LineArt HW

**------------------------------------------------------ Input images ------------------------------------------------------**

Symbols we use on this homework:

**'\' backslash**

**'^' caret**

**'-' dash**

**'/' forwardslash**

**'o' o**

**'|' straightline**

**'\_' underline**

As an initial input, we have 6 handmade images for each symbol. So, in total we have 6x7=42 initial images. They are all in directory **WordRecognition/symbols**

For example, for backslash, we have:

      8x16 pixels

**--------------------------------------------- Creating Train Data with Rasterization -------------------------------**

In order to create train data, we rasterize each of those 42 images 25 times.

So, as a training data we will have:

**42 initial original images + 42\*25 rasterized images = 1092 total images**

Note that each rasterization process takes place by shifting X and Y coordinates of the image pixels by randomly picked integer from the interval of [-2, 3].

So, as we now have 1092 training images for our Neural Network, it is time to represent them in binary form so that network can train on them.

**--------------------------------------------------- Encoding Train Data ---------------------------------------------------**

**Network Inputs**: for each input image, we flatten image pixels 8x16 into 128 pixels, then represent each pixel in binary form 0/1 depending whether the pixel color is white/black.

As a result we will have 128 neurons, e.g.:

[1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1]

**Network Outputs:** we represent output of network with 7 neurons (single neuron for each symbol). An example output would be;

[1 0 0 0 0 0 0] – means that network recognized input image as backslash

**Network Specifications:**

128 – 100 – 7 (128 input layer neurons, 100 hidden layer neurons, 7 output layer neurons)

Activation function: RelU

Solver: Adam

Batch size: 100

Learning rate: 0.001 (adaptive)

Iterations: 400