

# CSE 473/573 - Computer Vision and Image Processing

## Project #1 Report

### 1. Preprocessing

During enrollment and recognition, I have applied the following preprocessing steps to the images of targets and detected characters in the test images to be able to compare them.

#### Binarization

To properly extract connected components and crop the characters we need binary images. The targets images and test image, both have compression artifacts which need to be removed. To achieve this, we binarize the images. Selecting a threshold value is important as the characters have varying intensities in the test set.

To select a proper threshold for the image we need to make sure that the variance between the dark pixels and the light pixels is the highest. To achieve this, iterate over a histogram of the image to segment the pixels into two classes and calculate the variance between them using the following formula.

$$\sigma_B^2 = W_d W_l (\mu_d - \mu_l)^2$$

Where  $W_d$  = Number of pixels in dark class

$W_l$  = Number of pixels in light class

$\mu_d$  = Mean intensity of dark pixels

$\mu_l$  = Mean intensity of light pixels

By iterating over the histogram and maximizing the variance between the classes we can find the optimal threshold to split image based on intensities such that maximum information is retained in the image.

#### Cropping

The images have a lot of white space surrounding the characters, which will make comparing the characters and targets difficult. So, we can crop the characters from the images by iterating over them and finding the boundaries of the character.

#### Rescaling

To compare the characters and targets, they need to be of the same scale, so we resize them and adjust the aspect ratio of the images by scaling up both the targets and test image characters to an identical resolution. We get the pixel values of scaled images by mapping values from original image after approximating the position to be sampled from the original image.

## 2. Detection

- Binarize the image using the method described above
- Iterate over each row of the image to see if there are any black pixels to split the image into individual images of lines of characters
- Iterating over the pixels of the images to label pixels into groups based on neighboring pixels and resolving conflicts by replacing the group numbers when the neighboring pixels are a part of different groups.
- While doing the above operation we also keep track of bounding box of the group and update it every time we add a new pixel to the group.
- This results a list of bounding boxes of all the characters in the image

## 3. Recognition

- We obtain the images of individual characters using the bounding boxes calculated during detection and apply the preprocessing steps described in the first section get images of size (256,256), which are same size as the target images after the are preprocessed.
- We get the score of the difference between the targets and test characters by calculating the absolute of difference between the images. By doing this we get the number of pixels that do not match between the images.
- The character is classified based on which target has the least score. In case the least score is greater than a threshold the character is classified as “UNKNOWN”