An Attempt at First Letters

Technical
Report

Maktoob

There is nothing redeeming in succeeding once and failing a thousand times because failure was always the most probable outcome.

Omar Emad Mohamed

December 29, 2024

Contents

1 Introduction

This report is about sharing all of what I have learned while pursuing the first letters prizes for scrolls 2, 3 and 4.

The main goal was trying to incorporate scroll 1 into the training data, simply because I believed that ink should be captured by the model regardless of the scroll texture, later on I noticed that ink is a lot more complicated than just simple cracks, and there seems to be some kind of bias in the data that makes the models capture a specific quality of ink. Perhaps the most frustrating thing was the fact that regardless of the architecture, the models managed to learn "something", of course to varying degrees but still, most of the models I tried managed to perform well on scroll 1.

And I tried quite the variety of architectures, not including my variants of these models or the ones that failed to train:

- TimeSformer (From Grand Prize)
- RCAN [1]
- HSDT [2]
- UNet_CT_Multi_Att_DSV_3D [3]
- MIRNet [4]
- QRNN3D [5]
- ConvLSTM U-Net [6]
- VidelFocalNet [7]
- TransformerUNet (Current best) [8]
- ResNet plus Decoder (From Grand Prize)

My variants had small changes such as activation functions, loss functions, pooling layers, training data type (to be discussed later), number of layers, optimizers, data augmentations (to be discussed later), model parameters, and/or an extra model in parallel or in sequence; later on I started mixing and matching from multiple papers hoping to create the best of all the things I tried.

2 Methodology

Well, other than sheer grit, my setup had a T4 and sometimes an L4 GPU from **Google Colab**, I fixed the segments which are saved on Google Drive for training, I believed that I should be treating this like research chemistry, very small amounts of each scroll and if something worked I would switch into production chemistry gears and swap to the L4 GPU and pour in as much training data as possible to test an idea to its very limits.

Logging all the experiments on **wandb**, currently 2300+ training hours logged and 1815 models trained, and saving the predictions on my local because **wandb** provided only 5Gb of free space and right now all the predictions take more than 20Gb.

3 Initial Attempts

My very first experiment was with the TimeSformer and some manual labels on Scroll 2, back then I thought surely if you try enough times we can nail the amount of ink in the examples labeled and something would show up somewhere on some segment, that was quite wishful thinking, spent 3 months doing that, nothing of significance showed up, I saw so many model predictions I started hallucinating letters in real life, everything I saw seemed like paper with writing, right there, and then I realized this is not the path forward.

Started treating the TimeSformer as a black-box and switched my experiments to scroll 1 purely, varied the labels, made quite the variety of label types, for example, gradient labels with a gradient brush, skeleton-less labels, fuzzy labels created by posterizing the gradient brush and RL labels (that was a stupid but funny idea). Regardless of the shape, thickness, and color of the labels, the model learned something, So the labels are not a significant variable, they just have to be in the right place. With that variable out of the window, we can move on.

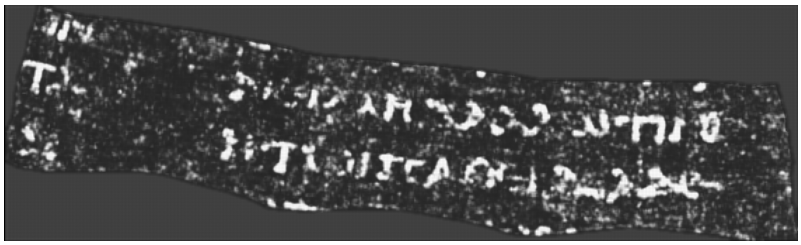
I then started attempting to ruin the GP model checkpoint by finetuning it on manual labels from scroll 2, this was super fun, because I learned that the GP model managed to somewhat overfit the ink cracks. So, to the GP model ink is almost detected purely by its cracks, it takes 3 epochs, 500 steps with a batch-size of 95 to

almost ruin the predictions of the GP model on some of scroll 1 segments.

I then switched my attention to layer composition, and this is when things started to become interesting.

4 New Layers?

After testing a few stupid ideas like weighted average layers, average of layers, difference of Sobels, and Sobel layers. I wanted to mend the gaps in the layers by taking the average of 3 or 5 layers and using that to train the model, this slightly enhances the GP model output and reveals partial letters.



(a) S1_20231004222109 Original GP Output

(b) S1_20231004222109 With Average Layers

Figure 1: Effect of Averaging Layers

What performed unexpectedly well was something entirely absurd. Inspired by how the TimeSformer treats the layers like a video and some video compression techniques, I wanted to target another quality of ink, specifically ink dispersion or at least this is what I think it is. I created a small example to explain this one. So, with a marker and some tissues, I wrote the letter "W" on the stacked tissues. I noticed that ink starts to fade at the last layer, expected behavior, but what was interesting is that I noticed that we can take the difference of layers to target another quality of ink, I knew that taking the difference would sacrifice a lot of information and this method would at best train an ink estimator, but beggars are not choosers, I dare say. I was willing to sacrifice a large portion of the information in the layers to gain generality. Later, I switched to the average of the normal layers and the difference of layers.

What I mean exactly by the difference of layers is:

$$L_j = |L_i - L_{i+1}| \ \forall i \in [0, 63] \tag{1}$$

(a) L1

(b) L2

(c) L3

Figure 2: Stacked Tissues Example (Normal Layers)

(a) New L1

(b) New L2

Figure 3: Stacked Tissues Example (Difference of Layers)

And the predictions of that experiment were interesting because the model was trained purely on scroll 1. For example, this was the output for **20240516205750** from scroll 2. I like to invert the predictions, lower the brightness, and enhance the contrast just for visual contrast:

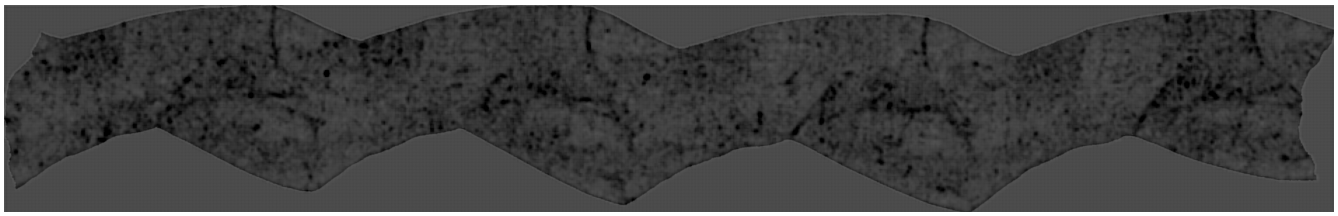


Figure 4: Difference TimeSformer Prediction on S2_20240516205750

But the TimeSformer struggled with this type of data, it failed to complete the letters in Scroll 1, quite expected but I thought we could do better, back then, I thought that perhaps we can try another architecture or architectures. And I started my lovely learning journey with nothing but my trusty T4 GPU, I tried anything and everything I found interesting on **Paperswithcode**. I leaned towards UNets because they had an interesting quality: larger batch-size which meant faster training which translated into more experiments, but they underperformed most of the time, until I noticed that the necessary component for a UNet architecture to work well was the addition of an attention mechanism.

Very recently, I noticed that the issue is perhaps not necessarily the model architecture, because I trained quite a few and they managed to learn what is ink on scroll 1 but whenever I tried to use data from multiple scrolls, I noticed that in order to learn something from any scroll other than 1 the models tended to "unlearn" whatever there is to be learned in scroll 1, at that moment, I shifted my focus towards the idea of harsher data augmentations, I wanted to make it as hard as possible for the models to learn from scroll 1 segments.

5 Augmentations

I believed that the temporal cutout augmentation from the grand prize submission was not enough. So, I created quite a few more, a bit harsh, but I deemed them later necessary because some models failed to adapt to these augmentations. Furthermore, the harder we make the pattern, the more confident we are that the model is not hallucinating. The issue while pursuing the first letters was never machine hallucination but simply my hallucination, hope is one hell of a drug.

5.1 Confused Tango!

Imagine with me here, a lovely ball somewhere in the world, where all the couples are dancing tango but they are not in sync at all but rather complete pandemonium, everyone is stepping on each other's toes. Now hold that image in mind.

Instead of taking **n** layers, we take **N** layers and, we use **n** layers where **N**>**n**. and we randomly start "Dancing" in those **N** layers, the idea here is to remove any layer dependency that could be learned by the model. This augment solved an issue I had to deal with when predicting on new segments, which layers should we predict on? Since we are loading more layers, the detection range effectively becomes the loaded range and not the training range, and that was indeed the case, although the ink signal is weaker but detectable.

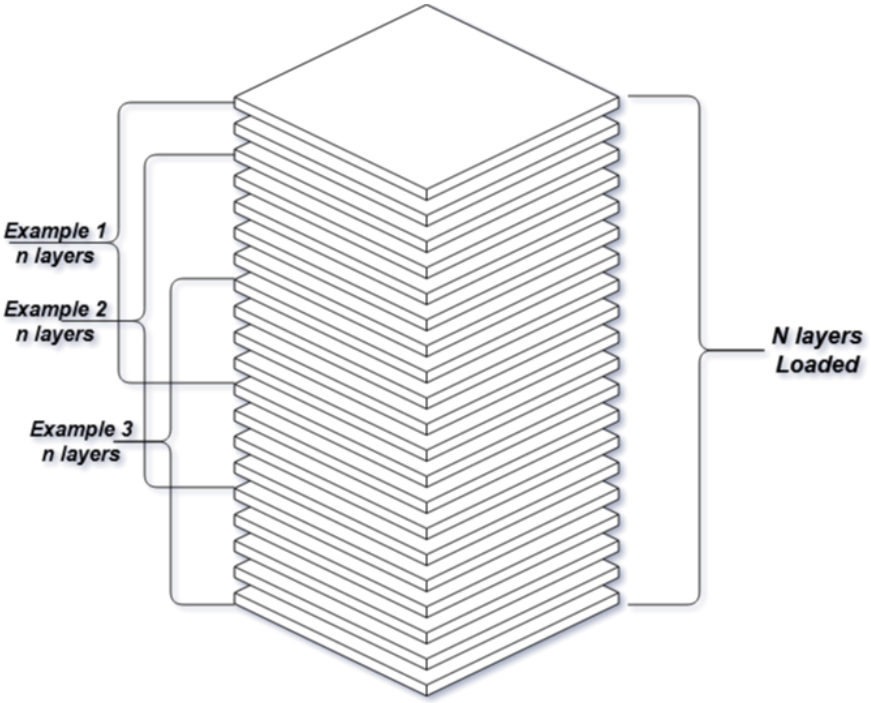


Figure 5: Example of Confused Tango Augment

5.2 Ah, ink! But what if?

So, I had another problem when predicting on a segment from a scroll other than 1, I was not sure if we have to reverse the layers for the model or not, so what if we use that as an augment? After all, detecting ink should be a bidirectional task. The model should not depend on the direction from which it is inspecting the segment.

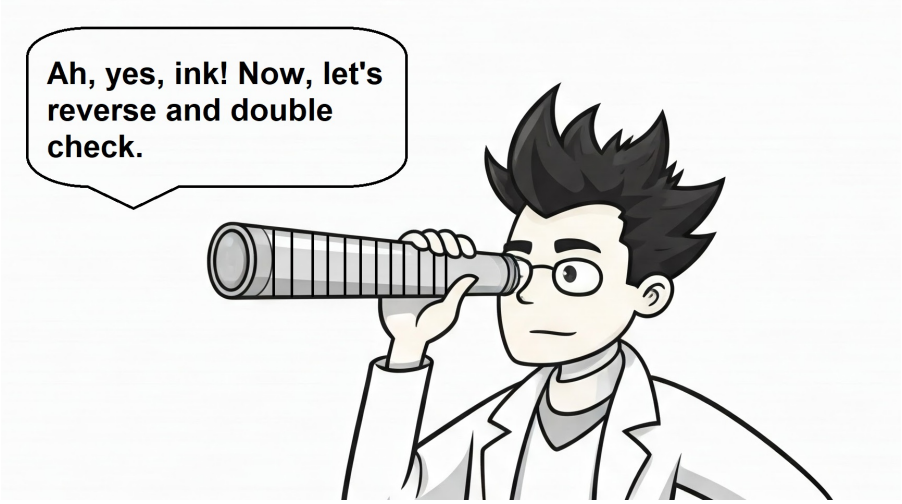


Figure 6: Ah, ink! But what if?

5.3 Good Ol' Morphology

Being a classical computer vision fanatic, I wanted to use erosions and dilations just because I love them, so when I was testing some stupid ideas while simulating scroll textures, for example, Gaussian blurs to mimic scroll-2, spatial cutouts and lower brightness for scroll-4 and erosions for scroll-3. At first, I thought, this is going too far, but with the erosions and dilations, I added another difficulty, and that is making the **CoarseDropout** augment even harder. Surprisingly, the models managed to overcome this one. I still think it is ridiculous.

6 Making it even harder!

Another very maddening observation is that scrolls 2, 3 and 4 do have something that looks like the ink cracks present in scroll 1, what happens is that, whenever we train with scroll 1 the predictions are always anything that resembles cracks and that should not be the case because scroll 2 segments have this very smooth texture and if we assume that there is ink in these segments then surely ink is no longer cracks at all, but cracks are nothing but mere by-products of ink and not ink. Hence, the issue became feature smoothing. An interesting solution I reached after reading [9] is adding Gaussian blurs inside the encoder (I like to call that smoothed convolutions) and the deeper the encoder, the heavier the blur, but that was not enough to solve this persistent issue. I had to use **FeatureAlphaDropout** inside the encoder to remove any dependency on specific encoder channels. Given how difficult training became, I had to adjust the label smoothing factor from **0.25** to **0.33**. With that, I managed to train a model on data from all five scrolls and still predict some letters on an unseen segment from scroll 1 and this "unlearning" phenomenon did not occur in one of my attempts. Here are four segments from all the scrolls for that experiment (inverted and with enhanced contrast).

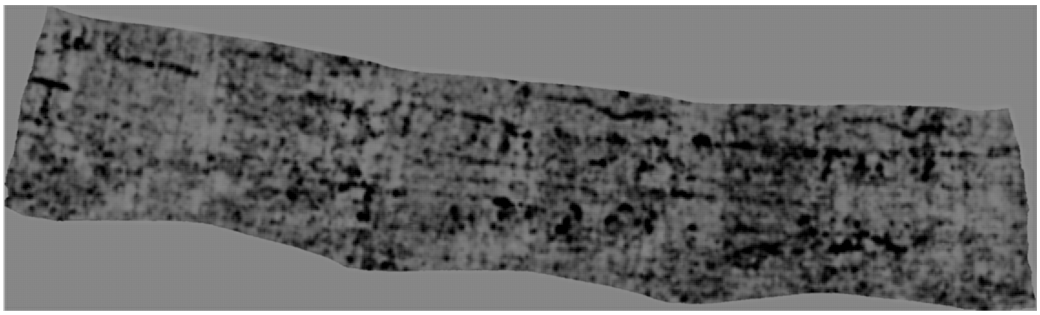


Figure 7: Unseen S1_20231004222109

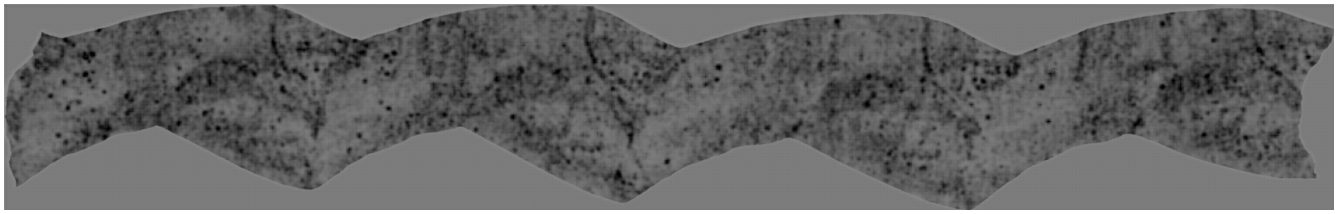


Figure 8: S2_20240516205750

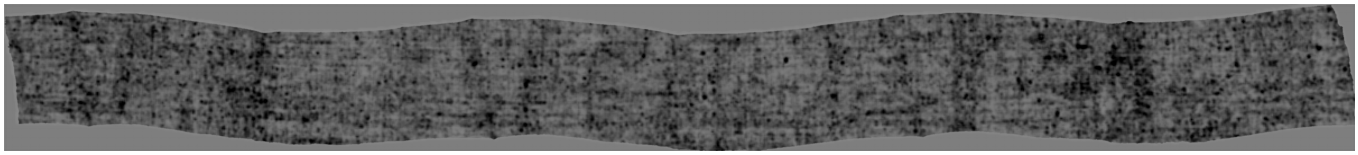


Figure 9: S3_20240716140050

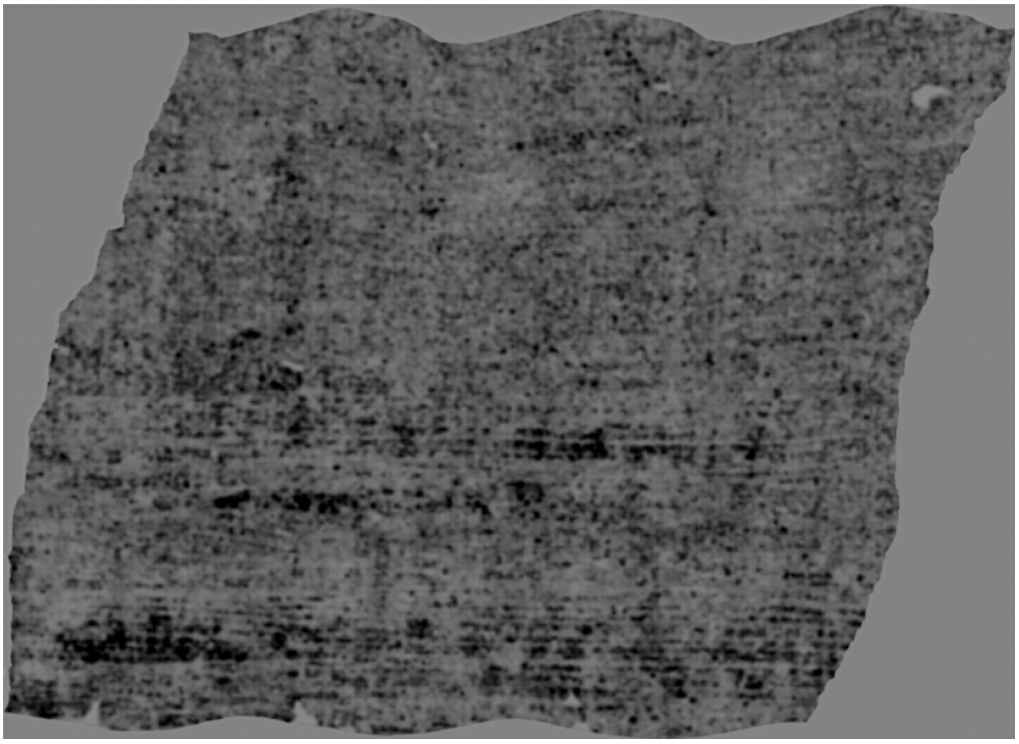


Figure 10: S4_20231210132040

All of the model predictions now have what looks like lines of writing, but they are too vague and very unclear. I suspected that my manual labeling is a bit too conservative, so I repeated that experiment again but with thresholded predictions (worst kind of labels), just to increase the number of examples for the model where scroll-1 and scroll-5 take around 25% of the data now. Once again we can still detect some ink on an unseen scroll-1 segment.

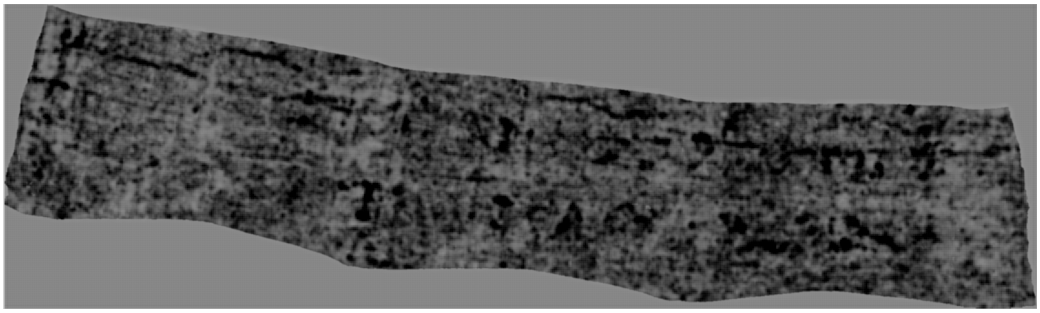


Figure 11: Unseen S1_20231004222109

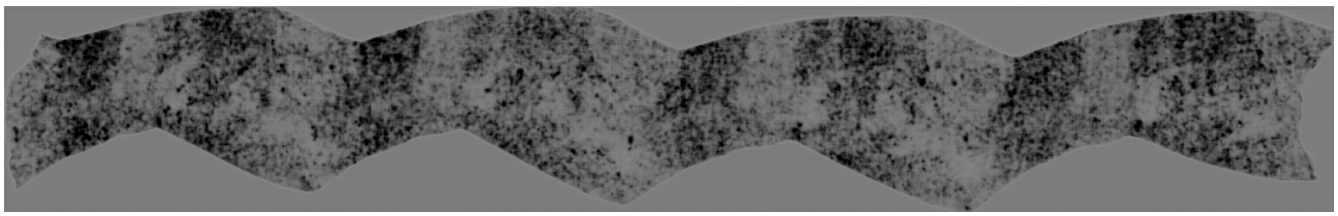


Figure 12: S2_20240516205750

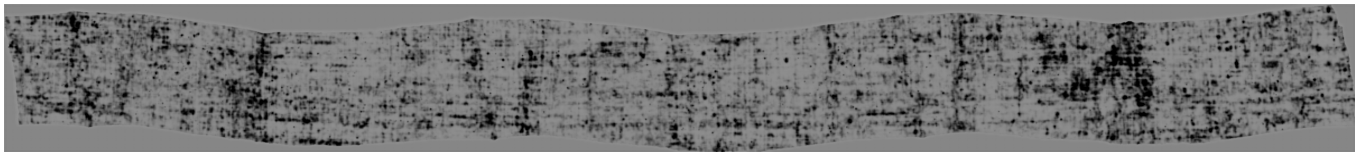


Figure 13: S3_20240716140050

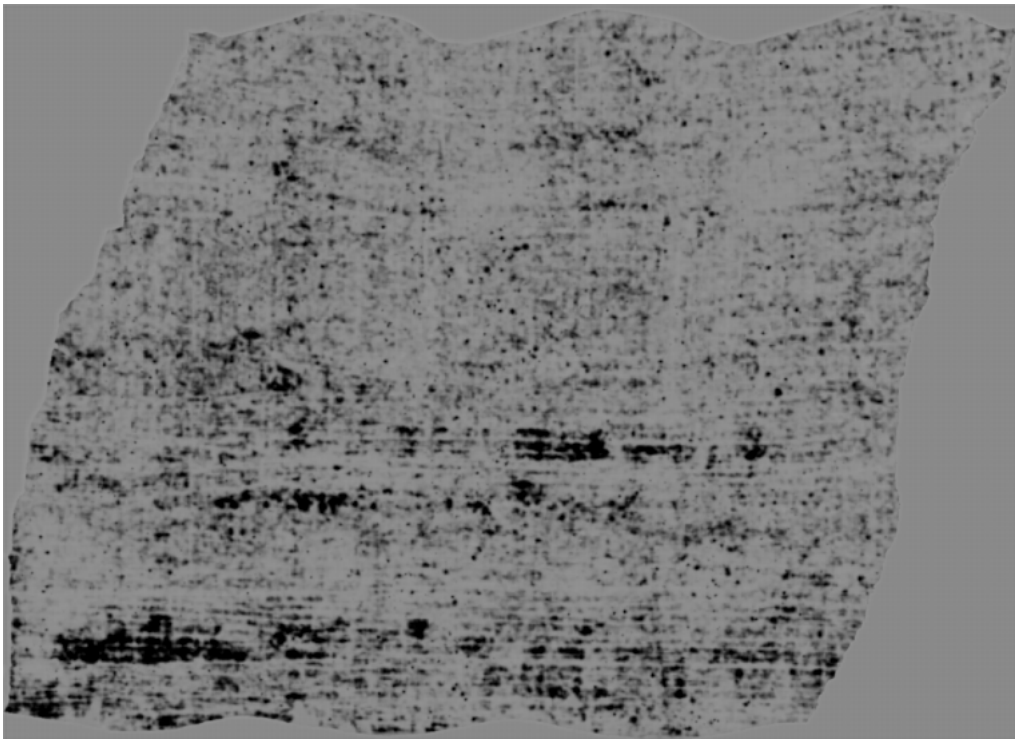


Figure 14: S4_20231210132040

7 Conclusion

In conclusion, Unets can handle harder and harsher augments, are significantly faster to train, and are very flexible as we can easily add feature abstraction layers to the encoder. Furthermore, layer composition seems to greatly affect performance while label shape, color, and thickness are not significant at all. Additionally, predicting on the average of layers reduces the noise and mends the gaps in the individual layers, prompting the prediction of partial letters. Lastly, although a purely experimental finding, the best losses were **SoftBCEWithLogitsLoss** and **DiceLoss**.

I like to think that perhaps my approach just needs more training data, like, way more, but I think that is just my cognitive dissonance kicking in to save whatever is left of my sanity. I don't regret a single second spent working non-stop on this, I was just playing around and having fun. I hope that this work will be the ember of some fire.

Being a fresh graduate, I used to finish all my reports with memes.

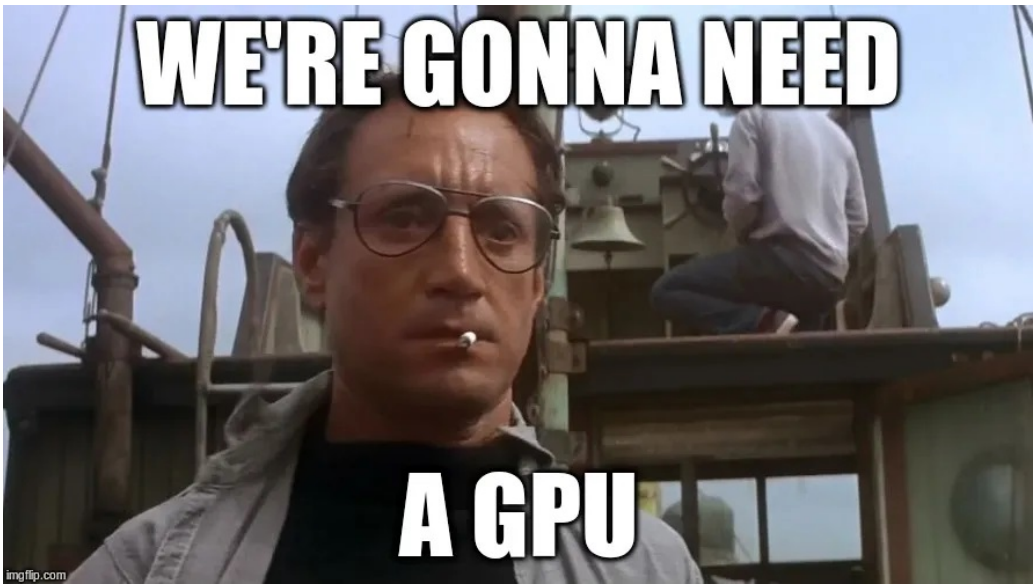


Figure 15: Old Habits Die Hard

References

[1] J. Chen, H. Sasaki, H. Lai, *et al.*, "Three-dimensional residual channel attention networks denoise and sharpen fluorescence microscopy image volumes," Aug. 2020. DOI: 10.1101/2020.08.27.270439. [Online]. Available: http://dx.doi.org/10.1101/2020.08.27.270439.

[2] Z. Lai, C. Yan, and Y. Fu, *Hybrid spectral denoising transformer with guided attention*, 2023. DOI: 10.48550/ARXIV.2303.09040. [Online]. Available: https://arxiv.org/abs/2303.09040.

[3] O. Oktay, J. Schlemper, L. L. Folgoc, *et al.*, *Attention u-net: Learning where to look for the pancreas*, 2018. DOI: 10.48550/ARXIV.1804.03999. [Online]. Available: https://arxiv.org/abs/1804.03999.

[4] S. W. Zamir, A. Arora, S. Khan, *et al.*, *Learning enriched features for real image restoration and enhancement*, 2020. DOI: 10.48550/ARXIV.2003.06792. [Online]. Available: https://arxiv.org/abs/2003.06792.

[5] K. Wei, Y. Fu, and H. Huang, *3d quasi-recurrent neural network for hyperspectral image denoising*, 2020. DOI: 10.48550/ARXIV.2003.04547. [Online]. Available: https://arxiv.org/abs/2003.04547.

[6] R. Azad, M. Asadi-Aghbolaghi, M. Fathy, and S. Escalera, *Bi-directional convlstm u-net with densley connected convolutions*, 2019. DOI: 10.48550/ARXIV.1909.00166. [Online]. Available: https://arxiv.org/abs/1909.00166.

[7] S. T. Wasim, M. U. Khattak, M. Naseer, S. H. Khan, M. Shah, and F. S. Khan, "Video-focalnets: Spatio-temporal focal modulation for video action recognition," *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13 732–13 743, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:259847742.

[8] O. Petit, N. Thome, C. Rambour, and L. Soler, *U-net transformer: Self and cross attention for medical image segmentation*, 2021. DOI: 10.48550/ARXIV.2103.06104. [Online]. Available: https://arxiv.org/abs/2103.06104.

[9] N. Park and S. Kim, *Blurs behave like ensembles: Spatial smoothings to improve accuracy, uncertainty, and robustness*, 2021. DOI: 10.48550/ARXIV.2105.12639. [Online]. Available: https://arxiv.org/abs/2105.12639.