Project Report - ECE 285

Ben Zhang
Electrical and Computer Engineering
A16268103

Abstract

This report presents the reimplementation of Pathak et al.'s Context Encoders for restoring damaged historical manuscript pages. We first motivate the need for automated document restoration and outline our modified context encoder–decoder architecture featuring single-channel grayscale input, a channel-wise fully connected layer for global context, and convolutional decoding of missing regions. We then describe experiments on the DIVA-HisDB medieval manuscript dataset using different masking strategies, and evaluation via PSNR as mentioned in context encoder against a diffusion-based inpainting baseline.

1 Introduction

Historic documents often suffer physical damage—such as tears, stains, and fading ink—that obscures both text and images, making manual restoration time-consuming and requiring specialized skills. Automating this process with deep learning models holds the potential to speed up digital preservation and enhance accessibility. We approach document restoration as a context-based inpainting challenge: given a scanned page with one or more masked areas, we aim to predict visually plausible content that similar to the page's structural layout, including text lines, margins, decorations, and stroke style. Our method reworks the Context Encoder model proposed by Pathak et al. (2016), tailoring it for single-channel grayscale inputs and adding a channel-wise fully connected layer to capture the global context of the page. Initial results indicate that our model can reconstruct missing areas with notable characteristics—achieving PSNRs that recreates similar results from original context encoder paper.

2 Related Work

2.1 Context Encoders for Image Inpainting

Pathak et al. (2016) developed a deep-learning-based inpainting by framing it as a context-completion task. Their model employs an encoder–decoder convolutional architecture: the encoder consists of five convolutional layers. Where it begin with a 5x5 filter that produces 64 feature maps and doubling up to 512 maps via 3x3 filters, mixed with BatchNorm and ReLU activations, and reduces spatial resolution by using convolutions.

The decoder mirrors this structure with four transposed convolutions that up-sample back to the input resolution and ends with a single-channel sigmoid output to reconstruct elements in the missing region. Training combines an reconstruction loss that applied to the missing part to encourage accurate average intensities, with an adversarial loss from a discriminator that examines the patch and its surroundings to push for sharper, more realistic textures. In our adaptation, we convert all inputs to single-channel grayscale to match manuscript scans, remove the adversarial term in favor of stability on our smaller dataset since we only looking to reconstruct words, and insert a channel-wise fully connected layer between the encoder and decoder to better capture the global page layout and context as paper did.

2.2 DIVA-HisDB Dataset

DIVA-HisDB by Simistira et al. (2016) is a focused benchmark that features 150 high-resolution pages from medieval manuscripts, sourced from three different codices and annotated at both line and region levels.

Although it was initially designed for layout analysis, the quality of its scans makes it suitable for pixel-level restoration tasks. We extract non-overlapping 256×256 grayscale patches, producing patches from each manuscript page, and normalize these by scaling pixel values to the range [0,1] and subtracting the mean of each patch. The dataset allocate different patches for training, validation, and the a public-test group for verification.

During training and evaluation, we employ three types of masks, including centered square, random square, and irregular free-form in real time to simulate damage like tears, stains, and ink loss.

The network is trained to reconstruct only the masked pixels using the surrounding context, with early stopping determined by validation PSNR.

2.3 Diffusion-Based Document Inpainting with Character Perceptual Loss

To establish a baseline, we apply a diffusion model by Yang et al. (2024) that is specifically designed for document scans. Their method couples a biharmonic diffusion solver—used to propagate known pixels into holes—with a Character Perceptual Loss (CPLoss) that enforces both global style and fine stroke fidelity inside masked regions.

Concretely, they employ a pretrained VGG network Simonyan and Zisserman (2014) to extract multi-scale features $\mathrm{VGG}_i(x_r)$ from the repaired image x_r and $\mathrm{VGG}_i(x_{\mathrm{target}})$ from the ground-truth x_{target}, then penalize their misalignment only within the damaged mask x_m:

$$\mathcal{L}_{CP} = \sum_{i=1}^{L} \omega_i \left\| \mathrm{VGG}_i(x_r) \; - \; \mathrm{VGG}_i(x_{\mathrm{target}}) \right\| \; \odot \; x_m.$$

Here, ω_i weights the contribution of each VGG layer, and x_m ensures the loss concentrates on character regions. This perceptual guidance yields sharper, more legible stroke reconstructions than diffusion alone. We adopt their biharmonic inpainting implementation as our baseline for comparison through their provided GitHub.

3 Method

Our inpainting pipeline is implemented in PyTorch and consists of three main components: data preparation with randomized masks, a Context Encoder and Decoder network augmented by a channel-wise fully connected layer, and training and validation routines that optimize mean square error loss and measure PSNR similar to the context encoder by Pathak et al. (2016).

1. **Data Preparation.** We define a custom Dataset that:
   - Loads each manuscript image as a single-channel (256×256) patch.
   - Generates a binary mask M of the same size according to one of three strategies: centered square, random square, or irregular free-form "tear" lines. Endpoints are sampled in $[0, 255]$ and indices clipped to valid bounds.
   - Applies the mask at runtime by zeroing pixels in the hole.
2. **Network Architecture.**
   - Encoder: Four convolutional blocks map $1 \to 64 \to 128 \to 256 \to 512$ channels, each using 3×3 kernels (5×5 for the first), stride-2 downsampling, BatchNorm, and ReLU.
   - Channel-Wise FC: A learnable weight per channel scales the entire 512-channel feature map, enabling global context re-weighting similar to the context generatin paper.
   - Decoder: Four transposed-convolutional blocks mirror the encoder ($512 \to 256 \to 128 \to 64 \to 1$ channels) with stride-2 upsampling, BatchNorm, ReLU, and a final Sigmoid to predict intensities inside the hole.

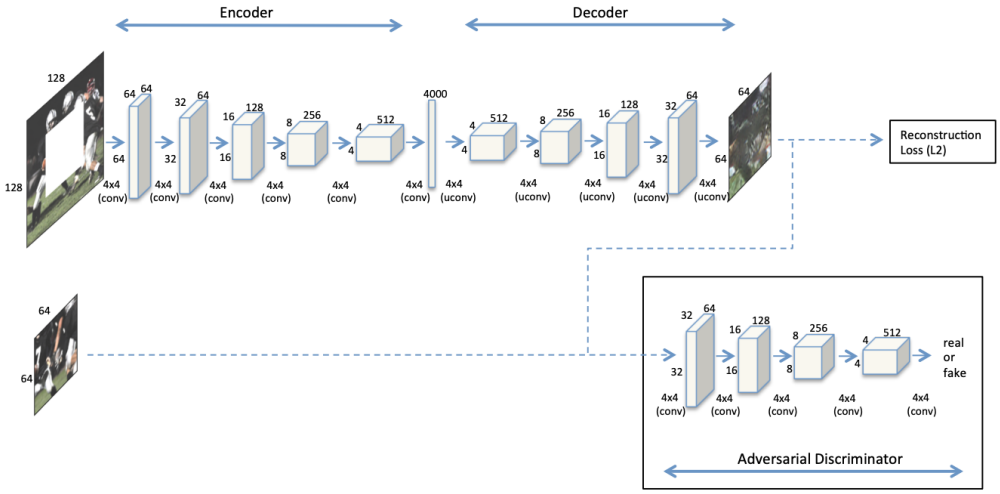


Figure 1: Context Encoder Training model based on Pathak et al. (2016) for Center Region Dropout

- Output Composition: The decoder output is masked so only missing pixels are returned, then composited with visible pixels for visualization and metric computation.

Overall the structure is exact replica of Context Generation Model provided.

3. **Training & Validation.**
   - Loss: Mean-square-error we learned for course computed only over masked pixels using MSE loss function for pytorch, where M and N are representing height and width, and X and Y are respective pixel between ground truth and prediction:

$$MSE = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(x_{ij} - y_{ij})^2$$

   - Optimizer: Adam with learning rate 1×10^{-4}.
   - Epoch: We are training only 50 Eepochs since the dataset is small to prevent overfitting.

4 Experiments

We evaluate our model on the CB55 manuscript in DIVA-HisDB where it split into three folders (`img/training`, `img/validation`, `img/public-test`). All experiments use 256×256 patches with irregular masks unless otherwise specified.

4.1 Model Structure

1. **Setup & Hyperparameters.**
   - *Patch Size:* 256
   - *Batch Size:* 16
   - *Learning Rate:* 1×10^{-4}
   - *Epochs:* 50
2. **Training & Validation.**
   - Based on the file directory given by DIVA-HisDB, we train on 70% of patches from training, validate on 15% in validation, and rest 15% for verification test cases.
   - Early stopping or model selection is based on maximum plateau of PSNR increment detected.

3. **Metrics.** PSNR is a matrix computed only over masked pixels of each validation patch that was used in the context generation paper, where it measures the ratio between the maximum possible pixel value of an image and the mean-squared error between a reference image and a reconstructed image.

$$\mathrm{PSNR} = 10\log_{10}\Big(\tfrac{L^2}{\mathrm{MSE}}\Big), \quad \mathrm{MSE} = \frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W}\big(I_{i,j} - \hat{I}_{i,j}\big)^2$$

where

- L is the maximum pixel value.
- H and W are image height and width.
- I is respective pixel between reference and reconstructed pictures

For the model, we simply called the MSE method as a torch function that is implemented as nn.MSELoss.

4. **Layers**
   - **Masking & Input:** Zeroes out the missing region as a defined function to mask it so that all subsequent convolutions only process valid pixels.
   - **Conv5×5 to 64 channels:** Extracts low-level features while preserving spatial resolution.
   - **Downsampling Conv3×3 blocks (64, 128, 256, 512 channels):** Three blocks halve spatial size and double feature depth to aggregate increasingly global context.
   - **Channel-Wise Fully-Connect Layer:** Applies a learned scalar weight to each of the 512 feature maps, allowing the network to globally re-emphasize or suppress encoded context before decoding.
   - **Upsampling ConvTranspose3×3 blocks (512, 256, 128, 64 channels):** Three transposed convolutions mirror the encoder, gradually restoring spatial resolution and reconstructing structures.
   - **Final Conv3×3 to 1 + Sigmoid:** Predicts pixel intensities in the masked hole. The output is then composited with the original image so only the missing region is replaced.

4.2 Baseline Plan

We evaluated inpainting baselines on the DiffHDR pipeline by Yang et al. (2024) through cloning the repository provided by the author, installing its dependencies, patching environment variables in huggingface, importing and loading the pretrained UNet checkpoint, and performing inference on each validation patch with classifier. We plan to measure PSNR on the hole region for direct comparison. Shown in Figure 2.

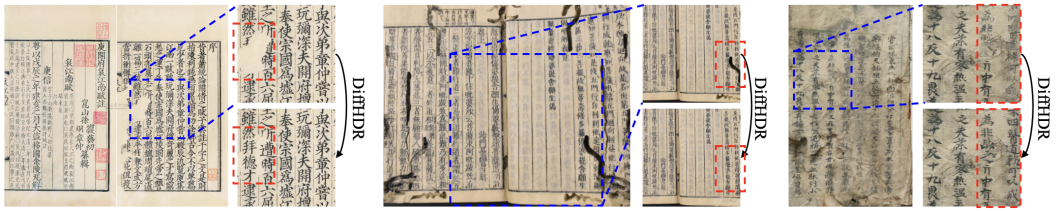
...

5 Result

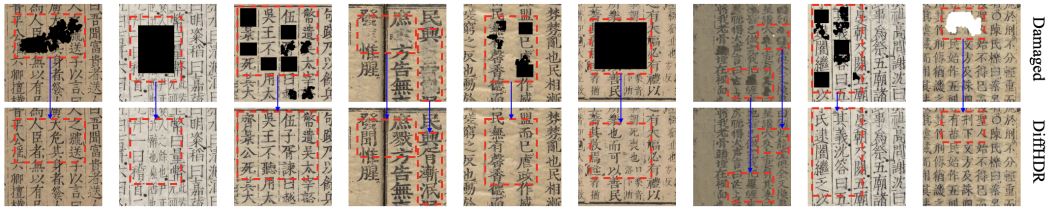
After training, we infer on the first three images in `img/public-test`, displaying side-by-side panels of: original, masked input, and reconstructed output. The Figure 4 shows the first 3 verification image and its reconstructed picture.

5.1 Training Result Analysis

Over the course of 50 epochs, the training loss steadily declines from about 0.012 to 0.0025, indicating that the network is effectively fitting the masked-patch regression objective. However, both the validation loss and PSNR show significant fluctuations rather than a smooth trend—validation loss spikes as high as 0.034 around epoch 14 and dips as low as 0.0026 by epoch 49. At the same time, PSNR fluctuates between 0.74 and 1.87. These fluctuations occur for several reasons. First, our irregular multi-block masking strategy exposes the model to vastly different contextual challenges in each batch, where some holes completely cover dense text. In contrast, others only include blank margins, so performance on a fixed validation split varies. Second, the small size and limited diversity

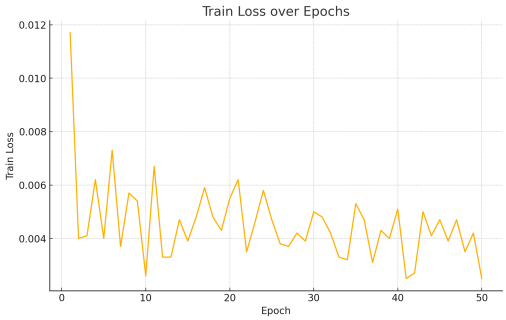


(a) Real Damaged Historical Document Repair by DiffHDR

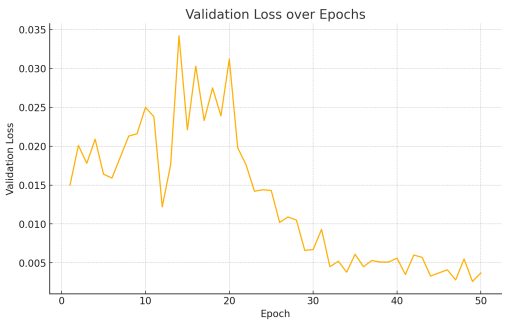


(b) Unseen Damaged Historical Document Repair by DiffHDR(Types of the documents are not encountered during training)

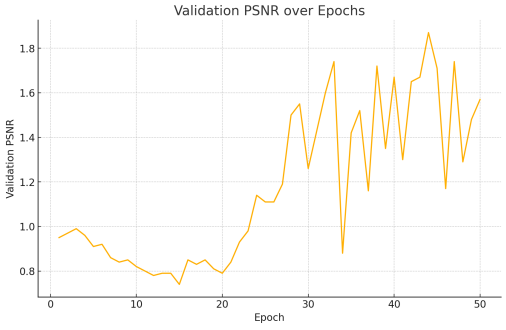
Figure 2: Predicting the Original Appearance of Damaged Historical Documents Yang et al., 2024



(a) Train Loss over Training Epoch



(b) Validation MSE Loss over Training Epoch



(c) Validation PSNR over Training Epoch

Figure 3: (a) Baseline restoration result; (b) Training loss; (c) Validation MSE; (d) Validation PSNR.

of the DIVA-HisDB's CB55 patch set means that the model can overfit specific patterns in one epoch and lose them when faced with new mask configurations. Finally, optimizing solely with an MSE loss biases the model toward local gray-level averaging rather than stable stroke reconstruction, resulting in abrupt PSNR swings due to small changes in the data distribution. Hence, these factors explain why, despite an overall increase trend in PSNR and a decrease trend in loss, the curves remain noisy rather than monotonic, explaining fluctuation shown in Figure 3.

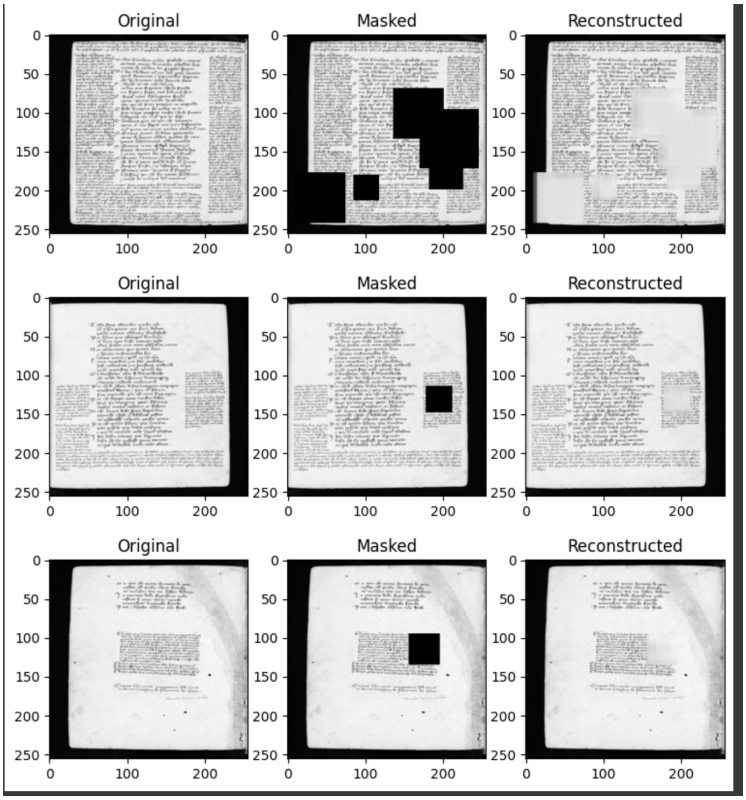


Figure 4: Result

5.2 Result Analysis

The reconstruction results illustrated in Figure 4 show that our Context-Encoder can roughly restore the general layout and gray-level continuity of missing areas, particularly in sections with a relatively uniform background. In the top row, large masked sections covering dense text lead to a smeared reconstruction: the network captures the page's margins and overall texture, but individual characters remain unclear, and edges appear softened. The middle examples feature smaller and more isolated gaps. It demonstrated improved recovery of stroke width and line orientation, suggesting that with enough surrounding context, the model can discern local text patterns. In the bottom row, where the hole is located in a largely blank area, the inpainting is decent, with a smooth blend into the parchment background. Overall, the network is proficient at fitting the picture with coarse background structures but struggles to generate high-frequency details, such as handwritten characters, often reverting to locally averaged textures instead. This is expected for the context generation model as it finds the average around the missing piece.

5.3 Baseline Issue

Our attempt to benchmark against the HDR diffusion baseline encountered an issue: the inference script requires specific versions of Diffusers and HuggingFace Hub, which no longer provide symbols like $cached\_download$ or HF_HUB_DISABLE_TELEMETRY. We have tried to resolve this issue by editing the code or downgrading the environment library, but it still didn't fix the problem. In summary, the baseline failed mainly due to version mismatches and missing constants in huggingface hub.

6 Conclusion

In this work, we reimplemented a Context-Encoder architecture for document inpainting and evaluated it on medieval manuscript patches from the DIVA-HisDB collection. Our single-channel encoder and

decoder, featuring a channel-wise fully connected layer to capture global context, learned to restore missing regions by minimizing MSE loss. Quantitatively, the training loss steadily decreased to around 0.0025, while the validation PSNR gradually increased to approximately 1.6 dB; however, both metrics displayed significant oscillations due to the variability of masking patterns and the model's tendency to average pixel intensities instead of reconstructing fine stroke details. Qualitatively, the network effectively propagated coarse background textures and page layouts but struggled to produce ancient characters or sharp edges within heavily masked text.

In our future work, we aim to enhance our loss function by incorporating various components to promote more defined stroke details. We will explore the use of partial convolutions with different attention modules to improve the transmission of information across irregular hole boundaries. Additionally, we plan to expand our training data with synthetic masking to represent stains, tears, and a variety of handwriting styles, allowing the model to learn a broader range of damage patterns. We also intend to extend beyond the DIVA-HisDB corpus by incorporating additional medieval codices to enhance font and layout diversity. Overall, I believe these efforts can help improve the model to effectively restore relics.

References

Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., & Efros, A. A. (2016). Context encoders: Feature learning by inpainting. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2536–2544. https://doi.org/https://doi.org/10.48550/arXiv.1604.07379

Simistira, F., Seuret, M., Eichenberger, N., Garz, A., Liwicki, M., & Ingold, R. (2016). Diva-hisdb: A precisely annotated large dataset of challenging medieval manuscripts. *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 471–476. https://doi.org/10.1109/ICFHR.2016.0093

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*. https://arxiv.org/abs/1409.1556

Yang, K., Lyu, P., Liu, Y., Xu, M., & Lyu, M. R. (2024). Predicting the original appearance of damaged historical documents. *arXiv preprint arXiv:2412.11634*. https://arxiv.org/abs/2412.11634