CS771 Introduction to Machine Learning
Assignment 3

Group No. 7
Avesh Kumar Agrawal(19111020)
Kamlesh Kumar Biloniya(160317)
Sanjeev Lal(19111077)
Shashank Kumar(19111084)

1 Description of Method used

1.1 Image Preprocessing

Following steps are involved:

(a) First we took the pixel value of the top-left corner of the image (presumably the background color) and setting all the value matching it to zero(black).

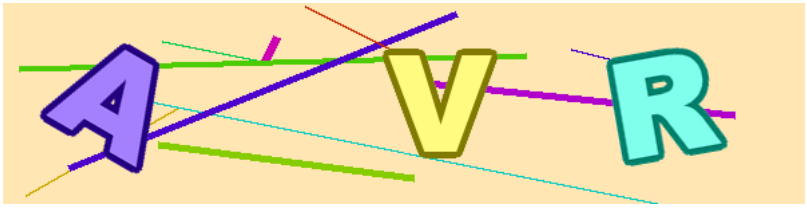

Figure 1: Orignal Image

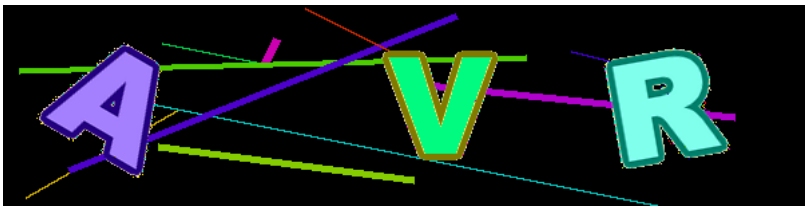

Figure 2: Changed Background

(b) Then we applied Morphological transformations[2] on the image obtained after step (a). It involves two operations: Erosion followed by Dilation.

So in Erosion, we took a kernel of size 6x6 (e.g. in 2D convolution) and slide it through the image and a pixel in the original image will be considered 1 only if all the pixels under the kernel is 1 otherwise it is considered 0. In simpler terms all the pixels near the boundary are discarded depending upon the size of the kernel.Using this we removed the small white noises and also detached the two connected objects.

Dilation is just opposite of erosion.Here a pixel element is taken '1' if atleast

one pixel under the kernel is '1'.What it does is, it increases the size of the white region in the image which helps in increasing the size of the foreground object. It is also useful in joining broken parts of an object.

Figure 3: Erosion and Dilation

(c) Converted background to white
Finding the areas of the image with the pixel value equal to zero and setting it to 255.

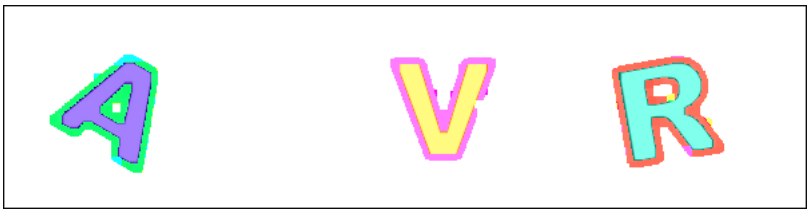

Figure 4: White Background

(d) Then converted image to grayscale

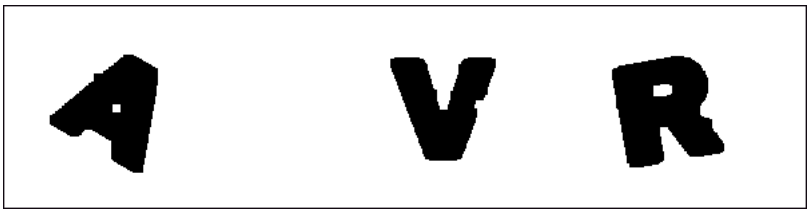

Figure 5: Grayscale Image

(e) Splitting the image
Scanning each column and check whether it contains at most 10 black pixels(or mostly white pixels) or not. If it contains mostly white pixel then we assume it is not a part of character in image so this is the basis for deciding start and end of each character in a image. Store each split character in corresponding alphabet folder for training.

Note:- We counted number of splits as number of characters in a image.

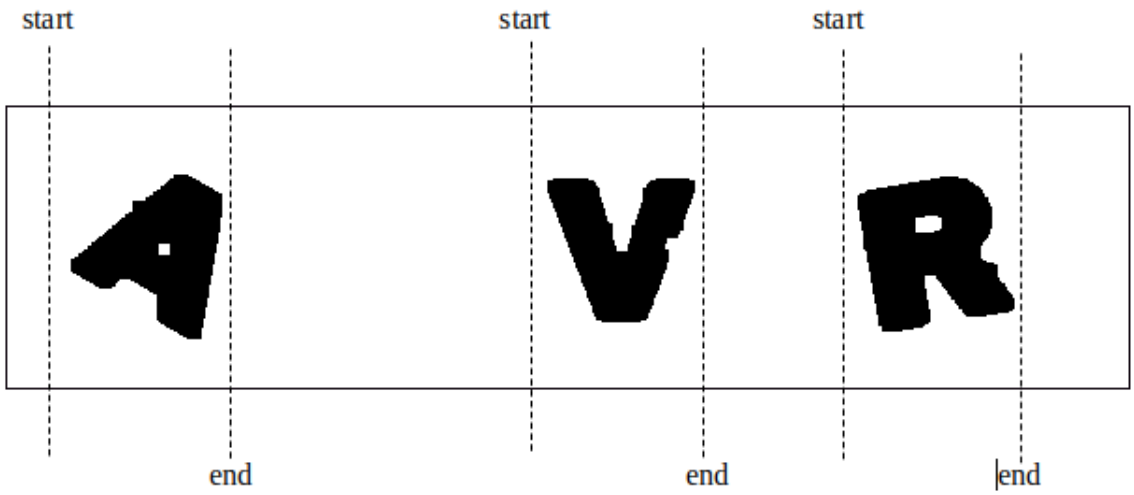

Figure 6: Cut marking in image

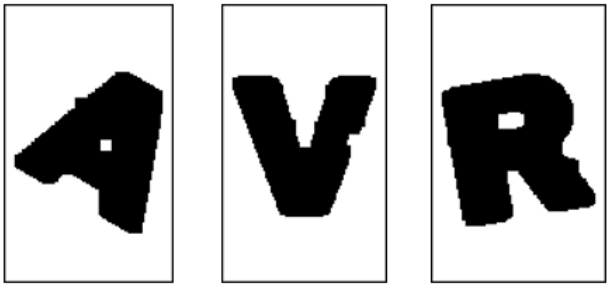

Figure 7: Split characters

1.2 Training

(a) We have trained our model using Convolution Neural Networks(CNN).
(b) Our CNN have an input layer, two hidden layers and an output layer.
(c) Input layer have 28 nodes.
(d) First hidden layer have 50 nodes and second hidden layer have 100 nodes.
(e) Output layer have 26 node each corresponding to an alphabet.
(f) Kernel size is 5*5
(g) Strides is 2*2 and pool size is also 2*2
(h) For hidden layers we used relu as an activation function and for output softmax as an activation function.
(i) We have also applied Max Pooling for reducing the dimensionality of the image.
(j) Model size is 3.4 MB

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 28, 28, 28)        728
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 14, 14, 28)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 14, 14, 50)        35050
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 7, 7, 50)          0
_________________________________________________________________
flatten (Flatten)            (None, 2450)              0
_________________________________________________________________
dense (Dense)                (None, 100)               245100
_________________________________________________________________
dense_1 (Dense)              (None, 26)                2626
=================================================================
Total params: 283,504
Trainable params: 283,504
Non-trainable params: 0
```

Figure 8: Different Layers while training with CNN

1.3 Validation

(a) First we split our data randomly as 80% training and 20% validation.
(b) Then we have trained our model over train data and validated with validation data.
(c) We did this training and validation using different values of parameters each time until we got better statistics of our model.
(d) We tuned parameters like filter size and kernel size for cnv2d layers, stride length for max pooling layers and number of nodes in hidden layer.
(e) We also validated with multiple cnv2d and max pooling layers.
(f) Different parameters used are
1. one hidden layer with 50 nodes
2. one hidden layer with 100 nodes
3. one hidden layer with 150 nodes
4. two hidden layer with 50 and 100 nodes
5. two hidden layer with 100 and 100 nodes
6. two hidden layer with 150 and 100 nodes
7. using Dropout(0.2)
8. kernel size with 4*4,5*5,6*6
but parameters written in training section gave best accuracy and optimal model size.

2 Code for DeCAPTCHA [1][2]

`https://cse.iitk.ac.in/users/aavesh/cs771/submit.zip`

References

[1] Orhan Gazi Yalcin *https://towardsdatascience.com/image-classification-in-10-minutes-with-mnist-dataset-54c35b77a38d*

[2] *https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_morphological_ops/py_morphological_ops.html*