Supplemental Materials

February 15, 2017

1 Detailed Overview of Algorithm

There are several parameter choices that impact the performance of the algorithm and its computational cost. As we describe the algorithm, we point out particular parameter entries in Table 1 that describe the parameter values that we used, and the corresponding name of these parameters in the publicly available software implementation.

Step 1: Image Preparation and Grid-line Extraction

Prior to any analysis, images are manually rotated and cropped to optimize the fit of the grid-lines within a rectangular bounding box (See Figure 1-A,B). Images are then converted to gray-scale, blurred with a Gaussian filter (Table 1-C), and dilated to help connect potential gaps in the grid-lines (Table 1-D). The grid-background may not be uniformly colored or illuminated, thereby complicating the identification of grid-lines later in the algorithm. Hence, adaptive thresholding [1] is applied to create a binary image without the effects of nonuniform illumination or coloring (Table 1-A,B). Next, the largest connected binary component in the image is extracted, which is assumed to be the grid-lines of the image(See Figure 1-C).

Step 2: Estimation of Row and Column Candidate Points

We next estimate the number of rows and columns in the preprocessed image. To determine the number of columns, the image is partitioned into multiple horizontal strips (Figure 1-C and Table 1-E). Smaller strips are increasingly useful as row or column lines grow decreasingly vertical or horizontal. For each horizontal strip, the Hough Image transform is applied to identify near vertical line segments (Table 1-G) that represent potential column grid-lines within the strip. In general, a higher number of Hough Peaks (Table 1-F) increases the reliability of the estimate at the cost of computation time. A large number of near vertical Hough peaks are expected near the location of the columns (See Figure 1-D). Hence, we may estimate the number of columns in the strip by identifying the optimal value of k, in the k-means clustering algorithm applied to the extracted Hough peaks (Table 1, I). More specifically, k is determined by incrementing over potential values (starting from 2), and inspecting the mean silhouette. The silhouette is a goodness of fit measure for each cluster in k-means. An optimal fit (where silhouette=1) is obtained when the inter-cluster distance between points is minimal relative to the intra-cluster distance between points. Silhouette values lower than one indicate the presence of points which may straddle multiple clusters. For more information see [4]. We choose the estimated number of columns in the strip as the value of k for which the mean silhouette is greater than 0.9, and the mean silhouette at $k+1$ is less than that observed at k. The estimated number of columns for the entire image is chosen as the statistical mode of the estimated number of columns across all strips (See Figure 1-D).

Once an estimated number of columns for the image is determined, we re-apply the k-means algorithm on the Hough peaks of each strip using the selected value of k. The set of cluster centroid locations returned by k-means in each strip are considered column candidate points (See Figure 1-E,F and Table 1, H). The approach to estimate the number of rows in the image is similar to the column-estimation approach just described, except that the preprocessed image is partitioned into vertical strips and near horizontal line segments are sought with the Hough transform.

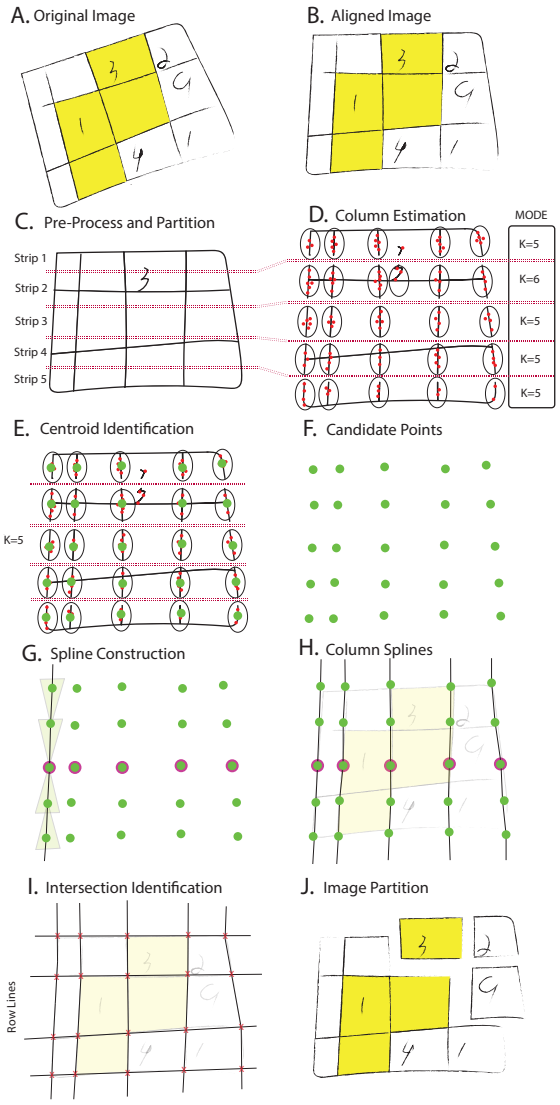

Figure 1: An illustration of Steps 1-4 of our method. (A) Example of a hand-drawn spreadsheet which we may wish to transcribe. (B) The spreadsheet after manual alignment of the image. (C) Black and white image following preprocessing with image strip lines. (D) Estimation of column number using k-means clustering on Hough transform vertical line peaks (illustrated as red dots). (E) Extraction of candidate points using k-mean centroids with k set according to the estimate. (F) A set of extracted column point candidates. (G) Column splines through the combination of candidate points. (H) Completed column splines. (I) Identification of intersection points between row and column splines. (J) Partitioning of the original image into individual cells.

Step 3: Extraction of Row and Column Splines

We construct row and column splines by connecting the identified row and column candidate points horizontally and vertically. To identify column splines, we first perform k-medoids on the set of column candidate points with k set equal to the estimated number of columns (Table 1,K). The resulting medoids are used as the starting locations for each of the constructed column splines (See purple dots in Figure 1-G). Beginning from these starting locations, we identify other points located above and below that may belong to the same spline. More specifically, we identify the nearest column candidate point which lies above the current point within a triangular search window (Table 1,J) and add this point to the column spline. We then re-reference the search with respect the identified point, and repeat the search until the top of the image is reached. The same process is used to identify points below the starting location (Figure 1-G). The result of this process yields k non-overlapping column splines Figure 1-H). A similar approach is utilized to identify k non-overlapping row-splines except that (1) starting locations are derived from the row candidate points and (2) we seek points located to the right and left of the starting locations.

Step 4: Extracting Cell Images

Using the set of identified row and column splines, we identify intersections between rows and columns. These intersection points outline the location of each cell in the grid, and are used to partition the original image into a matrix of smaller cell images which represent the contents of each cell in the grid (Figure 1-I,J). In many cases, the contents of a cell may not be entirely confined within the borders of the cell. Hence, extracted cell images may also include some area around the perimeter of the cell (Table 1-L).

Step 5: Machine Transcription

Following the extraction of cell-level images, we attempt to automatically transcribe contents using a machine learning approach. Prior to machine transcription, the cell images must first be pre-processed to remove cell border lines and noise while extracting individual digits for classification. The pre-processing steps applied before

Parameter	Suggested Settings	Name of Variable	Description
Step 1: Image preparation and Grid-line Extraction			
A	0.025	adap_thresh	Threshold of the adaptive filter
B	10	adap_size	Radius of the adaptive filter
C	1	pre_blur	Radius of the Gaussian filter
D	3	pre_dilation	Radius of the disk-shaped structuring element for the, morphological dilation
Step 2: Estimation of Row and Column Candidate Points			
E	20	seg_len	The strip size (in pixels).
F	4000	num_peaks	The number of Hough Transform peaks
G	10	theta_tol	The orientation of lines returned by the Hough Transform $+/-\lambda$ degrees from 0° for row lines
H	50	k_clust	Number of K-Mediods iterations for row and column estimation
I	100	k_times	Number of K-Mediods iterations for candidate point estimation
Step 3: Extraction of Row and Column Spline			
J	10	angle	Search Angle for spline construction
K	10	k_start	Number of K-Mediods iterations to determine starting locations for each spline.
Step 4: Extracting Cell Images			
L	0.05	area_around_cell	Area around the border of the cell images to include in cell images
Steps 5: Machine Transcription			
M	2	agresssion	The area around the spline to remove for the extracted cell images
N	.003	noise_chunk_perc	The minimum size (percent) of a connected component to be kept as a digit
O	5	border	The size of the border padding around the extracted digits
P	{[}2, 2{]}	cellSize	Histogram of Oriented Gradients, feature cell size

Table 1: A description of the default parameters of the spreadsheets transcription algorithm and he corresponding name of each parameter in our open-source implementation.

classification include: (1) removal of cell borders (and contents outside the border) from the images, (2) conversion of images to binary (black and white), and (3) image noise reduction.

The cell images are first converted from true-color to gray-scale intensity. Otsu's method is then used to determine a global grey-scale threshold and convert the image into a binary representation [5]. If 60% or more pixels are active (implying white text on black background), the binary image is inverted with a *not* operation. Images are then morphologicaly dilated using a disk-shaped structuring element (Table 1,P) [7]. The borders of the images are removed by setting all pixels within a given euclidean distance from the identified row and column splines to zero (Table 1,M). All pixels outside the estimated borders are also set to zero.

We further processed each cell image to extract individual digits prior to classification. We extract individ-

ual digits from the cell image by identifying the set of all connected components (pixel groups having one or more neighbor with a value of 1) in the image. All connected components whose total area is less than a given threshold are considered to be noise, while those above the threshold are considered as candidate digits (Table 1-N). The extracted candidate digits and padded with zeros (Table 1-O) and re-scaled to 28x28 pixels for classification by our machine learning algorithm. We consider cell images without any connected component that exceeds our threshold as blank cells.

Following digit extraction, a multi-class posterior probability Support Vector Machine model (MSVM) is used classify the individual handwritten digits (0-9) from the extracted cell images [6]. The classification scores returned by the MSVM reflect the probability of each digit, given the data. We will refer to maximum probability across all digits for a given image as the *confidence* of the classifier. The MSVM was trained using histogram of orientated gradient (HoG) features (Table 1-P) extracted from the publicly available MNIST dataset [3].

Step 6: Human Transcription

Human transcription is performed via Amazon's Mechanical Turk [2] service on any cell images for the which the MSVM's confidence value is less than a given threshold. Human agents are paid $0.01 per cell transcription and were provided with the following instructions: (1) You must transcribe the text shown in the image. (2) If the image is BLANK, then enter "NOTHING". (3) PLEASE NOTE: Your transcription will be automatically crossed checked against other Turk users to ensure accuracy. To help ensure the accuracy of the human transcription, we only utilized workers with an prior approval rating above 50%.

Step 7: Human-Machine Feedback

The digits for which the machine exhibited low-confidence, and are subsequently transcribed by humans, are introduced into a revised training data set for the MSVM. The revised classifier is then used on all future transcription tasks.

References

[1] D. Bradley and G. Roth. Adaptive thresholding using the integral image. *Journal of graphics, gpu, and game tools*, 12(2):13–21, 2007. 1

[2] M. Buhrmester, T. Kwang, and S. D. Gosling. Amazon's mechanical turk a new source of inexpensive, yet high-quality, data? *Perspectives on psychological science*, 6(1):3–5, 2011. 4

[3] Y. LeCun, C. Cortes, and C. J. Burges. The mnist database of handwritten digits, 1998. 4

[4] E. Mooi and M. Sarstedt. *Cluster analysis*. Springer, 2010. 1

[5] N. Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975. 3

[6] Q. Tao, G.-W. Wu, F.-Y. Wang, and J. Wang. Posterior probability support vector machines for unbalanced data. *Neural Networks, IEEE Transactions on*, 16(6):1561–1573, 2005. 4

[7] R. Van Den Boomgaard and R. Van Balen. Methods for fast morphological image transforms using bitmapped binary images. *CVGIP: Graphical Models and Image Processing*, 54(3):252–258, 1992. 3