StrikOR : PROJECT REPORT

ANKUR GURIA, 17039
ankur17@iiserb.ac.in

Objective:

Detection and Cleaning of strike-out texts in offline handwritten documents using deep learning networks.

Theory:

This project will manage the identification and processing of struck-out writings in unconstrained offline handwritten document images at word-level of three scripts viz. English, Devanagari, Bengali. Whenever kept running on an OCR, such texts will deliver drivel character-string yields. We provide a solution to this problem which requires some background as follows.

ResNet: Residual Block:

In order to solve the problem of the vanishing/exploding gradient, this architecture introduced the concept called Residual Network. In this network, we use a technique called ***skip connections***. The skip connection skips training from a few layers and connects directly to the output.

The approach behind this network is instead of layers to learn the underlying mapping, we allow the network to fit the residual mapping. So, instead of say H(x), initial mapping, let the network fit, $F(x) := H(x) - x$ which gives $H(x) := F(x) + x$. The advantage of adding this type of skip connection is because if any layer hurts the performance of architecture then it will be skipped by regularization. So, this results in training a very deep neural network without the problems caused by the vanishing/exploding gradient.

U-Net:

The main idea is to supplement a usual contracting network by successive layers, where pooling operations are replaced by **upsampling** operators. Hence these layers increase the resolution of the output. What's more, a successive convolutional layer can then learn to assemble a precise output based on this information.

One important modification in U-Net is that there are a large number of **feature channels** in the upsampling part, which allow the network to propagate context information to higher resolution layers. As a consequence, the expansive path is more or less symmetric to the contracting part, and yields a u-shaped architecture. The network only uses the valid part of each convolution without any fully connected layers. To predict the pixels in the border region of the image, the missing context is extrapolated by mirroring the input image. This tiling strategy is important to apply the network to large images since otherwise the resolution would be limited by the GPU memory.

Procedure:

The entire procedure can be briefed by the following steps: First, we had generated synthetic data. Then we shall present a combined (a) struck-out word classification and (b) an image segmentation method for identifying the strike-outs in such texts. In the case of (a), a two-class (normal vs. struck-out text) Residual Neural Network classifier will be used to detect struck-out components. In the case of (b), a U-Net-like neural network algorithm will be used to segment the strikeout from the image. Here, we propose an algorithm, trained on the data of all three scripts combined i.e. the same model has been used in all three scripts simultaneously, to detect most types of strike-outs including single, multiple, slanted, crossed, zig-zag, wavy, etc. The recognized struck-out writings will at that point be obstructed from entering the OCR. In another sort of utilization including historical documents, page images along with their annotated ground-truth are to be generated. For this situation, the strike-out strokes can be erased from the words and then fed to the OCR. For this reason, an inpainting-based cleaning approach will be utilized.

Getting into the details of each step:

Synthetic Data Generation:

- We have collected image data of English words from the IAM dataset and the same for Bengali and Devanagari had been provided by the authors of References[1] on request.
- The approximate center of each word in the image was noted manually.
- The average width of the penstroke/ pencil-stroke of each word was also noted manually.
- Total **9 types of strikeouts** including horizontal, slanted, zig-zag, wavy, criss-cross, single line, double-line, etc. were drawn through code randomly in an equal number of
  images for each strikeout.
- Finally, we have a database of **3055 images** of struck out words of **English, Bengali, and Devanagari** (same number of words of each language) having 9 types of most common strikeouts.

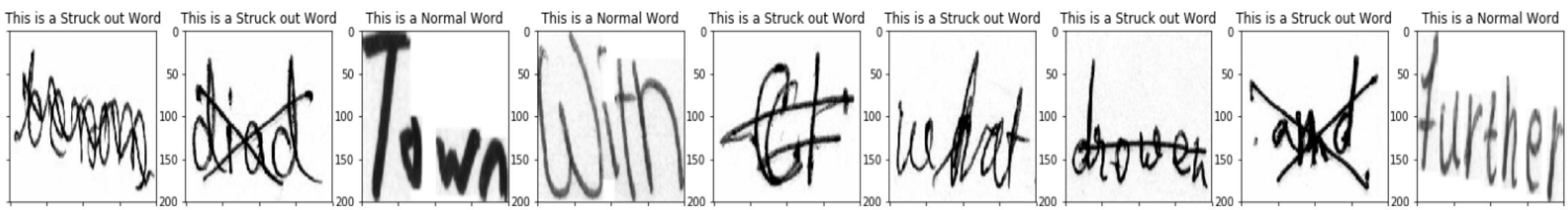
Data Pre-processing:

- We first converted the images to **GRAYSCALE** i.e. black and white as the opacity of the strikeouts drawn was quite contrasting to that of the actual word which does not happen in the real world. This step removed the above contrast of opacities.

- Then we have **normalized** the pixel values and applied them to **smooth**.
- Finally, we have applied **Gaussian blurring** which along with the smoothing removed the edgy sides of the strikeouts (as it was drawn through code pixel-wise) and made it look more realistic (thus the pixel values too).

Two-class classification:

As the first step of the problem, we have trained the original **ResNet** to classify the struck-out word images and the non-struck-out words.



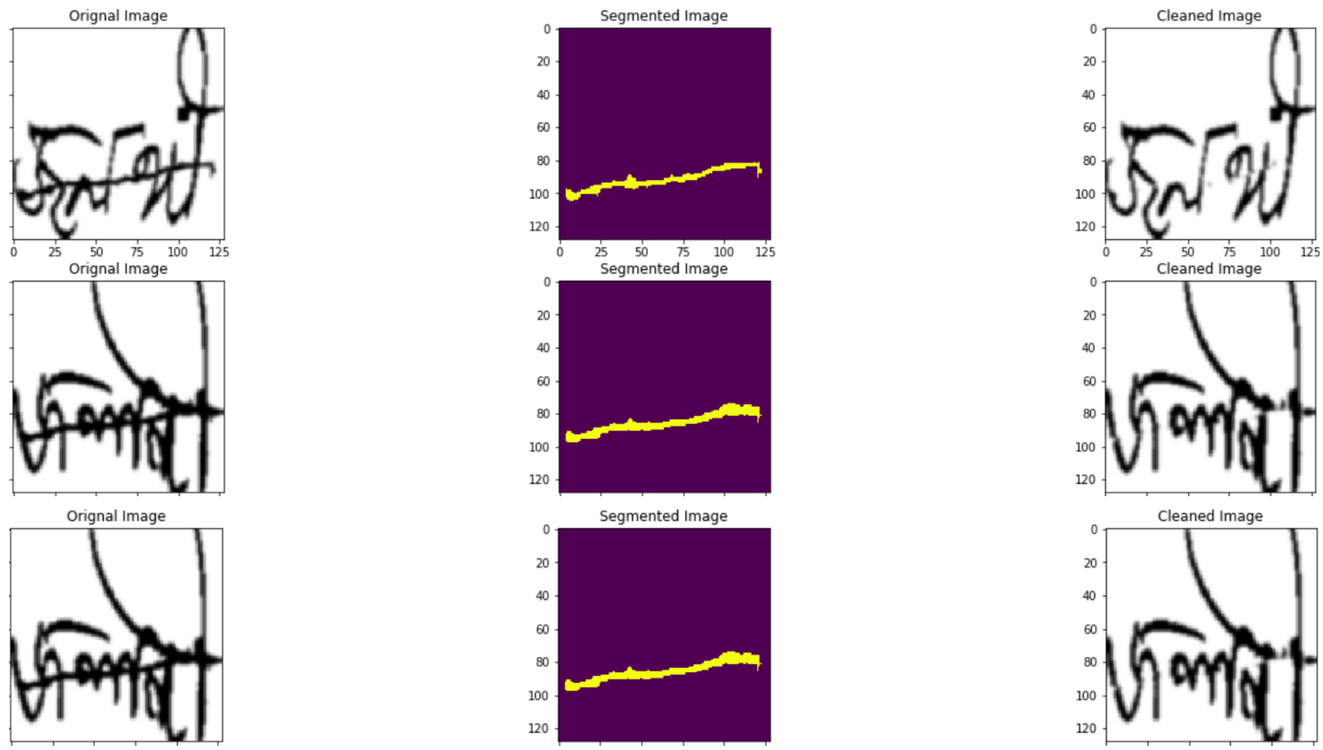
Strike-out segmentation:

- For the next step, we have tried to segment the strike-out from the struck-out word to decipher the original word using a **U-Net** architecture (as it is the current SOTA algorithm for image segmentation).
- We have tried the following things to improve its initial performance:
    - Several loss functions
    - Several activation functions
    - Regularization techniques
        - Dropout
            - On random layers
            - Just before the last layer
            - After every layer (except the last layer)
            - Before every layer (except the first layer)
        - Early Stopping
    - Data Augmentation (clockwise and counterclockwise 90-degree rotation)

The results and motivations of these experiments have been discussed in the observations section of the report. We observed results using metrics like F1score, pixel-wise accuracy, the mean intersection of the union, and dice loss.

Retrieving the original word:

We used the inpainting-based approach to erase the segmented strike-outs from their corresponding images to get the original word image.



Observations:

In the step for the two-class classification, without any tweaking in the ResNet architecture, we have achieved an accuracy of 98.7% in the classification of struck-out words from non-struck-out words.
In the next part to segment the struck out we faced **overfitting** with the original U-Net, where we got very good accuracy on the training data but the model performed very poorly on the test data. To overcome this problem we tried hyperparameter tuning where we found the combination of the "**Adam**" optimizer, "**Binary Cross Entropy**" loss function, and the "**Relu**" activation function to work the best. Plotting the loss function against the epochs we noticed that the loss function rather than forming a convex function i.e it hasn't nullified at a point but on a plan. This meant the gradient much before the last epoch had nullified and the model isn't learning i.e the issue of **vanishing gradient**. The most sought after way to deal with this issue is "**dropout**". We tried adding dropout layers to various places of the U-Net architecture based on some past research works regarding the same. We found out that adding a dropout layer before the second resizing layer in each hidden layer gave the best results.
Data Augmentation by 20% and **early stopping** further improved the results to stand at an F1 score of 96.2%.

Conclusion:

Solving this problem is very essential for an OCR to perform properly but unfortunately, not much work has been done for the same. The outcome of the experiments carried out regarding this problem in this report gives us an idea that the hypothesis algorithm could be a probable solution to this problem.

References:

1. B.B. Chaudhuri, C. Adak, “An Approach for Detecting and Cleaning of Struck-out Handwritten Text”, Pattern Recognition, vol.61, pp.282-294, January 2017.
2. Olaf Ronneberger, Philipp Fischer, Thomas Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation”, May 2015.
3. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, “Deep Residual Learning for Image Recognition”, December 2015.