Unstructured Data - Coursework 4
Autumn 2024

CID: 2153480

Exploring Advanced Imaging techniques for reversing facial distortions

GitHub Repo: My Repo, Including the dataset

Problem Statement, Topic Background and Overview

In 2007, law enforcement agencies faced a significant challenge in identifying Christopher Neil, a sex offender who had evaded capture by obscuring his face in illicit photographs using a digital swirl effect. Despite extensive efforts over three years, authorities were unable to reverse the distortion until they discovered the specific software used, which enabled them to reconstruct the original images. This case underscores the complexities involved in counteracting such image manipulations.

Motivated by this, I aim to investigate whether contemporary deep learning techniques can effectively reverse swirl distortions without prior knowledge of the distortion parameters or software used to make the distortions. The primary objective of this project is to explore advanced deep learning methodologies for reversing swirl distortions in images, particularly when the distortion parameters are unknown.

To address this problem I will choose to experiment with a technique we have not covered so far in the course, called a Distortion Field U-net. The Distortion Field U-Net Ronneberger et al. (2015) is a type of CNN which employs an encoder-decoder structure with skip connections to produce a dense flow field in a single pass. The model learns the pixel displacements required to reverse the swirl effect. It encodes high-level contextual information while simultaneously preserving fine spatial details, this architecture allows each pixel's displacement to be learned without relying on predefined parameters or knowledge the specific software that created the distortion. Once the displacement map is generated, the network warps the distorted image back to its original form, circumventing the need for manual insights into how the swirl was imposed.

I chose to use a synthetic dataset to train my model, since the technique is new and complex, and there were no publically available datasets which contained swirled imaged with unswirled counterparts. Also, by using a synthetic dataset, I can precisely control the variation in swirl distortions while ensuring that we can measure the performance of our model against original images affectively. We can then compare the reconstructed images with their original unswirled images using similarity metrics to assess the performance of the model. To evaluate model performance, I had to pick loss metrics which measured similarities between the reconstructed images and the originals. I decided on three metrics, which I will outline below:

1. Smoothness loss. A regularization term which ensures that the distortion field predicted

by the model varies smoothly across the image. This is critical for preventing abrupt, unrealistic changes in the transformation field which can look odd in the output. The loss is calculated by penalizing differences in adjacent pixel values within the distortion field. By encouraging spatial smoothness, this loss promotes plausible and coherent deformations in the reconstructed images.

2. Structural Similarity Index Measure (SSIM). Introduced by Wang et al. in 2004, SSIM loss models the human visual system's perception of image quality by penalizing differences in luminance, contrast, and structural information between images. Unlike traditional pixel-wise losses, SSIM loss focuses on preserving perceptual quality, with lower values indicating better similarity to the original image.

3. Perceptual Loss. This metric leverages a pre-trained convolutional neural network, called VGG, to evaluate differences between images in a high-level feature space rather than at the pixel level. This metric captures more semantic and perceptual differences, focusing on how the images would be perceived by a human rather than comparing exact pixel values. This is especially importnant for this task since visual coherence and high-level features matter more than exact pixel-wise similarity.

In this way I will evaluate models based on both 'obective'/ mathematical similarity as measured by PSNR, and on 'perceptive' similarity as measured by SSIM.

Following the evaluation of my model synthetic data, the final model will be applied to original family photographs that have been intentionally swirled. This will serve to assess the practical applicability of the techniques I have explored in real-world scenarios, where images may exhibit unique characteristics not present in the training data.

By systematically exploring these advanced correction methods, the project aims to contribute to the field of image restoration, offering potential solutions for forensic investigations and other applications where reversing image distortions is critical.

Dataset details

To facilitate this study, I have curated a synthetic dataset by applying swirl effects to a subset of images from the EasyPortrait dataset Alexander Kapitanov (2024). This data set comprises of high-resolution portrait images with clearly visible faces. In total, I downloaded 10,000 images from this dataset which I then used to create my synthetic dataset.

Reversing distortions like swirls is inherently challenging due to their non-linear and spatially complex nature. Swirls involve intricate pixel displacements governed by non-linear functions, and there is no explicit mathematical inversion that can directly reverse these transformations without prior knowledge of the parameters and function used. This complexity is further exacerbated when dealing with real-world images, where distortions may overlap with diverse backgrounds, textures, and noise. I therefore decided to create a synthetic dataset with controlled swirl parameters, so that I was able to isolate the distortion effects and focus on the core task of unswirling, rather than dealing with the additional complexity of very unclear faces or lots of background objects and noise in my data.

The EasyPortrait dataset is publicly accessible and widely used in computer vision research, ensuring compliance with data usage permissions. By creating a controlled synthetic environment with known distortions, I can systematically assess the performance of various correction techniques and train a model to specifically learn how to reconstruct swirled images. Because of the complex nature of this task, and the advanced imaging techniques I am using, a fairly simple dataset which deals with standardised images and distortion parameters, clear faces and will help me to train a functional model for specific use cases, which can then be built upon for more complex images and distortions.

To generate my synthetic dataset, I applied swirl transformations from the sci-kit image ’swirl’ function to the images to create distorted counterparts for training and evaluation. The swirl parameters, including rotation strength and radius, were randomly sampled from specified , but relatively small ranges: strength values ranged from 8 to 15, and radius values ranged from 300 to 500 pixels. The swirl was also held constant at the centre of each image, so that the mask would more effectively learn just the area of the swirl. This ensured a diverse set of swirl patterns while maintaining consistency in distortion intensity. All images were resized to 128×128 pixels to standardize input dimensions and reduce computational complexity. Before training, the dataset was normalized using the calculated mean and standard deviation of the pixel values, aiding convergence and improving model generalization. By carefully controlling the distribution of swirl parameters and preprocessing steps, I ensured the dataset was diverse yet structured, enabling the model to effectively learn and reverse the distortions.

An example of a (Swirled, Unswirled) image pair from my synthetic dataset is given in figure 1 below:

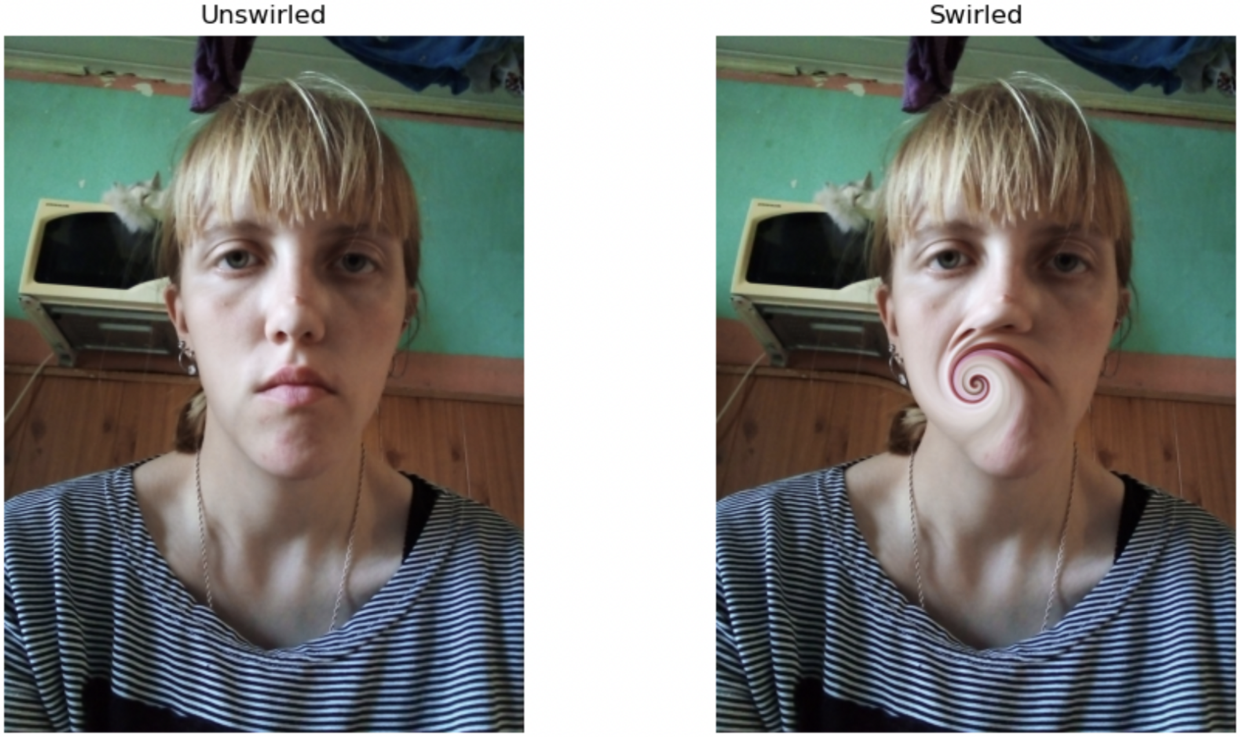

Figure 1: A randomn pair of images from my synthetic dataset

Due to the multiple loops in my dataset creation, on my local machine with no GPU, this dataset took 1h41mins to generate.

Description and justifications of methods and analysis

For my model, I chose a U-Net architecture enhanced with self-attention and a distortion field prediction output, rather than a regular CNN as studied on the course. Although a regular CNN would have been easier to understand and implement, I believe that a U-Net is better suited to my task, given the complicated nature of my image transformation. Regular CNNs are designed to learn hierarchical features from data which make them better suited to tasks like classification or segmentation. However, they are not inherently designed to model spatial distortions or to directly handle pixel-wise mappings or geometric transformations, which was ultimately what I wanted my model to predict. My U-Net design incorporates components that explicitly model spatial relationships, and for this task, the swirl introduces a structured distortion, where pixel displacements are governed by specific parameters (radius and strength). Fundamentally, we are trying to predict some unknown function in our state space which will transform our swirled image back to the original image given to the model. The architectural design of my U-Net enables this, since it includes mechanisms like skip connections, multiscale feature extraction, and deformable convolutions. These components enable the model to capture both local and global spatial dependencies, which are essential for reconstructing the original image. By using skip connections in the U-Net architecture, I was able to help the model preserve fine-grained spatial information that may be lost in deeper layers of the network. The encoder-decoder structure of a U-Net helps us in learning representations at multiple levels of abstraction, enabling the network to understand both the overall pattern of the distortion and pixel-level corrections. In addition, the direct prediction by the model of a distortion field allowed me to easily visualize the results, as for this task looking at arbitrary loss numbers will not allow me to analyze how good the task might actually be in practice at solving the problem at hand.

Another crucial decision in my methodology was chosing an appropriate loss function with which to train and evaluate my model. I chose to create a customer loss function, combining reconstruction loss (SSIM), smoothness loss, and perceptual loss. SSIM measures the structural similarity between the reconstructed and original unswirled images, while smoothness loss encourages a realistic and smooth distortion field. For Perceptual loss, I utilized a pretrained model called VGG16. This model helps the loss function to focuses on the visual similarity between the images in terms of facial features, making it highly suited for my task. This custom loss function helped me to balance different aspects of the unswirling task, leading to better results.

For training, I used standard deep learning practices for training, using the Adam optimizer and a ReduceLROnPlateau scheduler to reduce the learning rate if needed. I also included patience in the model, to halt training if the model did not continue learning, and 100 epochs to ensure the model was able to learn sufficiently with this complex task.

Initially, after building the model , I first overfit it to a single image just to check that it was able to learn any kind of distortion function. I discovered that, by printing gradients at each stage of the UNet, my model was unable to leave the trivial solution of returning to me the original swirled image. This is because, outside the radius of the swirl, the 'swirled' and 'unswirled' images were pixel-wise identical, and so any pixel-wise transformation as applied by the model would initially be applied to many or all of the pixels, meaning the loss would get bigger initially while moving away from the trivial solution. Though I experimented with leaky ReLU activations, gradient clipping, and dynamic hyperparameter tuning, I was not able to resolve this issue initially. I then introduced a 'mask' on my model - meaning that my UNet model would only operate within a set radius of the center of the images given to it, leaving all other pixels unchanged. By applying the mask, the model avoided applying transformations to areas outside the swirl radius, which reduced unnecessary distortions and helped focus the learning process on meaningful regions of the image, and since the loss was now only calculated within the mask, the model was able to break away from the trivial solution of predicting zero gradients everywhere and returning the swirled image. In the future, to extend this research, I could experiment with an additional layer which predicts the center of the swirl, so as to make my architecture applicable to a less structured dataset. Another one of my primary issues was ensuring that the unswirl function correctly reversed the geometric transformations applied by the swirl function, which initially produced outputs that were either nonsensical or barely distinguishable from the swirled images. This was due to mismatches in the deformation grid calculations and an improper alignment of pixel displacements, necessitating a deeper understanding of both polar coordinate systems and the specific mechanics of skimage.transform.swirl. Normalizing the radius and strength parameters to align pixel-space transformations with normalized grid coordinates was a critical early step, but even this required fine-tuning to ensure consistency. The implementation of a distortion field U-Net posed its own set of challenges, as the model had to predict parameters like radius and strength while simultaneously learning pixel-level mappings to unswirl the images. Early experiments with simpler architectures, such as a regular CNN, highlighted their limitations in modeling spatial distortions, which prompted the shift to a U-Net with deformable convolutional layers and skip connections. Alongside this architectural adjustment, I faced difficulties with the loss functions, which initially failed to guide the model effectively. Integrating metrics like SSIM and incorporating penalties for physically impossible predictions, such as negative radii or excessively large strengths, proved instrumental in addressing these issues. Even with these changes, gradients often stagnated at zero, requiring strategies like leaky ReLU activations, gradient clipping, and dynamic hyperparameter tuning to stabilize training. Iterative hyperparameter optimization, including adjustments to learning rate, penalty weights, and gradient clipping thresholds, was essential to improve convergence. Throughout the process, visualizing reconstructions at regular intervals revealed further nuances, such as the model predicting uniform blocks of color, which pointed to deeper issues in how swirl parameters were being predicted or applied. Debugging these problems highlighted the importance of ensuring alignment between the forward swirl transformation and the inverse unswirl process, both in the implementation and in the training loop. Despite these challenges, I was able to refine both the model and the dataset preparation process, making thoughtful choices at every stage to ensure that the final approach was both robust and capable of reversing the swirl transformations effectively.

Even though I normalized and shrank the images, and only used a small subset (1000 images in total across train, test and validation), my computer still struggled with the size and complexity of the model. When training on more than 10 images, my kernel consistently crashed. I did not want to shrink the images further, since preserving facial detail was a key part of helping the model generalise and solve the problem at hand. To get around thesse contraints, I worked using a Tesla 4 GPU in Google Collab, which allowed my to train my model in 2 hours and 21 minutes when using 1000 images, and did not crash my kernel. I would have ideally liked to train on a larger subset of my synthetic dataset, but this would have increased the runtime taken to train significantly and cost a lot of money, but to further improve the model it could definitely be done in the future.

Interpretation and Reflection on Output

My primary technique was a U-Net-based convolutional neural network (CNN) augmented with self-attention layers, designed to predict a distortion field for image unswirling. I trained the model on a dataset of swirled and unswirled image pairs using a composite loss function that combined SSIM loss, smoothness loss, and perceptual loss. This combination aimed to balance structural similarity, spatial consistency, and perceptual quality in the reconstructed images. I ran my model over 100 epochs, with a patience of 10, meaning that if the validation loss did not improve over 10 epochs in a row, early stopping would be enacted. My model trained for 68 epochs before early stopping was triggered. The training and validation loss are given in Figure 2:

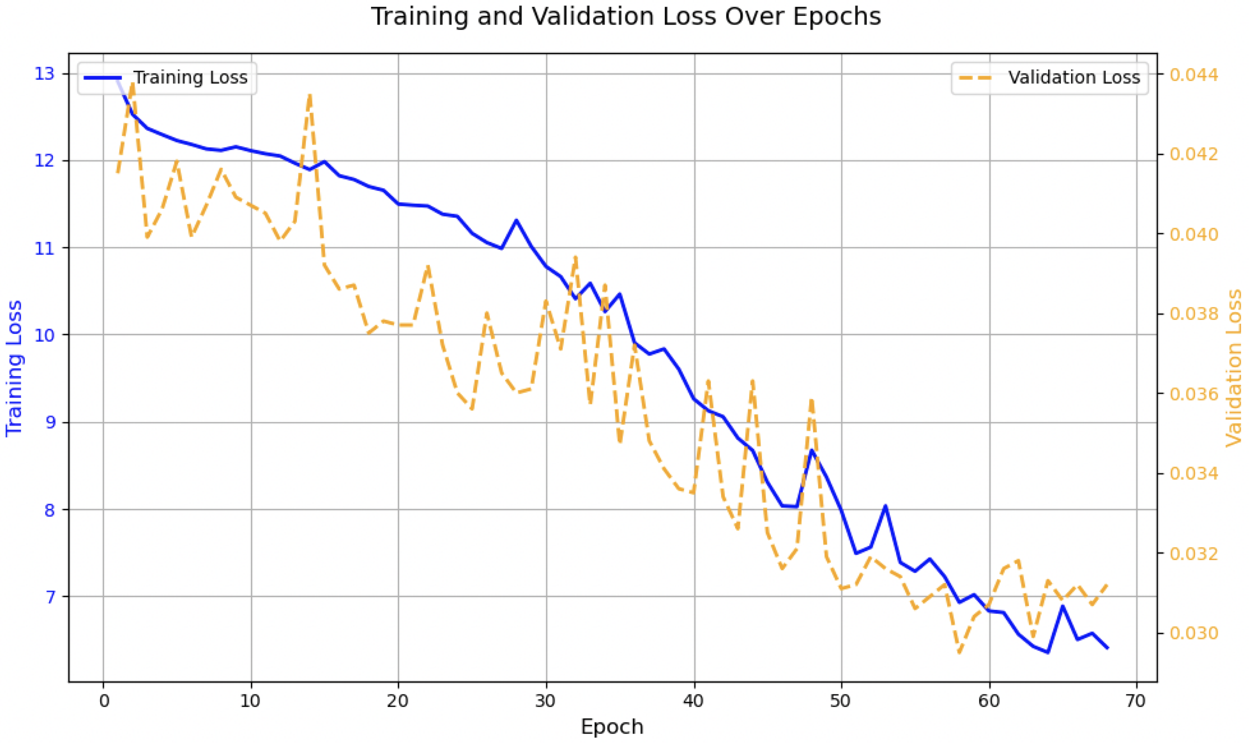

Figure 2: Training and validation loss over epochs

The results demonstrate a consistent decrease in both training and validation loss over epochs, indicating the model was effectively learnning the flow field mapping from swirled to unswirled images. Note that, the training loss was consistently higher than the validation loss, since the I calculated training dataset loss per batch, whereas the validation loss was averaged over all 100 validation samples every epoch. In figure 2, the gradual but significant reduction in validation loss suggests the model generalized well to unseen data. My choice to include masked loss was particularly effective in focusing the learning on the swirl region, avoiding trivial solutions where the model could ignore swirl effects and simply reconstruct unchanged image areas. Additionally, the ReduceLROnPlateau learning rate scheduler dynamically adjusted the learning rate, and although my learning rate was not reduced, for bigger batches of images and more unseen data, this will enable smoother convergence and improved final performance. In terms of visual and practical success, the model did reasonably well in unswirling images, with reconstructed outputs showing reduced swirl effects and improved alignment with the original unswirled images. It was able to recontruct the eyes and mouth of many images to a fairly accurate degree, and make the people in the images far more recognisable. It struggled most in the centre of the swirl, usually placed by the nose, where the swirl was strongest and the pixels most displaced. In general, from examining many of the outputs from the , test and validation datasets, the model worked well on unseen data, but some residual swirl artifacts remained in certain cases, suggesting potential areas for improvement. I reconstructed all images from my test dataset using my fitted model to examine visually, and I display in figure 3 a 'median' image (i.e. not the best perceptually, but not the worst) from the test dataset is displayed in Figure 3:

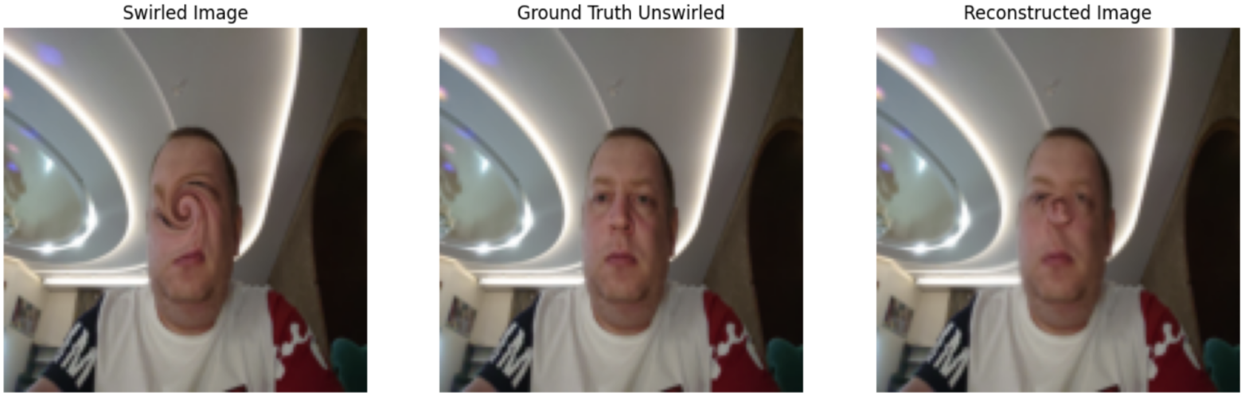

Figure 3: A random image from my test dataset reconstructed using my trained model

The results generally aligned with my expectations- I knew that even with a carefully curated synthetic dataset, completely unswirling an image would be a very difficult task, especially given my storage, time and GPU contraints. However, U-Nets have a well-documented history of success in image-to-image translation tasks, and their encoder-decoder structure is particularly suited for tasks requiring spatial information preservation, like unswirling. The ability of my model to learn effectively, and in many cases accurately recreate facial features from jumbled up swirled pixels was encouraging. While the model's performance met my initial expectations, fine-tuning architectural components, hyperparameters, and loss weightings could enhance myresults further. Some unexpected challenges, such as the tendency of the model to overfit certain swirl patterns or struggle in high-swirl intensity cases, indicate areas for refinement.

Several techniques were explored to improve model performance, many of which have been detailed in my 'Description and justifications of methods and analysis' section. These included self-attention layers, a mask to focus the model on only the area of the swirl, a deep U-Net with many layers, and a composite loss function which captured three different types of loss. Experimenting without a mask caused the model to converge to the trivial solution of giving me back the unchanged input image, and experimenting with just using a single type of loss gave me less accurate results on the test and validation datasets. I also initially experimented with different values of hyper parameters, such as a lower patience and learning rate, but the values that I ended up with proved to be the most effective out of all the values that I tried. I also initially experimented with a larger image size, but this slowed down training too much, and un-normalized images, but this made our model less accurate. Initially, to experiment with all these techniques, I just fit the model on a single image and printed gradients at each step to check that it was learning effectively, and monitored processing times and loss components at each step. Once I was happy with my framework and architecture, I trained the model in Google Collab on a GPU, and then examined the results qualitatively by looking at each of the reconstucted images from the test dataset compared to their originals, and by calculating the test loss. The test loss was very similar to final validation loss, suggesting that the model generalised well to unseen data: $testloss = 0.0325$, $valloss = 0.0312$

Some specific characteristics of my synthetic dataset made my specific approach better. Defining the swirl region to be at the centre of images of a standard size allowed my mask (which was perhaps the most crucial part of helping the model learn a non-trivial solution), to work effectively on every image in the dataset. My dataset consisted of relatively simple images, all containing clear centralized faces, with identifiable swirl patterns. This simplicity made my U-Net architecture, sufficient for the task. For datasets with higher complexity or greater variability, a more advanced architecture (e.g., incorporating much deeper U-Nets, or transformers) might be necessary. While the current approach performed well for this dataset, further adjustments—such as augmenting the dataset, experimenting with alternative architectures, fine-tuning hyper parameters (including loss weightings), or using much more data to train the model on, may lead to even better outcomes.

Concluding remarks

In this project, I explored the application of a U-Net-based architecture, augmented with self-attention layers and a composite loss function, to address the complex task of reversing swirl distortions in images. The combination of a synthetic dataset, a distortion field prediction framework, and a carefully designed loss function enabled the model to effectively reconstruct unswirled images, achieving promising initial results in both quantitative metrics and visual quality. The model demonstrated a consistent decrease in training and validation loss over epochs, highlighting its ability to learn and generalize to unseen data. The inclusion of a mask to focus the learning process on the swirl region was instrumerntal in overcoming challenges with trivial solutions, and the composite loss function—integrating SSIM, smoothness, and perceptual loss—proved to be an effective balance between structural, spatial, and perceptual quality. The test loss, with a value close to the validation loss, further reinforced the model's generalisability and robustness. Visual evaluation of reconstructed test images revealed significant improvement in unswirling while maintaining facial coherence, though residual artifacts persisted in high-swirl-intensity regions. These results align with expectations given the inherent complexity of the task and the constraints on computational resources and dataset size. This study highlights the importance of dataset characteristics in designing and evaluating deep learning models. The simplicity and controlled variability of the synthetic dataset were critical to the success of the chosen approach. However, these characteristics also limit the generalisability of the model to more complex, real-world scenarios. Expanding the dataset with additional variations such as data augmentations and using less clear facial data, exploring alternative architectures like transformers, and leveraging larger computational resources would likely yield further improvements. Overall, this project demonstrates that advanced deep learning techniques can effectively address the challenging problem of reversing image distortions, with potential applications in forensic investigations and other domains. While there is room for further refinement, the methods and results presented here represent a significant step toward robust solutions for image restoration tasks.

Bibliography

Alexander Kapitanov, Karina Kvanchiani, S. K. (2024). Easyportrait dataset. High-resolution portrait dataset used for computer vision research. Dataset accessed and adapted for synthetic swirl generation.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241. Springer International Publishing.