Deep Learning
CS898BD
Zachary Pearson
D439H842
Assignment 3

Introduction

This question had us build an encoder-decoder model to train to denoise images.

Methodology

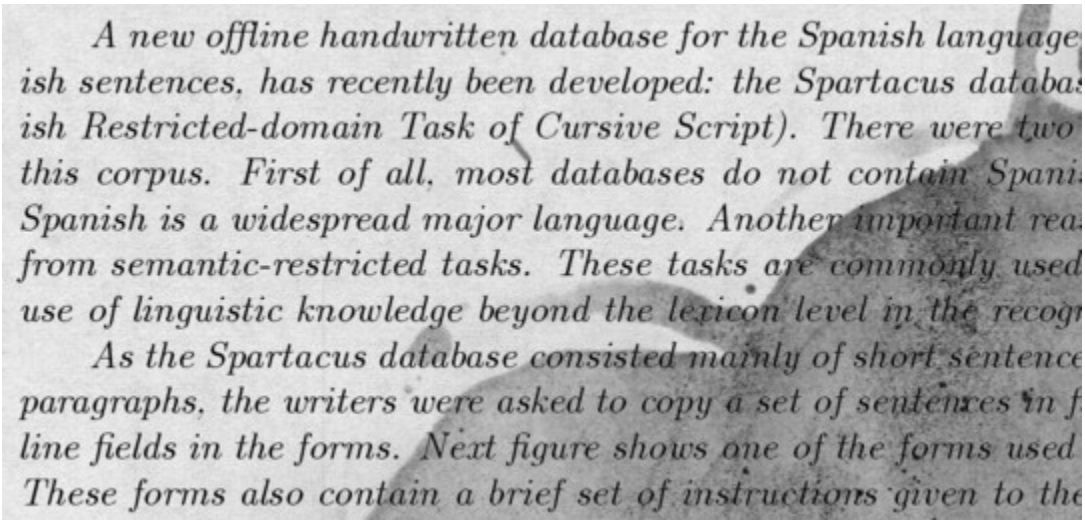
144 images were used to train. Each image had a dirty (noisy) and clean version. 74 images are used to test. Results show original image and image after applying the encoder-decoder to it.

Architecture

The encoder is built of 2 2d convolutional layers using strides of 2 to down sample the image. First convolutional layer has 16 filters and the second layer has 8, both with kernel size of 3. The decoder was built with 2 2d transpose convolution layers, upsampling the image using a stride of 2. the first layer had 8 filters and the second had 16. Both used a kernel of 3. A 3rd 2d convolutional layer with a single filter to reconstitute image. SSIM was used to measure loss. Softplus was the activation function.

Prior to this architecture, a seperate architecture containing 6 convolutional layers with 3 maxpool downsampling between sets of 2 convolutional layers for the encoder and 6 transpose convolutional layers for the decoder, however, this architecture failed to ever learn and would not produce a recognizable image so those results were not included here.

Results



Original Test Image

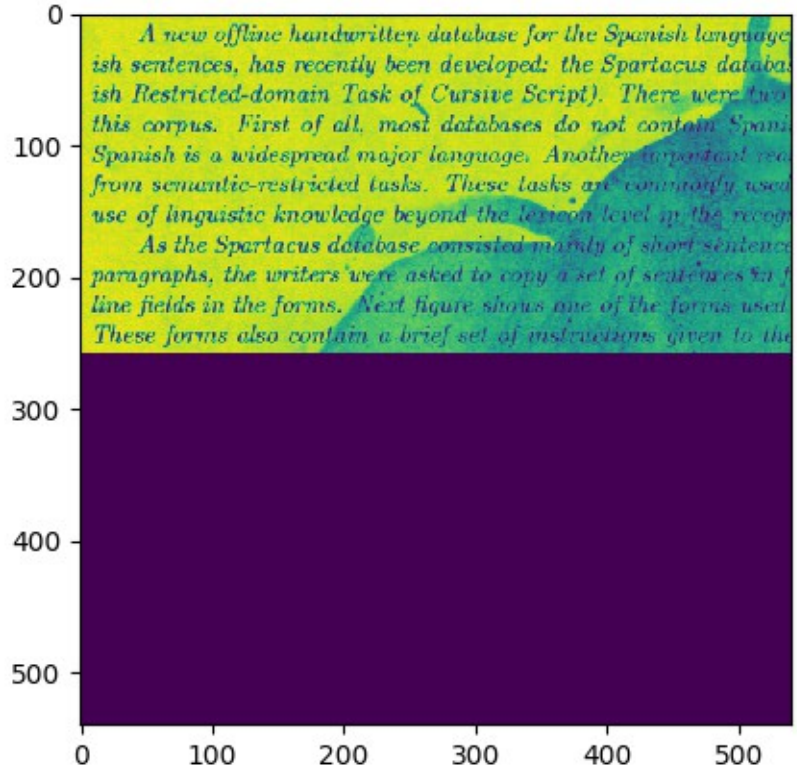
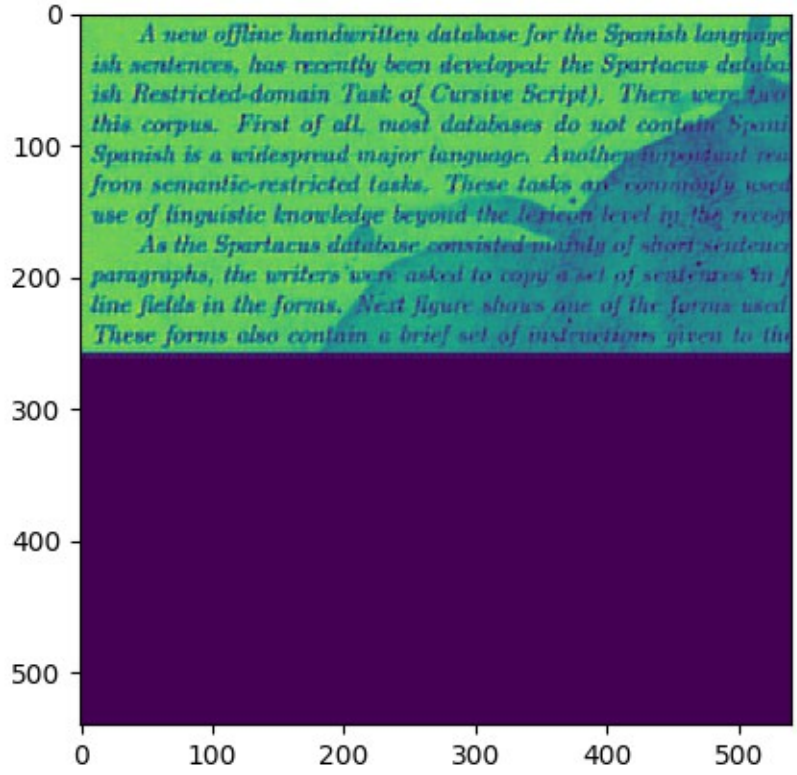
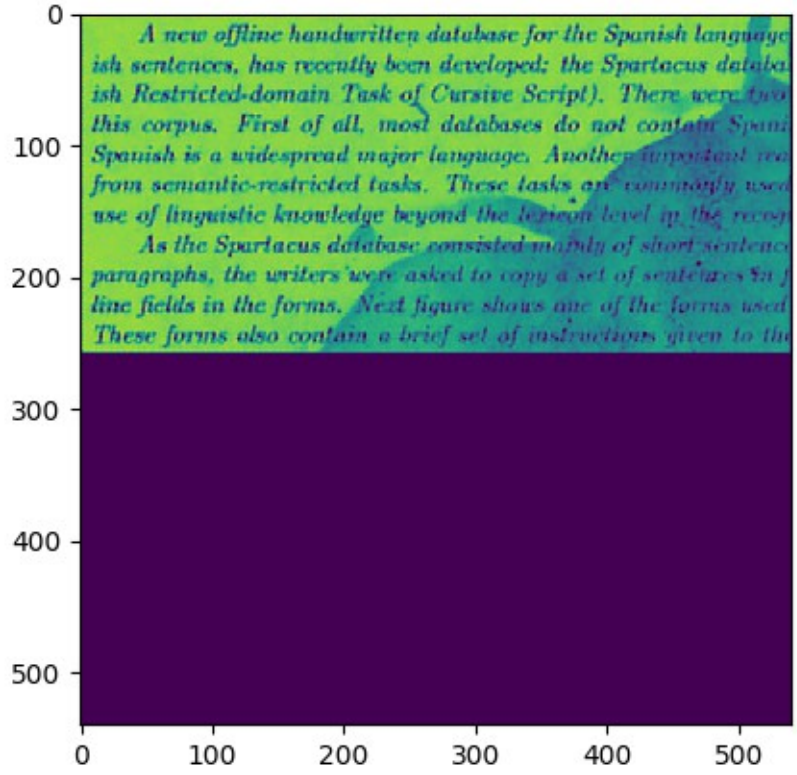


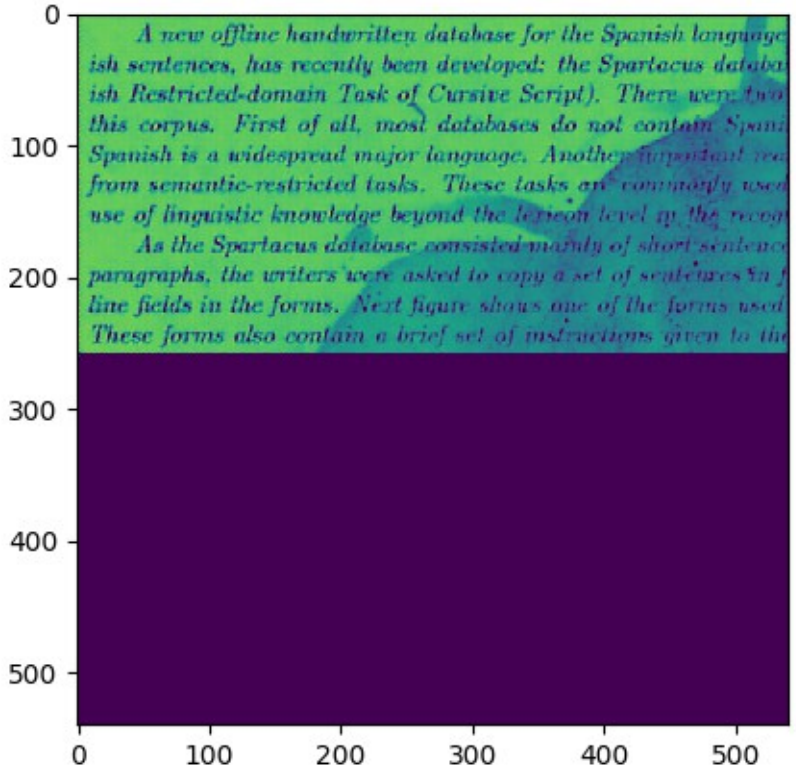
Image after initial processing



Denoised after 10 training epochs



Denoising after 25 epochs



Denoising after 50 epochs

Conclusion

It appears that the model attempted to denoise by reducing the contrast between the stained and unstained areas. Additional training epochs did not appear to improve performance of denoising. The model was particularly sensitive to initial weights as multiple runs could potentially encounter an exploding gradient.

References

https://www.tensorflow.org/tutorials/generative/autoencoder#second_example_image_denoising
https://stackoverflow.com/questions/57357146/use-ssim-loss-function-with-keras