CS771 Assignment 3

Group number: 45

Roll no.	Name	Roll no.	Name
19111055	Nancy Garala	19111080	Saurabh Pandey
19111059	Niravkumar Panchal	19111412	Shiwam Singh
19111071	Renuka Ahirwar		

1 Description of the method for DeCAPTCHA

1.1 Image Preprocessing

Firstly, we have converted each training image from RGB to HSV color space because HSV perform efficient detection by picking one of the available hues H and then adjusting the shade S (saturation or amount of gray) and brightness values V. Unlike RGB which is defined in relation to primary colors, HSV is defined in a way that is similar to how humans perceive color.

After splitting HSV image into H, S and V channels, we have applied threshold to S channel with THRESH_BINARY flag which will help in background separation. After that, we have removed background from V channel as well by performing bitwise_and(&) function between thresholded S channel and V channel.

Then we have calculated average intensity from non-zero pixels of V channel since zero pixel intensity represents background. We have applied thresholding on channel V and set pixel intensity to zero if pixel intensity < avg intensity. This thresholding will remove major part of stray lines from the image. Now, our image contains characters and some part of stray lines as noise.

As a final step of image preprocessing, we have used morphological operation called erosion to remove noise (of stray lines) present with characters and to erode away the boundaries of foreground object (characters). Hence, all lines are now removed and we have binary image in which all pixels having intensity 0 and 255 represents background and character pixels respectively.

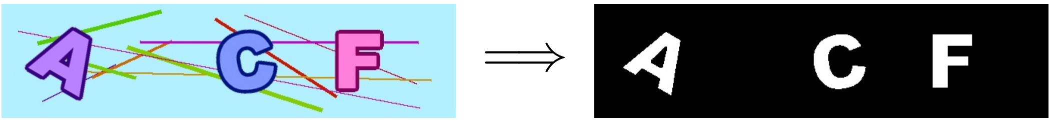

Figure 1: Conversion of original image to processed image

1.2 Character Separation

After performing steps described in section 1.1, we have observe that there is ample space between characters. We have used this fact and performed the following operation on image $\mathbf{X}$ of size $M \times N$ to create a vector v of size N.

$$v_i = \sum_{j=1}^{M} \mathbf{X}_{ji} \qquad ; \forall i \in [N]$$

Preprint. Under review.

If $v_i = 0$, we can say that it is an empty column having no part of character pixel in it.

To separate characters, we have found a range of i having continuous non-zero values. Each such range denotes starting and ending column index of a character. Number of such ranges is the number of characters in that particular image. This technique have also helped us in preserving order of characters in an image.

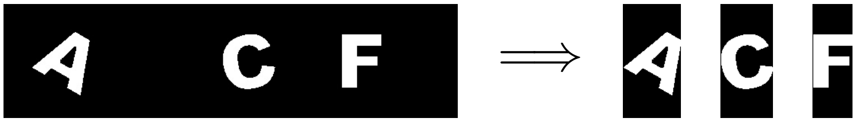

Figure 2: Character Separation

1.3 Training with kernel SVM

After character separation, we have stored characters as 20×20 grayscale images. We have splitted training data using held-out validation with 80:20 ratio and trained SVM with polynomial kernel on training set.

$$K_{poly}\left(\mathbf{x_i}, \mathbf{x_j}\right) = \left(\mathbf{x_i} \cdot \mathbf{x_j} + 1\right)^d$$

For the same, we have used sklearn.svm.SVC[1] library function with poly as kernel parameter and test the model with splitted validation set and got accuracy 1.0.

1.4 Prediction

Firstly, we preprocessed the given image as described in section 1.1,1.2. After performing this steps, we got separate character images. We used model trained in section 1.3 to predict character for each of these character images.

1.5 References

[1] Chang, Chih-Chung, and Chih-Jen Lin. "LIBSVM: A library for support vector machines." ACM transactions on intelligent systems and technology (TIST) 2.3 (2011): 27.