU1965420037-2011-10-18-12-57-16ScoredV3.7.csk
YDU1966488010-2011-10-17-17-06-12ScoredV4.4.csk
YDU1966654365-2012-02-02-09-44-24ScoredV3.7.csk
YDU1980877338-2012-06-21-10-42-42ScoredV3.8.csk
YDU1980935602-2012-02-09-10-18-13ScoredV3.7.csk
YDU1980985828-2012-08-31-07-34-34ScoredV4.1.csk
YDU2005771610-2012-09-18-09-31-10ScoredV4.1.csk
YDU2010621421-2012-04-18-10-02-57ScoredV3.8.csk
YDU2012172477-2012-02-06-18-45-26ScoredV3.7.csk
YDU2015396656-2012-06-11-10-56-48ScoredV3.8.csk
YDU2018030890-2012-10-19-08-25-27ScoredV4.1.csk
YDU2032502788-2012-02-13-17-10-52ScoredV3.7.csk
YDU2045447907-2012-11-05-17-59-55ScoredV4.1.csk
YDU2060021469-2012-05-23-10-53-28ScoredV3.8.csk
YDU2061476618-2011-10-14-08-58-36ScoredV3.7.csk
YDU2061880442-2012-01-27-08-27-04ScoredV3.7.csk
YDU2071144394-2012-10-31-09-19-46ScoredV4.1.csk
YDU2075451915-2012-12-18-09-17-12ScoredV4.1.csk

YDU2076742150-2011-10-11-14-57-26ScoredV3.7.csk
YDU2084403185-2012-02-06-17-30-39ScoredV3.7.csk
YDU2090469603-2013-03-22-08-45-26ScoredV4.4.csk
YDU2092354574-2013-05-31-11-14-08ScoredV4.4.csk
YDU2092607378-2012-03-14-10-24-53ScoredV3.7.csk
YDU2100188187-2012-06-26-09-47-45ScoredV4.4.csk
YDU2100492527-2013-04-12-09-27-06ScoredV4.4.csk
YDU2134735600-2012-02-09-09-54-31ScoredV3.7.csk
YDU2139094310-2012-12-13-07-51-40ScoredV4.1.csk
YDU2140910798-2013-03-28-11-23-55ScoredV4.4.csk
YDU2141877161-2012-02-28-15-45-54ScoredV4.0.csk
YDU2144586219-2012-11-19-15-05-43ScoredV4.1.csk

A.3 VIN-96 Clinical Test Set

VIN0063271152-2011-05-13-10-31-30ScoredV3.6.csk
VIN0078085048-2009-05-05-13-30-03ScoredV3.6.csk
VIN0091872821-2007-07-30-16-10-59ScoredV3.0.csk
VIN0098771936-2011-11-22-12-04-09ScoredV3.6.csk
VIN0102603075-2011-08-03-10-08-28ScoredV3.6.csk
VIN0150986070-2011-11-10-15-03-08ScoredV3.6.csk
VIN0176770694-2006-02-20-13-19-09ScoredV3.0.csk
VIN0212724106-2006-04-05-11-46-30ScoredV3.0.csk
VIN0216801333-2011-02-25-16-32-24ScoredV3.6.csk
VIN0250487960-2009-04-22-10-05-59ScoredV3.0.csk
VIN0292653489-2009-08-21-12-37-33ScoredV3.0.csk
VIN0320646684-2006-03-13-13-40-53ScoredV3.0.csk
VIN0360962039-2009-07-08-14-40-12ScoredV3.0.csk
VIN0398883252-2007-03-19-09-45-19ScoredV3.0.csk
VIN0408418879-2011-03-30-11-48-21ScoredV3.6.csk
VIN0441352371-2007-04-25-09-55-48ScoredV3.0.csk
VIN0487443598-2009-12-09-13-13-59ScoredV3.0.csk
VIN0500200077-2011-03-04-16-28-30ScoredV3.6.csk
VIN0516652623-2011-11-15-08-44-46ScoredV3.6.csk
VIN0525310967-2009-02-18-09-47-22ScoredV3.6.csk
VIN0536748850-2009-08-28-11-18-54ScoredV3.0.csk
VIN0542469514-2011-02-18-16-37-10ScoredV3.6.csk
VIN0588654376-2009-07-22-13-06-22ScoredV3.0.csk
VIN0624107039-2011-07-25-13-17-30ScoredV3.6.csk
VIN0676446680-2006-03-22-09-56-04ScoredV3.0.csk
VIN0711833317-2011-11-14-10-52-39ScoredV3.6.csk
VIN0762360910-2007-04-11-14-40-56ScoredV3.0.csk
VIN0775888172-2006-07-19-10-06-33ScoredV3.0.csk
VIN0828812527-2007-06-25-10-52-45ScoredV3.0.csk
VIN0829757788-2009-10-29-10-28-11ScoredV3.0.csk
VIN0833140908-2011-07-25-09-05-47ScoredV3.6.csk
VIN0842680555-2007-03-27-09-11-57ScoredV3.0.csk
VIN0847029653-2007-01-23-14-22-06ScoredV3.0.csk
VIN0847870313-2007-06-14-15-49-26ScoredV3.0.csk
VIN0873509087-2007-04-12-10-37-10ScoredV3.6.csk

VIN0880699204-2011-12-21-11-26-50ScoredV3.6.csk
VIN0902133667-2006-12-14-10-15-57ScoredV3.0.csk
VIN0903667079-2009-10-21-09-16-28ScoredV3.0.csk
VIN0929937545-2011-11-14-13-24-56ScoredV3.6.csk
VIN0934868080-2011-07-19-09-54-30ScoredV3.6.csk
VIN0969671570-2011-07-27-15-32-26ScoredV3.6.csk
VIN0983535487-2009-12-02-11-21-16ScoredV3.0.csk
VIN1019479827-2007-03-12-09-32-31ScoredV3.0.csk
VIN1021133203-2006-06-21-09-56-44ScoredV3.0.csk
VIN1030668968-2009-02-23-09-39-31ScoredV3.6.csk
VIN1052780286-2007-05-22-13-18-35ScoredV3.0.csk
VIN1082466239-2006-04-18-10-02-09ScoredV3.0.csk
VIN1106145152-2006-05-10-13-15-06ScoredV3.0.csk
VIN1152904937-2009-03-16-10-18-38ScoredV3.0.csk
VIN1158639961-2009-02-09-10-22-03ScoredV3.0.csk
VIN1181803228-2011-10-11-15-41-25ScoredV3.6.csk
VIN1185197344-2006-03-20-14-01-00ScoredV3.6.csk
VIN1202464077-2011-12-16-12-51-14ScoredV3.6.csk
VIN1245859164-2011-11-26-13-10-18ScoredV3.6.csk
VIN1250546032-2009-12-15-12-09-06ScoredV3.0.csk
VIN1263579883-2007-01-04-09-26-53ScoredV3.0.csk
VIN1270987934-2010-07-26-13-29-18ScoredV3.0.csk
VIN1290845432-2011-04-01-13-50-59ScoredV3.6.csk
VIN1303699403-2006-11-15-09-12-48ScoredV3.0.csk
VIN1313750347-2006-05-04-16-19-12ScoredV3.0.csk
VIN1335376767-2011-11-16-15-15-23ScoredV3.6.csk
VIN1341672470-2007-01-11-13-42-38ScoredV3.0.csk
VIN1355430577-2006-03-06-14-25-58ScoredV3.0.csk
VIN1381455645-2011-06-10-08-41-19ScoredV3.6.csk
VIN1475529716-2009-07-07-09-55-26ScoredV3.0.csk
VIN1487156944-2007-07-17-13-35-36ScoredV3.0.csk
VIN1520402976-2007-03-05-13-32-22ScoredV3.0.csk
VIN1542794379-2007-04-10-13-31-20ScoredV3.0.csk
VIN1582668235-2007-01-02-09-33-30ScoredV3.0.csk
VIN1638973340-2006-05-24-10-45-45ScoredV3.0.csk
VIN1660017801-2009-03-17-15-35-05ScoredV3.0.csk
VIN1687212776-2006-04-20-11-13-56ScoredV3.0.csk
VIN1699673802-2009-11-06-14-37-36ScoredV3.0.csk
VIN1715687822-2007-07-19-11-09-04ScoredV3.0.csk
VIN1724539676-2006-02-21-14-45-09ScoredV3.0.csk
VIN1724927675-2009-09-29-14-56-48ScoredV3.0.csk
VIN1728476201-2006-05-30-10-23-30ScoredV3.0.csk
VIN1737987862-2011-11-16-12-59-56ScoredV3.6.csk
VIN1745613379-2007-05-31-15-43-49ScoredV3.0.csk
VIN1786441936-2009-12-08-12-12-26ScoredV3.0.csk
VIN1792676811-2007-01-18-11-31-56ScoredV3.0.csk
VIN1842240487-2006-03-27-16-17-04ScoredV3.0.csk
VIN1851737753-2007-06-27-09-39-27ScoredV3.6.csk
VIN1926831987-2011-10-24-08-52-21ScoredV3.6.csk

VIN1931807909-2009-11-06-11-29-55ScoredV3.0.csk
VIN1973134116-2009-04-29-16-12-05ScoredV3.0.csk
VIN1989294807-2009-12-04-15-30-48ScoredV3.0.csk
VIN1996974147-2011-03-17-18-43-37ScoredV3.6.csk
VIN1998384886-2006-03-06-15-43-42ScoredV3.0.csk
VIN2015542200-2011-02-25-12-03-29ScoredV3.6.csk
VIN2016358114-2009-09-16-14-38-42ScoredV3.6.csk
VIN2099133980-2006-11-16-13-46-07ScoredV3.0.csk
VIN2100335296-2009-04-21-10-11-50ScoredV3.0.csk
VIN2107627400-2006-03-22-16-34-03ScoredV3.0.csk
VIN2109969259-2009-10-19-13-35-21ScoredV3.0.csk
VIN2123733155-2011-04-07-11-30-15ScoredV3.6.csk

A.4 EGE/ORU-112 Clinical Test Set

EGE0048723202-2010-06-21-12-57-03ScoredV3.0.csk
EGE0056676390-2010-07-14-09-25-41ScoredV3.0.csk
EGE0091443850-2010-05-21-09-22-22ScoredV3.0.csk
EGE0092099221-2010-05-26-12-56-40ScoredV3.0.csk
EGE0143851472-2010-07-20-10-14-13ScoredV3.0.csk
EGE0167599527-2010-07-12-12-57-41ScoredV3.0.csk
EGE0270005059-2010-06-18-13-15-38ScoredV3.5.csk
EGE0271665761-2010-07-06-14-23-16ScoredV3.0.csk
EGE0348579752-2010-05-13-08-59-58ScoredV3.0.csk
EGE0375558151-2010-07-13-09-29-48ScoredV3.0.csk
EGE0430106867-2010-01-22-13-22-42ScoredV3.5.csk
EGE0458758770-2010-06-01-09-30-50ScoredV3.0.csk
EGE0468185867-2009-05-13-10-43-23ScoredV3.0.csk
EGE0477008340-2010-06-10-09-35-01ScoredV3.0.csk
EGE0477340547-2009-04-23-11-04-36ScoredV3.0.csk
EGE0488148287-2010-08-09-09-19-33ScoredV3.0.csk
EGE0489394608-2010-07-14-14-09-05ScoredV3.9.csk
EGE0495314770-2010-03-10-13-26-24ScoredV3.0.csk
EGE0528412617-2010-04-13-14-22-10ScoredV3.9.csk
EGE0556887562-2010-08-31-09-35-00ScoredV3.0.csk
EGE0603076589-2010-05-27-13-13-45ScoredV3.0.csk
EGE0628057102-2010-03-31-13-33-01ScoredV3.0.csk
EGE0706902923-2010-07-09-13-09-46ScoredV3.0.csk
EGE0719336199-2010-04-05-09-22-44ScoredV3.0.csk
EGE0720764651-2010-05-13-14-25-42ScoredV3.0.csk
EGE0744818560-2010-06-30-14-07-11ScoredV3.0.csk
EGE0766567854-2010-07-30-13-44-54ScoredV3.0.csk
EGE0813752608-2010-04-21-14-17-02ScoredV3.0.csk
EGE0838358279-2010-06-11-09-28-04ScoredV3.9.csk
EGE0882081772-2010-05-27-09-14-35ScoredV3.0.csk
EGE0887135619-2010-03-31-09-31-58ScoredV3.0.csk
EGE0922540745-2010-07-15-13-20-16ScoredV3.0.csk
EGE0960117541-2010-04-19-10-16-16ScoredV3.9.csk
EGE0969601126-2010-06-16-09-20-44ScoredV3.9.csk
EGE1059111135-2010-06-23-09-59-34ScoredV3.0.csk

EGE1077000824-2010-07-20-09-41-05ScoredV3.0.csk
EGE1152023406-2010-04-14-09-24-23ScoredV3.0.csk
EGE1165700469-2010-08-02-13-24-32ScoredV3.5.csk
EGE1168812095-2010-03-30-14-08-48ScoredV3.0.csk
EGE1174719586-2010-07-12-10-09-13ScoredV3.0.csk
EGE1194738423-2010-06-09-13-35-23ScoredV3.0.csk
EGE1199219204-2010-04-28-13-21-55ScoredV3.5.csk
EGE1210175927-2010-04-12-09-47-36ScoredV3.0.csk
EGE1227383573-2009-04-22-04-57-14ScoredV3.0.csk
EGE1229216930-2010-04-26-09-17-58ScoredV3.0.csk
EGE1230811213-2010-07-01-13-08-52ScoredV3.5.csk
EGE1238003390-2010-06-02-12-44-08ScoredV3.0.csk
EGE1240346762-2009-04-20-05-25-08ScoredV3.0.csk
EGE1252519043-2010-06-23-13-15-35ScoredV3.0.csk
EGE1318066790-2009-04-15-05-45-20ScoredV3.0.csk
EGE1331665504-2010-07-08-10-15-42ScoredV3.0.csk
EGE1370827434-2009-04-16-05-31-15ScoredV3.0.csk
EGE1377705409-2010-04-26-13-27-01ScoredV3.0.csk
EGE1418164226-2010-07-28-09-27-05ScoredV3.0.csk
EGE1466972375-2010-04-16-11-25-25ScoredV3.0.csk
EGE1477402218-2010-06-17-10-05-57ScoredV3.0.csk
EGE1516019668-2010-06-14-13-17-14ScoredV3.0.csk
EGE1528314806-2010-07-08-13-19-36ScoredV3.0.csk
EGE1551806340-2010-04-23-13-25-45ScoredV3.0.csk
EGE1555519969-2010-06-16-13-00-56ScoredV3.0.csk
EGE1564463898-2010-06-25-10-23-04ScoredV3.0.csk
EGE1579069590-2010-08-24-14-29-28ScoredV3.0.csk
EGE1595650202-2010-05-19-13-19-21ScoredV3.0.csk
EGE1635278100-2010-07-22-09-39-32ScoredV3.5.csk
EGE1663533090-2010-03-01-09-54-36ScoredV3.0.csk
EGE1703572234-2010-04-20-14-31-03ScoredV3.0.csk
EGE1724348272-2010-08-27-13-27-54ScoredV3.9.csk
EGE1729607156-2010-02-02-14-06-26ScoredV3.0.csk
EGE1731929614-2010-08-10-09-21-32ScoredV3.9.csk
EGE1765900491-2010-06-07-13-13-50ScoredV3.0.csk
EGE1813478853-2010-04-30-10-52-35ScoredV3.5.csk
EGE1865516148-2010-07-13-14-42-27ScoredV3.0.csk
EGE1923413357-2010-07-27-14-19-36ScoredV3.0.csk
EGE1940454417-2010-06-28-13-21-41ScoredV3.0.csk
EGE1962770219-2010-03-17-09-19-15ScoredV3.0.csk
EGE1969144806-2009-04-29-09-27-31ScoredV3.0.csk
EGE2038384853-2010-05-12-12-40-20ScoredV3.0.csk
EGE2089890253-2010-07-16-15-11-01ScoredV3.0.csk
EGE2100566174-2010-06-28-14-19-49ScoredV3.0.csk
ORU0033873510-2010-06-04-09-56-39ScoredV3.0.csk
ORU0141409120-2011-06-27-09-45-21ScoredV3.7.csk
ORU0214002724-2009-11-06-11-01-07ScoredV3.0.csk
ORU0234651310-2010-07-29-10-50-19ScoredV3.0.csk
ORU0260103282-2010-03-01-11-18-51ScoredV3.0.csk

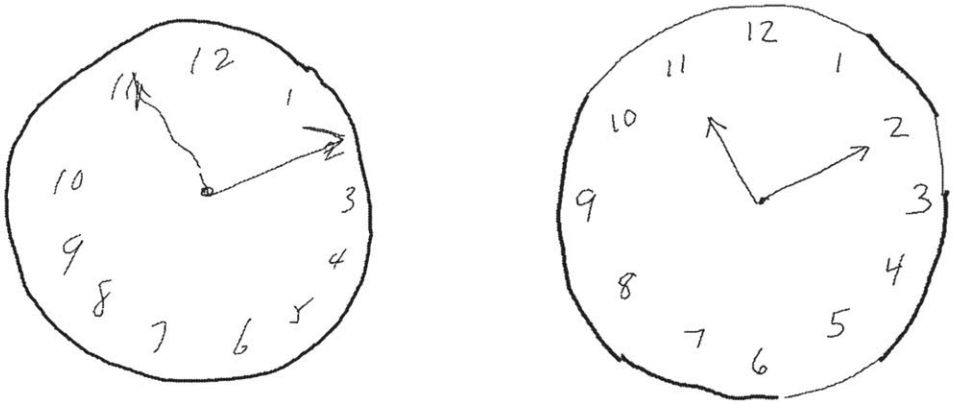
ORU0263782536-2010-08-25-11-10-07ScoredV3.7.csk
ORU0271454768-2009-12-17-14-27-21ScoredV3.0.csk
ORU0278498759-2009-09-15-10-52-37ScoredV3.0.csk
ORU0355087110-2009-06-23-15-22-47ScoredV3.0.csk
ORU0377348240-2011-09-09-11-05-39ScoredV3.7.csk
ORU0435765892-2009-05-11-11-45-40ScoredV3.7.csk
ORU0526107987-2010-02-15-16-27-57ScoredV3.0.csk
ORU0566807646-2011-06-14-15-30-35ScoredV3.0.csk
ORU0577384976-2010-05-20-14-27-09ScoredV3.5.csk
ORU0702838292-2011-06-22-14-58-57ScoredV3.0.csk
ORU1184517177-2009-06-29-10-43-39ScoredV3.6.csk
ORU1298744135-2009-05-27-11-17-58ScoredV3.7.csk
ORU1504064473-2010-07-08-14-59-54ScoredV3.0.csk
ORU1513723722-2010-03-02-10-56-57ScoredV3.0.csk
ORU1523209376-2010-09-23-10-39-28ScoredV3.0.csk
ORU1545747048-2009-05-19-16-26-10ScoredV3.7.csk
ORU1572369380-2009-05-13-11-09-35ScoredV3.7.csk
ORU1579143314-2010-05-27-12-06-45ScoredV3.0.csk
ORU1594504278-2010-04-20-15-29-54ScoredV3.0.csk
ORU1614808862-2009-07-15-09-44-15ScoredV3.0.csk
ORU1823106156-2009-11-17-11-01-55ScoredV3.0.csk
ORU1852069863-2009-09-29-14-27-11ScoredV3.0.csk
ORU1952963752-2011-10-04-10-49-41ScoredV3.0.csk
ORU1967217631-2009-04-07-11-48-39ScoredV3.7.csk
ORU1979847017-2010-11-02-13-57-42ScoredV3.0.csk
ORU2120290419-2009-10-13-15-02-54ScoredV3.0.csk
ORU2123368665-2009-10-26-10-52-34ScoredV3.0.csk
ORU2124971720-2009-04-24-11-07-33ScoredV3.6.csk

EMD-20 Clinical Test Set

TCD1V3.5CIN2109964091-2010-10-15-15-31-17ScoredV3.0.csk
TCD2V3.5CIN0939054722-2011-01-07-12-17-59ScoredV3.0.csk
TCD3V3.5CIN2076705627-2011-03-16-14-48-47ScoredV3.0.csk
TCD4V3.5CIN0781482116-2011-05-10-15-10-49ScoredV3.0.csk
TCD5V3.5CIN1370460189-2011-05-11-15-02-05ScoredV3.0.csk
TCE1V3.5CIN1080838032-2010-05-07-12-10-52ScoredV3.0.csk
TCE2V3.5CIN0665001820-2010-06-03-13-55-53ScoredV3.0.csk
TCE3V3.5CIN1792930152-2010-06-08-08-59-22ScoredV3.0.csk
TCE4V3.5CIN1841775235-2010-06-23-11-32-41ScoredV3.0.csk
TCE5V3.5CIN0752776689-2010-12-09-15-56-30ScoredV3.5.csk
TCE6V3.5CIN1970844532-2011-01-25-14-31-04ScoredV3.0.csk
TCE7V3.5CIN0988448711-2011-04-05-15-25-31ScoredV3.0.csk
TCE8V3.5CIN0668952640-2011-05-13-10-48-08ScoredV3.0.csk
TCM1ScoredV4.0.csk
TCM2V3.5CIN0788977646-2010-07-08-13-35-31ScoredV3.0.csk
TCM3V3.5CIN1788079352-2011-03-11-13-07-53ScoredV3.0.csk
TCM4V3.5CIN0196784358-2011-04-08-16-19-15ScoredV3.0.csk
TCM5V3.5CIN1878643352-2011-05-11-10-12-04ScoredV3.0.csk

TCM6V3.5CIN1029212004-2011-05-11-12-10-16ScoredV3.0.csk
TCM7V3.5CIN2022489965-2011-05-13-14-41-23ScoredV3.0.csk

B. Clock Drawings Referenced in Images



a) b)

Figure 3.2.1.1
a) VIN1021133203, command clock
b) VIN0829757788, copy clock

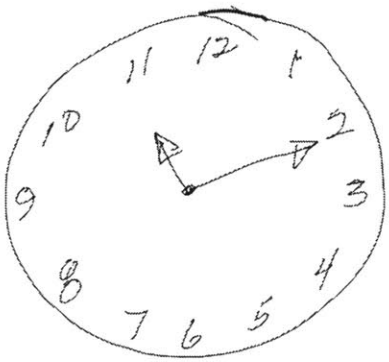
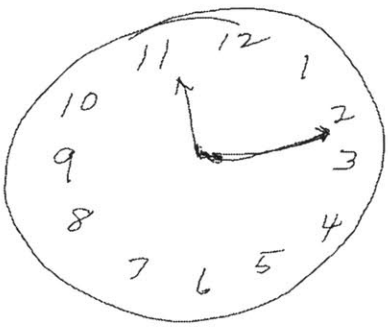


Figure 3.2.1.2 TCM7, copy clock



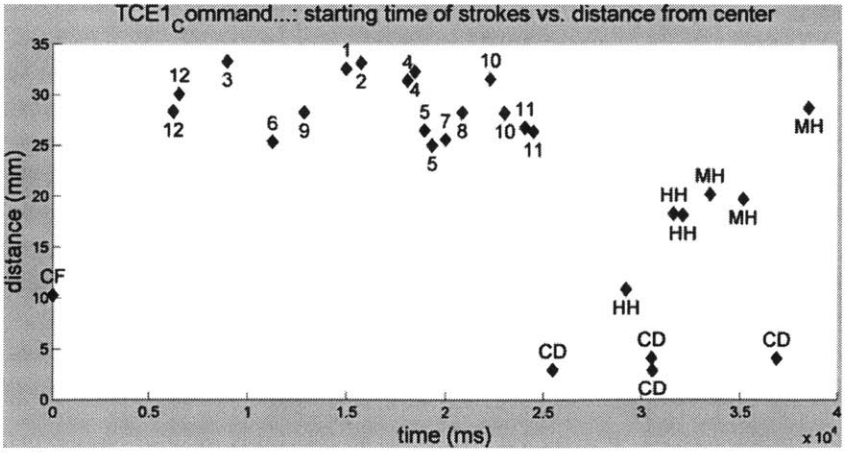


Figure 3.3.1.1 TCE1, command clock

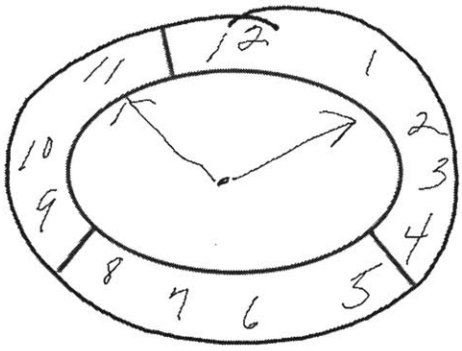


Figure 3.3.2.1 TCE3, copy clock

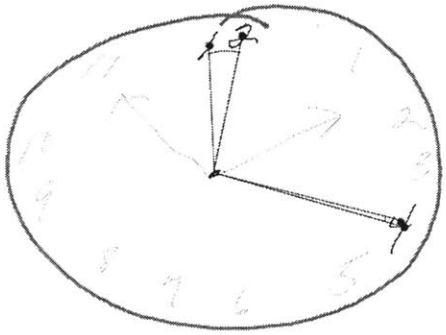


Figure 3.4.2.1 TCE3, copy clock

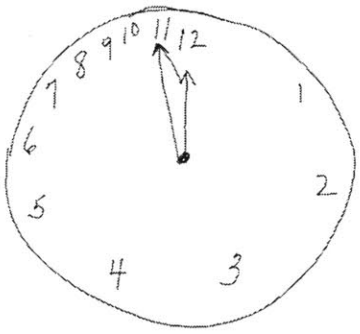


Figure 3.4.2.2 TCE8, command clock

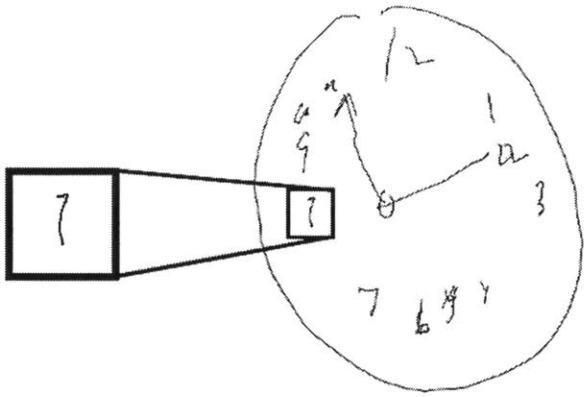


Figure 3.5.1.1 TCD4, copy clock

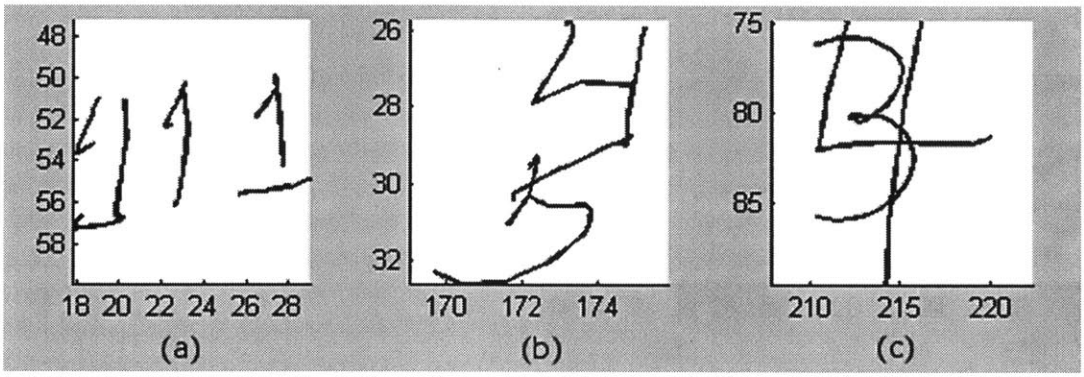


Figure 3.5.1.2

a) EGE0882081772, copy clock
b) TCD3, command clock
c) EGE0143851472, command clock

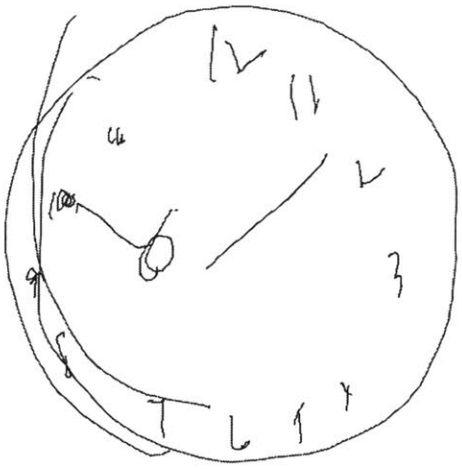
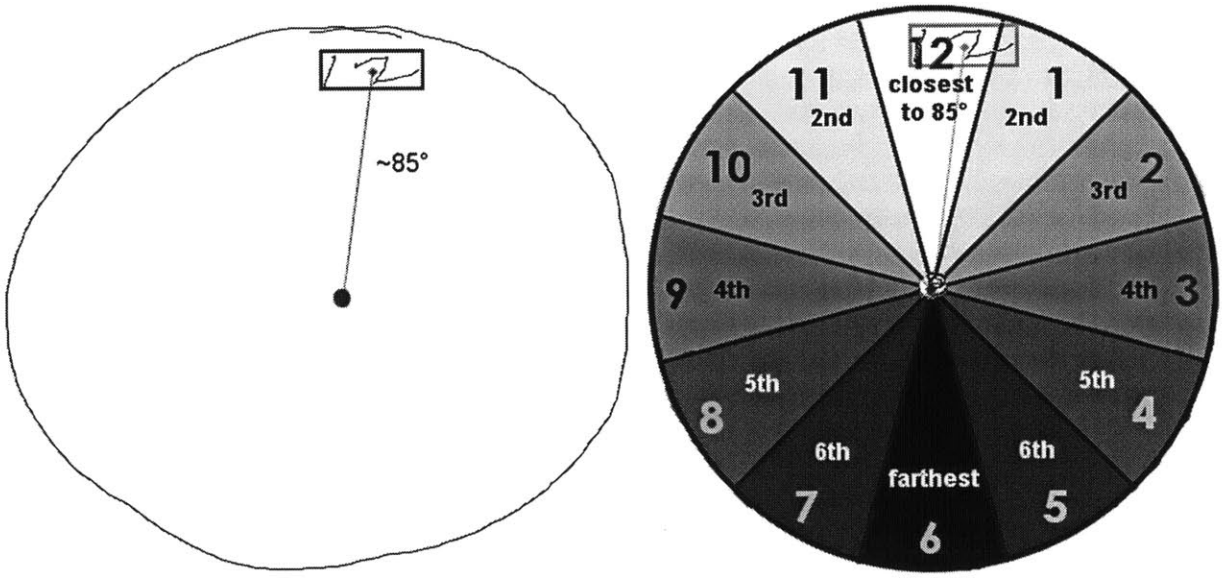


Figure 3.5.3.1 TCD4, copy clock



a) A symbol at 85 degrees. b) Proximity of each numeral type to symbol in a)

Figure 3.5.3.2 TCE3, copy clock

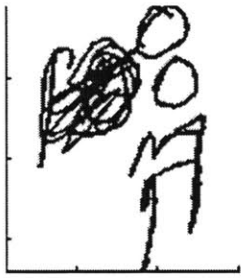


Figure 3.7.2 TCD3, copy clock

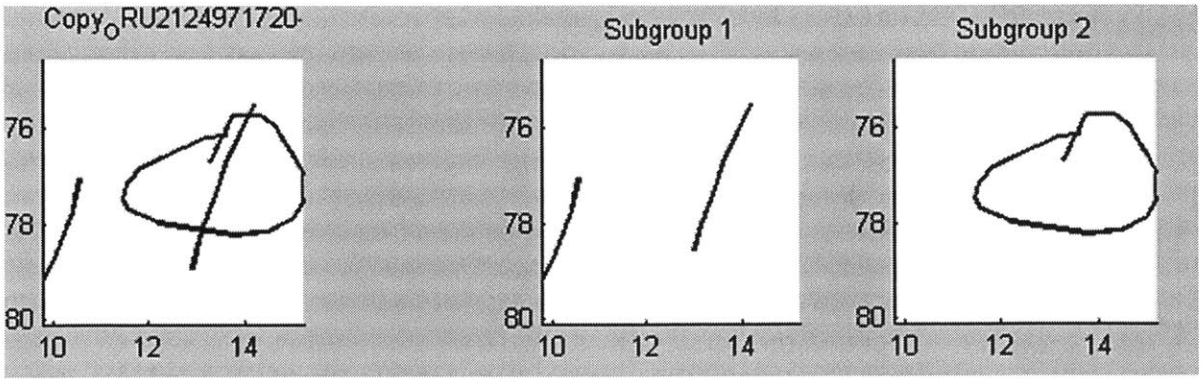


Figure 4.1.1 ORU2124971720, copy clock

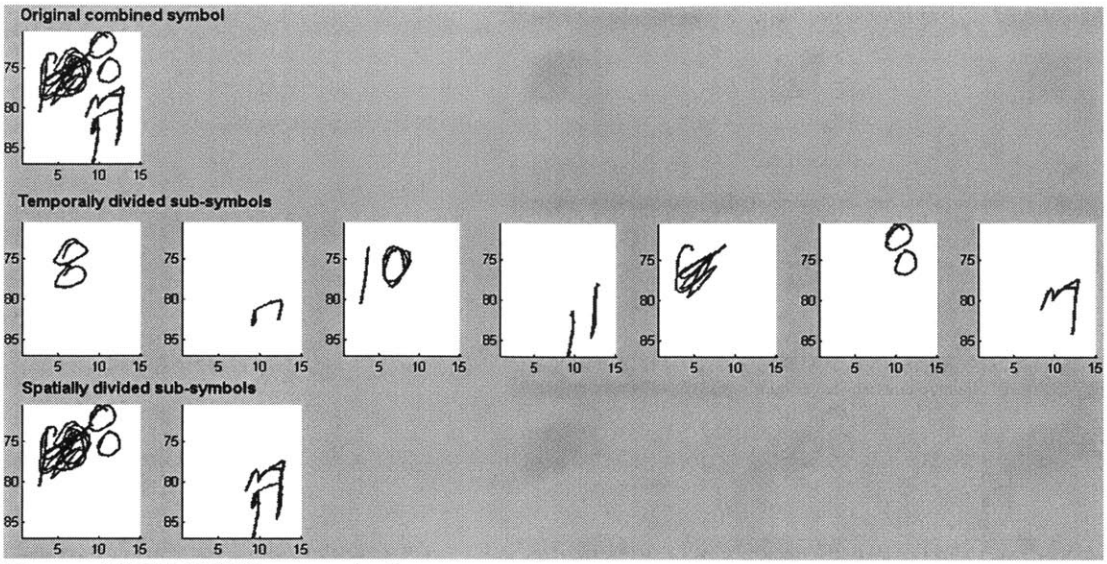


Figure 4.2.6.1 TCD3, copy clock

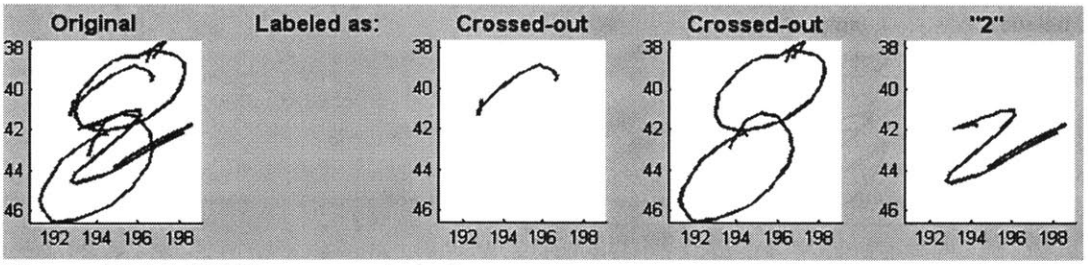


Figure 4.3.2.1 TCD5, command clock

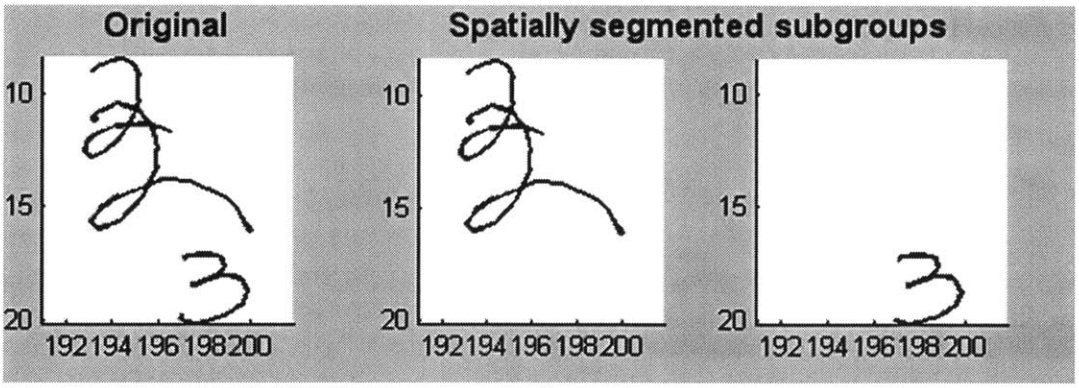


Figure 4.3.3.1 EGE0720764651, command clock

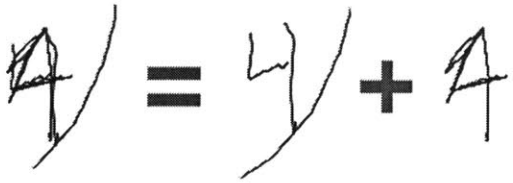


Figure 5.1 YDU2100188187, copy clock