Balancing appearance and context in sketch interpretation

Balancing Appearance and Context in Sketch Interpretation

Yale Song[1], Randall Davis[2], Kaichen Ma[3], Dana L. Penney[4]
[1]Yahoo Research, [2]Massachusetts Institute of Technology,
[3]Google Inc., [4]Lahey Hospital and Medical Center *

Abstract

We describe a sketch interpretation system that detects and classifies clock numerals created by subjects taking the Clock Drawing Test, a clinical tool widely used to screen for cognitive impairments (e.g., dementia). We describe how it balances appearance and context, and document its performance on some 2,000 drawings (about 24K clock numerals) produced by a wide spectrum of patients. We calibrate the utility of different forms of context, describing experiments with Conditional Random Fields trained and tested using a variety of features. We identify context that contributes to interpreting otherwise ambiguous or incomprehensible strokes. We describe ST-slices, a novel representation that enables "unpeeling" the layers of ink that result when people overwrite, which often produces ink impossible to analyze if only the final drawing is examined. We characterize when ST-slices work, calibrate their impact on performance, and consider their breadth of applicability.

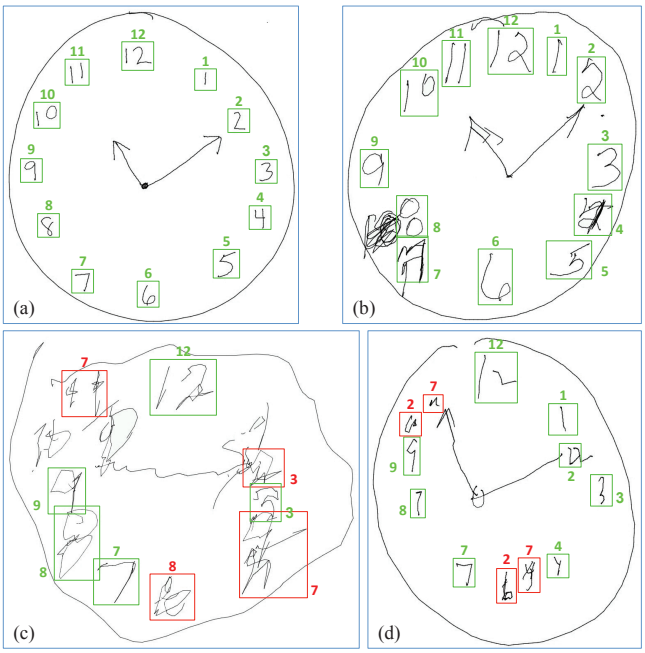

Figure 1: Our task is to detect and to classify clock numerals from drawings created by subjects taking the Clock Drawing Test. Shown here are (a) simple example (healthy subject), (b) overwritten, (c,d) impaired subjects, and our numeral detection/classification results (green: correct, red: incorrect).

1 Introduction

Our real-world sketch interpretation task comes from the Clock Drawing Test, a deceptively simple test that has been used for more than fifty years to help determine cognitive status. This is a task of growing importance given the "greying" of populations around the world and the increasing impact of cognitive impairments [Alzheimer's Association, 2013]. The test asks a subject to draw on a blank sheet of paper a clock face showing a particular time, then asks them to copy a predrawn clock shown on another sheet. Decades of experience with the test have made it a widely used screen for cognitive impairment of many sorts, e.g., Alzheimer's, dementia, stroke, etc. [Solomon *et al.*, 1998].

Interpretation of the test is traditionally based on what the patient drew (e.g., are all the numbers present at correct locations) and how accurately it was drawn (e.g., how closely do the hands point to the correct numerals). Interpretation is currently entirely manual, requires extensive clinical experience and knowledge, is at times labor-intensive, and has low inter-rater reliability. These issues limit its clinical effectiveness, particularly when the goal is detecting impairment early in disease development.

Since 2006, our research group has been conducting the THink project [Davis *et al.*, 2015], administering the digital Clock Drawing Test (dCDT) using a digitizing ballpoint pen (from Anoto Inc.). The pen records its position on the paper with considerable precision ($\pm$ 0.002 inches) every 13.3 ms, providing a sequence of time-stamped coordinates. The data thus captures both the drawing and the behavior that created it. We have accumulated a database of several thousand tests from both healthy and cognitively impaired subjects, a

*This work was done when Y. Song and K. Ma were at MIT Computer Science and Artificial Intelligence Laboratory.

Appeared in *Proceeding of the International Joint Conference on AI, 2016 (NY, NY)*

unique dataset of real-world drawing behavior. We have established and recorded ground truth labels for each pen stroke (i.e., which numeral it is, which hand, etc.), allowing us to calibrate the performance of our stroke classifiers.

Our task in this paper is a slightly unusual form of sketch interpretation: we know from the outset what the user is trying to draw – a clock. The challenge is to understand what they actually produced, i.e., determine the right label for every stroke. Correctly labeled strokes are crucial to the ultimate aim of this work: automated screening (disease vs healthy) and diagnosis (disease selection). Recent work has shown that correctly labeled strokes enable both of these to be done with demonstrably higher performance than existing algorithms [Souillard-Mandar *et al.*, 2015]. We focus here on clock numerals because numeral features play a particularly important role in clinical interpretation of the test.

Our task is clearly not traditional isolated digit recognition (e.g., MNIST [LeCun *et al.*, 1998]), as we have to determine which strokes are likely to be the digits, and where they are. It also is considerably more difficult than most digit recognition because impaired users at times draw digits that cannot possibly be interpreted accurately in isolation.

Despite its domain-specificity, the task presents interesting challenges widely applicable to sketch recognition. First, our system must deal with the range of phenomena produced by a population that is in some cases impaired and is in all cases completely naive. Our users are the furthest thing from trained users: A fundamental premise of the test is in fact that it captures a person's normal, spontaneous behavior. (This is one reason why we use the digitizing ballpoint; a tablet could distort the results by its different ergonomics and novelty, particularly for older users.)

Second, our system must handle corrections in natural handwriting. When users correct mistakes, they often do so by crossing out and over-writing, adding to the difficulty of interpretation, sometimes making recognition impossible to do from the image alone (e.g., the overwritten regions in Fig. 1 (b)). Finally, our system must be prepared for elements that are distorted, misplaced, repeated, or missing entirely (Fig. 1 (c) and (d)), as our users range in age from their 20's to well into their 90's, and span from completely healthy to those with cognitive impairments.

2 Our Approach

Our system starts its stroke recognition by finding an initial approximation of the center of the drawing, for use as a reference point for angular measures. This is frequently provided by identifying the clock circle, typically the longest circular stroke(s). We find a best-fit ellipse to the points in these strokes, giving both a good initial approximation to a center and a measure of clock size. If there is no clock circle (as can happen), we use the center of mass of all of data points.

2.1 Digit / Non-digit Stroke Classification

The first step is to separate digit stokes from non-digit strokes. We expect digits to be further from the clock center, and have observed a tendency for users to complete the task in categories: clock circle, numerals, hands (not necessarily in that order). There is thus both a temporal and a spatial dimension in which digit strokes may be grouped. Accordingly, we use k-means clustering [Arthur and Vassilvitskii, 2007] in a 2-D space defined by stroke starting time and distance from the clock center.

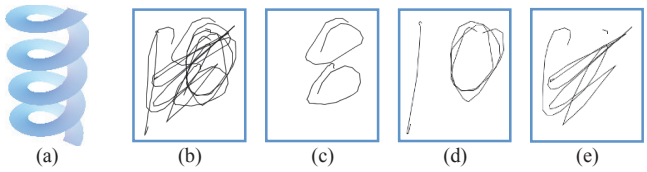

Figure 2: An illustration of ST-slices: (a) a helical ramp; (b) the final ink excerpt from Fig. 1b near the 8 position; (c, d, e) the final ink unpacked into ST slices.

The result is three groups of strokes, one intended to contain the clock circle, another intended to contain hand strokes, and the last intended to contain digits and other strokes sharing the same space/time region, e.g., tick marks. Since the majority of strokes represent numerals in a clock drawing, we consider the largest cluster the digit cluster and focus here on those, discarding others.

2.2 ST Segmentation of Digit Strokes

In terms of the original drawing, the set of presumed digit strokes lie in a very roughly annular area. The next task is to divide the annulus into segments intended to contain individual clock numerals (1 to 12). In a clean drawing, a simple angular difference threshold works well (e.g., Fig. 1 (a)). But an angular measure would not work for drawings like Fig. 1 (b), (c) or (d); in fact it is impossible to tell from the drawing alone what happened near the 8 position in Fig. 1 (b).

As our data is time-stamped, we can (literally) play a movie of the test, a remarkably effective way of enabling a person to make sense of complex, overwritten ink.

Wanting the computer to similarly unravel the layers of ink motivated a hybrid representation combining space and time. A spatial representation alone is inadequate because of overwriting, which can produce incomprehensible ink (e.g., Fig. 1 (b)). A temporal representation alone is inadequate because users may go back and add strokes to areas where they drew previously; these additional strokes make sense only in the context of what was drawn there (perhaps much) earlier.

We thus segment the annulus by considering both space and time. For any pair of strokes in time sequence, we compute their central angle (with respect to the estimated center of a clock) and their time difference, producing a 2-D feature vector. We then examine each pair sequentially, and segment them using a boosted logistic regression classifier [Friedman *et al.*, 2000] trained on a held-out dataset, annotated with binary labels indicating the ground truth segments.

We call the resulting representation spatio-temporal slices (ST-slices) and find that they are an effective representation for untangling the ink in complex drawings. One way to make the idea more intuitive is to imagine the digits being drawn on a helical ramp of paper (Fig. 2 (a)), where the vertical dimension is time. We segment the ramp whenever two consecutive

Algorithm 1 Detecting Overwriting and Augmentation

Input: Chronologically ordered set of ST-slices $\mathcal{S}$
procedure OVERWRITING–AUGMENTATION($\mathcal{S}$)
  for $(s_i, s_j) \in \mathcal{S}, j > i$ **do**
    if overlap$(s_i, s_j) > \theta_1$ **then**
      // Deemed overwrite: s_j overwrites s_i
      $\mathcal{S} \leftarrow \mathcal{S} \setminus \{s_i\}$
    else if overlap$(s_i, s_j) > \theta_2$ **then**
      // Deemed augmentation: s_j augments s_i
      $s_i \leftarrow$ merge(s_i, s_j)
      $\mathcal{S} \leftarrow \mathcal{S} \setminus \{s_j\}$
    end if
  end for
end procedure

strokes are sufficiently far apart angularly and/or temporally according to the segmenter. The result is a collection of ST-slices, each holding a set of strokes that is likely to represent a single clock numeral.

The clock in Fig. 1(b) provides one example of the effectiveness of the ST-slice representation. The ST-slices produced near the 8 position of the clock reveal what is otherwise hidden: As shown in Fig. 2, there was first an "8" drawn in the appropriate position on the clock (Fig. 2(c); undetectable in the final drawing). The 8 was later overwritten by a 10 (Fig. 2(d)); both of these were later scratched out (Fig. 2(e)). (The "8" clearly visible in Fig. 1(b) is a distinct pair of strokes added later next to the over-writes.)

In terms of the ST-slices, each of these drawing actions appears on a separate layer of the helical ramp, with their overlapping angular position on the clock face captured in their matching angular positions on the ramp. ST-slices are effective in "unpeeling" layers of ink produced when people overwrite. The resultant ability to see into the layers and interpret overwritten characters appears to be unique to our work. They are also "low cost," in the sense that they default to ordinary angular slicing in the absence of overwriting.

2.3 Overwriting and Augmentation

When ST-slices overlap spatially, we look at pairs of those slices to determine whether they represent overwriting (the second set of strokes is intended to replace the first), or augmentation (the second augments the first).

Algorithm 1 describes our approach. Given a chronologically ordered set of ST-slices, we examine each pair of slices (s_i, s_j), $j > i$, and determine overwriting and augmentation based on stroke area overlap. This is based on the assumption that two slices that capture an overwrite are likely to have extensive area overlap, as measured by the intersection of the convex hulls of their respective stroke sets. (The intersection is measured in either direction – s_i as percent of s_j and vice versa – if one does not wholly contain the other, and as the smaller as a percentage of the larger if one does contain the other.) Augmentations, by contrast, are likely to have small area overlap (e.g., adding just a "hat" or a "foot" to a 1). We empirically set the threshold values based on training data (θ_1 is 60% and θ_2 is 5%).

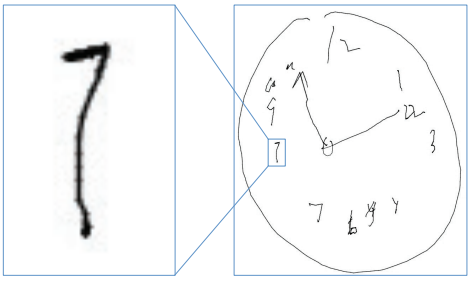

Figure 3: Context can be crucial in clock numeral interpretation. When considered in context, the number on the left side is interpreted by trained analysts as an 8, because it sits between the 7 and 9.

Strokes in a slice deemed overwritten are classified for future reference, then removed from further consideration, because the user replaced them. We classify the strokes using the sketched symbol recognizer of [Ouyang and Davis, 2009b], trained on a set of properly segmented clock numerals drawn by healthy participants.

A slice deemed to be an augmentation is merged with its corresponding slice(s).

The result at this point is a set of slices that we believe represent what the subject intended the final drawing to show, with overwritten ink analyzed and removed.

2.4 Clock Numeral Classification

As has long been recognized, context can be crucial in interpretation. Consider the digit in Fig. 3. It is clearly ambiguous and could be a 1 or a 7. Yet when considered in context it is interpreted by trained analysts as an 8 (it sits between the 7 and 9). We want our system to have a similar ability to consider more than just the appearance of each set of pen strokes.

We take context into account by formulating the problem as structured prediction, and use a conditional random field (CRF) [Lafferty *et al.*, 2001] that provides one way to combine information about what pen strokes look like with contextual information such as where they are in the clock, what's on either side of them angularly, etc.

We could encode the spatial relationship between the n digit slices using a circular chain structure, mimicking the annular area in a clock face. But this loopy structure makes it difficult to perform inference in the CRF model, as it requires performing approximate inference using methods such as alpha expansion [Szummer *et al.*, 2008] or mean field approximation [Koltun, 2011].

In response, we take advantage of the fact that our graph is a simple circular chain and break it right after the north slice of the clock face (often the numeral 12), creating a loop-free linear chain graph. We then append to each slice the adjacent slices on either side of it, so that $x_i = [x_{i-1};\ x_i;\ x_{i+1}]$. Each slice feature vector now has a dimension three times the original size. This combination of breaking the circle and concatenating feature vectors permits the use of an efficient belief propagation algorithm [Pearl, 1982], while still taking account of spatial context.

Algorithm 2 Repair Strategy for Under-segmentation

Input: Set of ST-slices $\mathcal{S}$, ST-slice s
procedure REPAIR–UNDERSEGMENTATION($\mathcal{S}, s$)
  // Find the best scoring partition of s
  $\mathcal{P} \leftarrow$ partitions of s into sets of chronological strokes
  for each partition $P_i \in \mathcal{P}$ **do**
    for each set of strokes $p_j \in P_i$ **do**
      score of $p_j \leftarrow$ recognize(p_j)
    end for
  end for
  $P^* \leftarrow$ partition with the highest average score
  score of $s \leftarrow$ recognize(s)
  if the highest average score $>$ score of s **then**
    // Find non-overlapping subset of slices
    for $(p_i, p_j) \in P^*, j > i$ **do**
      if overlap$(p_i, p_j) > \theta_1$ **then**
        $P^* \leftarrow P^* \setminus p_i$
      end if
    end for
    // Replace s with P^*, preserving stroke order in $\mathcal{S}$
    $\mathcal{S} \leftarrow (\mathcal{S} \setminus \{s\}) \cup P^*$
  end if
end procedure

Algorithm 3 Repair Strategy for Over-segmentation

Input: Set of ST-slices $\mathcal{S}$, consecutive ST-slices s_i, s_j
procedure REPAIR–OVERSEGMENTATION($\mathcal{S}, s_i, s_j$)
  merged score $\leftarrow$ recognize(merge(s_i, s_j))
  original score $\leftarrow$ recognize(s_i)
  if merged score $>$ original score **then**
    $s_i \leftarrow$ merge(s_i, s_j)
    $\mathcal{S} \leftarrow \mathcal{S} \setminus \{s_j\}$
  end if
end procedure

We define the CRF with a singleton term that learns the *appearance* of digits as the compatibility between a digit label y_i and the feature vector x_i for the i-th slice, and a pairwise term that captures the *context* as the compatibility between two angularly consecutive digit labels y_{i-1} and y_i. For each ST-slice the CRF returns a 12-element vector indicating the probability of each numeral label for that slice.

Training of the CRF follows the standard gradient descent approach [Lafferty *et al.*, 2001]. To avoid over-fitting, we used an l_2 regularization with its scale term set at 0.01, chosen based on 10-fold cross validation.

To determine the appropriate feature representations to use, we trained and tested 12 CRF classifiers with different combinations of features, including the 24 x 24 feature images from the [Ouyang and Davis, 2009b] symbol recognizer, the angular position of the slice, and the number of pen strokes in the slice. We review the results in Section 3.

2.5 Repairs

The ST-slice mechanism is of course not perfect: it will under-segment in some cases and over-segment in others. In Fig. 1(b), the 2 overwritten with the 4 (at the 4 position) produces an under-segmentation, because the 2 was immediately overwritten with the 4, yielding a single ST-slice with all three strokes in it. Conversely, ST-slices can over-segment in circumstances like those in Fig. 4, described below.

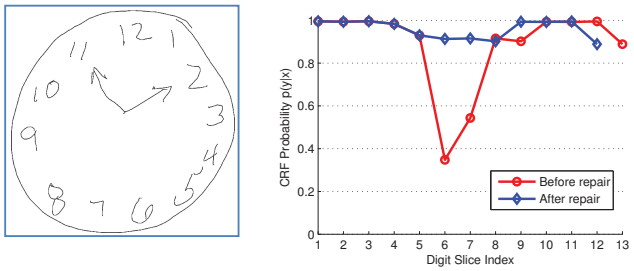

Figure 4: A clock drawing sample and the corresponding CRF probabilities before and after applying our repair strategy for over-segmentation. The 5 is initially over-segmented into the 6th and the 7th digit slices, but later corrected.

In response, the next step is to examine the posterior probabilities provided by the CRF for the highest ranking interpretation for each slice in sequence, and detect "valleys" in this score as a way of finding places where the stroke interpretation may be incorrect (e.g., the 6th and the 7th digit slices) Fig. 4). We consider a slice to be a valley if its probability score is at least 30% lower than the score for either of its adjacent slices. Looking at score differences from slice to slice rather than absolute scores adjusts to some extent for clocks with generally badly drawn digits.

When multiple potential repair sites are found, we start with the repair site whose adjacent slice has the highest score, trying to work from what we are relatively more sure of (the high-scored slice) to help guide the repair (referred to as the "islands of certainty" strategy in [Erman *et al.*, 1980]).

After each repair, the entire set of (revised) ST-slices is sent back to the CRF to determine whether the repair improved the interpretation. An improvement in one slice can of course propagate, affecting scores in adjacent slices, leading to an overall improvement and/or the identification of additional repair site candidates. This process continues until no new repair sites are identified.

We developed repair strategies to handle both over-segmentation (e.g., where a single numeral was split into two slices) and under-segmentation (e.g., multiple digits so close angularly they end up in a single ST-slice). Slice properties determine which strategy to apply: oversegmentation is applied to valleys formed by two consecutive slices, while undersegmentation is applied to valleys containing only a single slice that is unusually wide angularly and that has an unusually large number of strokes in it ("unusual" is defined as more than one standard deviation larger than the average of that property).

Algorithm 2 attempts to deal with under-segmentation by partitioning strokes in a slice into multiple subsets of chronological strokes. For example, given a slice with three strokes $\{1, 2, 3\}$, we create three partitions $\mathcal{P} = \{P_1, P_2, P_3\}$ where $P_1 = \{\{1\}, \{2, 3\}\}$, $P_2 = \{\{1, 2\}, \{3\}\}$, and $P_3 = \{\{1\}, \{2\}, \{3\}\}$. It then applies the sketched symbol recog-

nizer mentioned in Section 2.3 [Ouyang and Davis, 2009b], which has been trained to recognize the twelve clock numerals in isolation, and uses the results to determine whether any of the partitions improves the interpretation. Once the slice is re-segmented using the optimal partition, we remove any subset of strokes if it has been overwritten by strokes in a subsequent subset in that slice (using the same $\theta_1 = 60\%$ criterion as in Algorithm 1). As one example, this deals successfully with the 2 overwritten by a 4 in Fig. 1(b).

Fig. 4 illustrates an over-segmentation: the top stroke of the 5 is sufficiently far angularly from the midpoint of the base stroke that the two are put in separate slices. Algorithm 3 looks for a valley two slices wide, tries merging them, and evaluates the interpretation of the new set of slices. The graph in Fig. 4 shows the successful repair, with the CRF probabilities demonstrating significant improvement from initial segmentation (red, 13 slices) to the repaired segementation (blue, 12 slices).

3 Experiments

We used a set of 2,024 clock drawings collected from clinical facilities: 1,654 clocks drawn by healthy participants and 370 clocks randomly selected from mildly cognitively impaired participants. Ground truth labeling was provided by trained analysts, including segmentation and identification of each clock numeral slice.

3.1 Clock Numeral Identification

Our final digit identification accuracy ranged from 99.14% for healthy-user clocks to 92.32% for impaired-user clocks. For healthy users virtually all errors were in segmentation; using ground truth segmentation labels, digit identification was 99.32% (see overall performance in Table 1). The segmentation errors were typically over-segmentations (e.g., splitting the two digits in a "12"). Our repair strategies dealt successfully with about half the segmentation errors that occurred, producing the final result of 99.14%.

Unsurprisingly, impaired clocks had more errors in identification (3.66%) and segmentation (4.08%), yielding an overall accuracy of 92.32%. Errors in these cases typically resulted when a sequence of digits were drawn badly (e.g., the 1,2,3,4,5,6 sequence in Fig. 1 (c)), preventing context from offering sufficient guidance.

Overall our performance compares favorably to the state of the art isolated digit recognition – an accuracy of 99.79% on the MNIST dataset [Wan *et al.*, 2013] – considering the added difficulties we face (non-isolated digits, 12 labels to assign, some extremely challenging data). It also illustrates the power of context to aid interpretation, something obviously not used in isolated digit recognizers.

3.2 Quantifying The Impact of ST-Slices

Our segmentation method (ST-slices) can deal with overwriting that is otherwise incomprehensible even to humans. To see its effectiveness, we selected 70 clocks in our dataset that contain at least one incomprehensible part resulting from overwriting.

Our ST-slice mechanism separated the layers of ink accurately in 79% of these cases. As a comparison, a purely temporal segmenter – a boosted logistic regression classifier [Friedman *et al.*, 2000] trained on just the time difference between successive strokes – performed significantly worse at an accuracy of 66%.

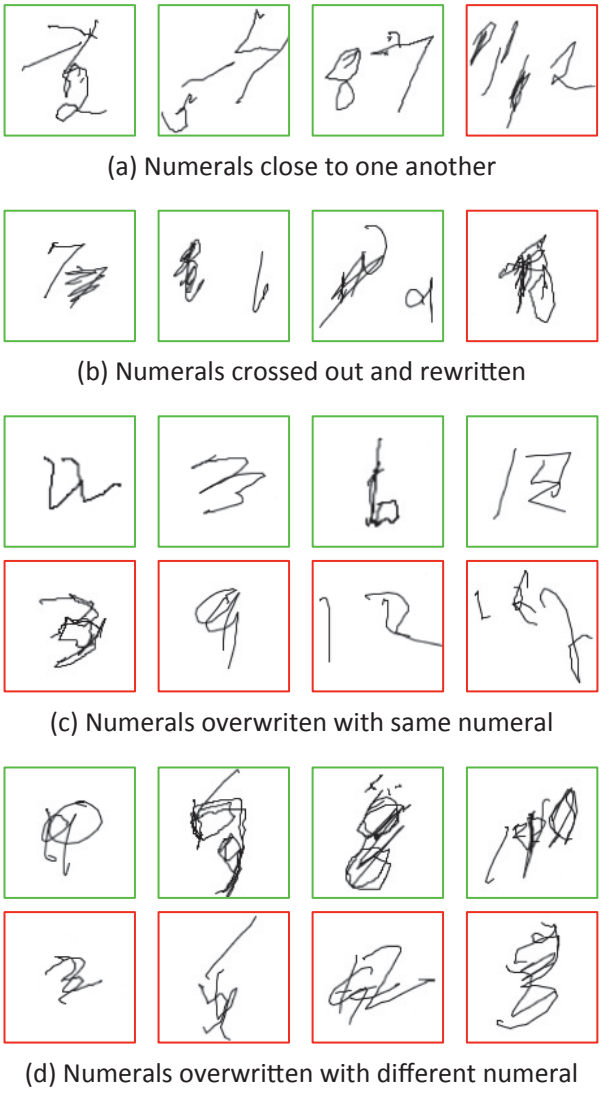

Figure 5: Success (green) and failure (red) cases of ST-slices, categorized into four canonical types of complications.

Fig. 5 provides some insight into when our technique works well and when it fails. ST-slices performed well on cases where two slices are spatially close (which confused a purely spatial segmenter), but still temporally distinguishable. This handles cases in which two numerals are written near each other but at different times (Fig. 5a) and cases in which a numeral is crossed out and another written next to it (b). Our technique correctly segmented most of the slices produced in these cases (the green boxes in Fig. 5a and Fig. 5b).

Given the complexity of the ink that results from overwriting, none of these would have been handled by existing digit recognition or sketch interpretation techniques.

Trained on	Input	Ang, # Stk	Overall Perf (sd)	Perf Healthy (sd)	Perf Impaired (sd)
Healthy	single		89.93% (0.0101)	93.83% (0.0103)	71.03% (0.0309)
Healthy	single	Y	87.51% (0.0068)	90.51% (0.0074)	73.01% (0.0321)
Healthy	concat.		98.43% (0.0049)	96.76% (0.0018)	92.01% (0.0243)
Healthy	concat.	Y	99.01% (0.0045)	99.90% (0.0011)	94.66% (0.0232)
Impaired	single		75.11% (0.2420)	78.43% (0.0268)	59.04% (0.0232)
Impaired	single	Y	76.76% (0.0285)	79.05% (0.0328)	65.67% (0.0234)
Impaired	concat.		98.84% (0.0055)	99.75% (0.0024)	94.45% (0.0234)
Impaired	concat.	Y	99.05% (0.0092)	99.82% (0.0025)	95.31% (0.0429)
Both	single		88.96% (0.0080)	92.88% (0.0094)	69.95% (0.0224)
Both	single	Y	88.29% (0.0137)	92.18% (0.0140)	75.31% (0.0264)
Both	concat.		98.85% (0.0048)	99.82% (0.0018)	94.13% (0.0228)
Both	concat.	Y	**99.32% (0.0028)**	**99.93% (0.0009)**	**96.34% (0.0160)**

Table 1: We report mean accuracy and standard deviation across 10 folds. We measured the effect of several factors, including choice of training set (healthy, impaired, both), whether feature vectors for the CRF were concatenated for three adjacent slices, and whether the feature vector contained only 24 x 24 feature images, or included slice angular position and stroke count.

One source of failure came from immediate overwriting, with either the same (Fig. 5c) or a different digit (Fig. 5d), as these strokes end up in the same ST-slice. As noted, we handle this by examining sequential subsets of strokes in the slice, looking for subsets recognizable as a digit. Fig. 5c and d shows examples of both successes and failures.

A relatively rare problem arises from numerals assembled with non-chronological strokes; our system always fails to correctly (re)group temporally interspersed strokes. These scenarios require determining intent, and are a focus of continued work.

3.3 Quantifying The Impact of Context

Given the important role of context in interpretation, particularly in clocks by impaired users, we did a set of experiments to provide a quantitative calibration of the contribution of different context features.

We trained and tested our digit classifier with 12 different combinations of context features, using as the base feature the 24 x 24 feature images from the [Ouyang and Davis, 2009b] symbol recognizer. To focus on calibrating the contribution of context features, training and testing was done on correctly segmented pen strokes (i.e., isolated clock numerals), using 10-fold cross validation.

We report mean accuracy and standard deviation across the folds. As Table 1 suggests, adding angle and stroke count alone increases accuracy only a small amount, from 98.85% to 99.32% (the last two rows). Concatenating data from adjacent slices provides a more significant performance boost, from 88.96% to 98.85% (the second and the fourth to the last row). Not surprisingly, concatenation provided a particularly significant benefit for impaired clocks, where performance improved from 69.95% to 94.13%.

In terms of training sets, training on only the impaired clocks typically gave the worst results, even for performance on impaired clocks, perhaps because some digits are so distorted that they are blurring the ability to distinguish more clearly drawn digits.

Overall these results give quantitative evidence for the significant contribution made by considering each digit in the context of the two on either side of it. While not unexpected, it is useful to see that it provides the kind of capability evidenced by human performance, e.g., for the digit in Fig. 3, which is now properly classified as an 8.

4 Related Work

Work in [Kim, 2013] is also focused on the clock drawing test, but emphasizes interface design, seeking to create a tablet-based system that is usable by both subjects and clinicians. It does basic digit recognition but does not report dealing with distorted or over-written digits, or the other complexities described here.

Digit recognition has a long history, for both online and offline capture: [Plamondon and Srihari, 2000] provides a survey, recent work on convolutional neural networks offers current performance levels [Ciresan *et al.*, 2012; Wan *et al.*, 2013]. While we need a good digit recognizer, our larger focus is on the identification of digits in context in challenging conditions (e.g., distorted by impairment, crossed out, overwriting, etc.), which has not been studied extensively.

Our work on ST-slices builds on and extends a long history of using temporal information in sketch understanding, where it has been used in a variety of ways, including stroke segmentation (e.g. [Ouyang and Davis, 2009a]) and interpretation [Sezgin and Davis, 2008; Taele and Hammond, 2008]. While previous work has explored over-tracing (writing the same thing on top of itself, often for emphasis, e.g., [Sezgin and Davis, 2004]), overwriting (writing to replace) appears to

be rarely tackled. [Hürst *et al.*, 1998] take on the task, but set it in the context of interactive handwriting recognition, in which the user and the system work cooperatively, which is inappropriate for our task. Work in [Dickmann *et al.*, 2010] combines spatial and temporal information to do stroke segmentation, but appears to deal only with consecutive strokes.

Spatio-temporal information have also been used together in a variety of circumstances, as for example dense trajectories [Wang *et al.*, 2013] and C3D [Tran *et al.*, 2015] for video analysis in the computer vision community. Our visual feature representation from [Ouyang and Davis, 2009b] can be thought of as 2D convolutional features obtained from pre-defined filters; using 3D convolutional features [Tran *et al.*, 2015] to represent our pen stroke data could give us extra performance boost. This is the focus of our ongoing work.

5 Conclusions

Our sketch interpretation system balances visual appearance and context, enabling it to handle some of the complex phenomena found in drawings produced during a real-world task performed by a wide range of naive users behaving as they do ordinarily.

We demonstrate our system on detecting and classifying clock numerals from drawings created by subjects taking the digital Clock Drawing Test. We report performance on clock numeral detection and classification at the 96%–99% level even for sketches drawn by those with impairments, calibrating and demonstrating the utility of different forms of context. We show that a novel representation – ST-slices – and the processing that accompanies it enables unpeeling and interpreting the layers of ink that result when people overwrite, giving our system a novel ability to classify symbols in a drawing even though they have been over-written or crossed out.

As the performance figures suggest, the system works well on clock numeral isolation and detection for clocks from both healthy and impaired users. Even so there are several avenues of improvement to be explored.

Perhaps the most important is to explore alternatives to the sequential character of our system (segmentation and recognition). Human segmentation of strokes is clearly guided in part by understanding the drawing, i.e., segmentation and interpretation work simultaneously. Recent research has shown that convolutional neural networks combined with region proposals are very effective at simultaneous detection and recognition of visual objects from images [Girshick *et al.*, 2014]. Applying techniques like this to our domain would require additional steps to deal with cross-outs and over-writing. We are looking into ways to accomplish this and expect it to reduce the role of segmentation as a major source of clock numeral identification error.

We have focused here on clock numeral identification because it is the most difficult part of the task, but full sketch interpretation for the clock means recognizing all the other elements (hands, tick marks, etc.). We have a start on this but there is considerably more to do.

Where possible we compared our work to the state of the art, e.g., its performance on healthy clocks, which is comparable to the best isolated digit recognizers on the MNIST dataset [Wan *et al.*, 2013]. But an important, novel part of our system is space-time slicing for sketch recognition, and its ability to handle the cross-outs and over-writing that occur in ordinary drawing and writing. Since there is yet no standard dataset or alternative program to compare directly against for this part of the system, we provided the qualitative analysis (Fig. 5) that explains in detail when space-time slicing works well and when it fails.

While ST-slices have been applied here to a specific task, the concept is more generally applicable to any sketch understanding task where layers of ink are encountered and need to be unraveled. More generally, they are a combination of the spatial and temporal properties of pen strokes and as such are applicable to a variety of sketch recognition domains, e.g., drawings of chemical structures [Ouyang and Davis, 2011].

The general idea of using context to guide signal understanding is of course both known and widely applicable. This work provides a calibration of its power on a specific task and more generally shows how spatial and temporal information can be captured in a way that is powerful on this specific task, yet general enough to be applicable to a variety of other hand drawing interpretation tasks.

Acknowledgement

This work was supported in part by NSF Grant IIS-1404494 and by the REW Research and Education Institution.

References

[Alzheimer's Association, 2013] Alzheimer's Association. Alzheimer's disease facts and figures. *Alzheimer's & Dementia*, 9(2):208–245, 2013.

[Arthur and Vassilvitskii, 2007] David Arthur and Sergei Vassilvitskii. k-means++: The advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics, 2007.

[Ciresan *et al.*, 2012] Dan Ciresan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *CVPR*. IEEE, 2012.

[Davis *et al.*, 2015] Randall Davis, David J. Libon, Rhoda Au, David Pitman, and Dana L. Penney. THink: Inferring cognitive status from subtle behaviors. *AI Magazine*, 36(3):49–60, 2015.

[Dickmann *et al.*, 2010] Lutz Dickmann, Tobias Lensing, Robert Porzel, Rainer Malaka, and Christoph Lischka. Sketch-based interfaces: Exploiting spatio-temporal context for automatic stroke grouping. In *Smart Graphics*, pages 139–151. Springer, 2010.

[Erman *et al.*, 1980] Lee D Erman, Frederick Hayes-Roth, Victor R Lesser, and D Raj Reddy. The hearsay-ii speech-understanding system: Integrating knowledge to resolve uncertainty. *ACM Computing Surveys*, 12(2):213–253, 1980.

[Friedman *et al.*, 2000] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder

by the authors). *The annals of statistics*, 28(2):337–407, 2000.

[Girshick *et al.*, 2014] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jagannath Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*. IEEE, 2014.

[Hürst *et al.*, 1998] Wolfgang Hürst, Jie Yang, and Alex Waibel. Error repair in human handwriting: an intelligent user interface for automatic online handwriting recognition. In *Intelligence and Systems, 1998. Proceedings., IEEE International Joint Symposia on*, pages 389–395. IEEE, 1998.

[Kim, 2013] Hyungsin Kim. *The ClockMe system: computer-assisted screening tool for dementia*. PhD thesis, Georgia Institute of Technology, 2013.

[Koltun, 2011] Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2011.

[Lafferty *et al.*, 2001] John Lafferty, Andrew McCallum, and Fernando Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.

[LeCun *et al.*, 1998] Yann LeCun, Corinna Cortes, and Christopher Burges. The mnist database of handwritten digits, 1998.

[Ouyang and Davis, 2009a] Tom Ouyang and Randall Davis. Learning from neighboring strokes: Combining appearance and context for multi-domain sketch recognition. In *NIPS*, 2009.

[Ouyang and Davis, 2009b] Tom Ouyang and Randall Davis. A visual approach to sketched symbol recognition. In *IJCAI*, 2009.

[Ouyang and Davis, 2011] Tom Y Ouyang and Randall Davis. Chemink: a natural real-time recognition system for chemical drawings. In *IUI*, 2011.

[Pearl, 1982] Judea Pearl. Reverend bayes on inference engines: A distributed hierarchical approach. In *AAAI*, 1982.

[Plamondon and Srihari, 2000] Réjean Plamondon and Sargur N Srihari. Online and off-line handwriting recognition: a comprehensive survey. *IEEE PAMI*, 22(1):63–84, 2000.

[Sezgin and Davis, 2004] Tevfik Metin Sezgin and Randall Davis. Handling overtraced strokes in hand-drawn sketches. *Making Pen-Based Interaction Intelligent and Natural*, 2, 2004.

[Sezgin and Davis, 2008] Tevfik Metin Sezgin and Randall Davis. Sketch recognition in interspersed drawings using time-based graphical models. *Computers & Graphics*, 32(5):500–510, 2008.

[Solomon *et al.*, 1998] Paul R Solomon, Aliina Hirschoff, Bridget Kelly, Mahri Relin, Michael Brush, Richard D DeVeaux, and William W Pendlebury. A 7 minute neurocognitive screening battery highly sensitive to alzheimer's disease. *Archives of neurology*, 55(3):349–355, 1998.

[Souillard-Mandar *et al.*, 2015] William Souillard-Mandar, Randall Davis, Cynthia Rudin, Rhoda Au, David J Libon, Rodney Swenson, Catherine C Price, Melissa Lamar, and Dana L Penney. Learning classification models of cognitive conditions from subtle behaviors in the digital clock drawing test. *Machine Learning*, pages 1–49, 2015.

[Szummer *et al.*, 2008] Martin Szummer, Pushmeet Kohli, and Derek Hoiem. Learning crfs using graph cuts. In *ECCV*. 2008.

[Taele and Hammond, 2008] Paul Taele and Tracy Hammond. Using a geometric-based sketch recognition approach to sketch chinese radicals. In *AAAI*, 2008.

[Tran *et al.*, 2015] Du Tran, Lubomir D. Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[Wan *et al.*, 2013] Li Wan, Matthew Zeiler, Sixin Zhang, Yann LeCun, and Rob Fergus. Regularization of neural networks using dropconnect. In *ICML*, 2013.

[Wang *et al.*, 2013] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. *IJCV*, 103(1):60–79, 2013.