Symbol Recognition using Average Images

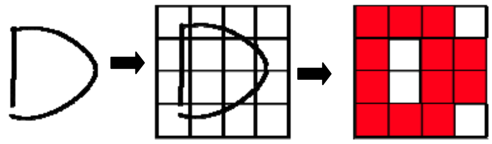
Devin Smith, Christine Alvarado, and Sarah Harris

Problem: Many symbol recognizers make limiting assumptions. For example, some assume consistent stroke order, correct orientation, or a fixed number of strokes. Such recognizers are usually very specific and generalize poorly. We aim to create a general symbol recognizer that gives users complete freedom while drawing symbols.

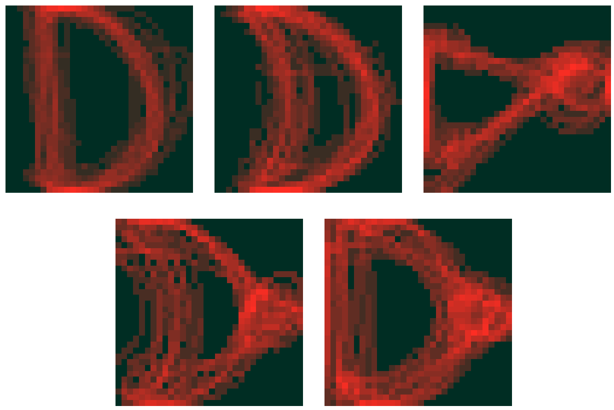
Approach: In the stroke based form, useful information is hard to extract because any number of strokes can be drawn in any order. We simplify the data by transforming the sketch into a bitmap image.

To capture the variations of each symbol, we take an aggregate collection of similar bitmap symbols and create an average image.

Sketch to Visual Transformation



Average Images (32 x 32) for And, Or, Not, Nor, and Nand Symbols



Approach Cont.: To classify new sketches, we require a distance metric between two images. We are using four distance metrics, thus guaranteeing that the combined metric never performs worse than the worst metric. The combined metric can also perform better than any single distance metric. We use the Hausdorff distance, modified Hausdorff distance, Tanimoto coefficient, and Yule coefficient metrics. For each sketch, we find the distance betweens its image and each of the average images. Thus, given N average images, we have 4N features for every sketch. These 4N features are then fed into a Support Vector Machine which is trained on labeled examples of similar symbols.

Metrics between Images A and B

Hausdorff:
$$H(A,B) = \max(h(A,B), h(B,A))$$
$$h(A,B) = \max_{a \in A}(\min_{b \in B} ||a-b||)$$

Modified:
$$H_{mod}(A,B) = \max(h_{mod}(A,B), h_{mod}(B,A))$$
$$h_{mod}(A,B) = \frac{1}{|A|}\sum_{a \in A} \min_{b \in B} ||a-b||$$

Tanimoto:
$$T(A,B) = \frac{n_{ab}}{n_a + n_b - n_{ab}}$$

Yule:
$$Y(A,B) = \frac{n_{ab}n_{00} - (n_a - n_{ab})(n_b - n_{ab})}{n_{ab}n_{00} + (n_a - n_{ab})(n_b - n_{ab})}$$

Where n_a, n_b is the number of black pixels in A, B, n_{ab} is the number of overlapping black pixels in A and B, and n_{00} is the number of overlapping white pixels in A and B.

Results: We consider recognition results to be correct if the recognizers applies the appropriate label to a new symbol. Using the cross validation technique during Support Vector Machine training, the recognizer applies the appropriate label 95% of the time. That is, 95 / 100 sketches are recognized correctly.

Future Work: We may be able to achieve better recognition results using more distance metrics. We would also like to compare this technique to the current state of the art techniques in symbol recognition. Does this technique present any advantages over a full library matching technique?

Funding: NSF Career IIS-0546809