Adaptive Document Rectification Network for Paintings

Progress Report of UROP2100, Summer 2020

Hanfang LYU, Joszef Maximillian ADIGUNA, Yuheng SHAO

Hong Kong University of Science and Technology

{hlyuag, jmadiguna, yshaoam}@connect.ust.hk

Supervisor: Dr. Desmond Yau-chat TSOI *desmond@ust.hk*

Abstract

Document digitization efforts have been made on document rectification tasks. Traditional methods for document digitization focused mainly on correcting distortions using additional information other than the original image. Improvements in the field have allowed the use of deep learning approaches in favor of traditional algorithms, as they have been proven to be more accurate, resulting in higher quality digital images more suitable to digital analysis. We propose that these methods can also be adapted to perform correction on non-text documents, mainly paintings and pictures. Preliminary training using paintings has shown that existing methods work to some extent, correcting certain geometric distortions akin to paintings, albeit with lesser degree of success.

1. Introduction

The widespread usage of mobile devices equipped with high quality cameras allows for the digitization of many documents and images. The outset of digital cameras allows for easier and accessible photographing technology, replacing the need for clunky technology to extract relevant information from a certain document. The limitation of taking pictures is that there may be shading errors or geometric distortions present within the document itself, decreasing overall quality of the photograph. Different lighting and camera angles may also affect the quality of scanned documents or paintings.

Modern improvements on algorithmic approaches to document rectification involves the use of a deep learning network, which amplifies the accuracy and robustness of rectification, even allowing the rectification of severely distorted images. Recent approaches, such as [1] and [2] have opted to use a U-Net or stack of U-Nets, which are convolution networks used for precise segmentation of images, while [3] opts to use patches to train two networks which handle geometric and illumination, which then stitch the patches back together into the rectified image. These methods, however, focus mainly on textual images, with uniform background and low variety of colors.

During this term, we pursued a modified version of previous methods, but trained using new data which focuses on simpler distortion meshes along with painting textures, to signify the differences between textual and graphical documents. Furthermore, training with simpler meshes allows for experimentation on how robust the network is in performing rectification.

2. Investigation on Previous Literature

2.1. DocUNet

DocUNet [1] is a deep learning approach that utilizes two U-Nets that are stacked together, with the output of the first U-Net being used as the input for the second U-Net. The design of this method is to use the first U-Net to be the predictor, and then use the second U-Net to refine the prediction of the first U-Net. The result is evaluated using a measure of image similarity, and that this method achieves a measure of 0.41 in terms of MS-SSIM with an average LD of 14.08 pixels, improving upon methods that depend on text line tracing.

2.2. Patched CNN

Patched CNN [3] is a method developed with the intention of document rectification by determining the distortion flow of the images. After correction of each patch, a robust technique is used to stitch the images together allowing for further networks to provide illumination correction. This is possible due to the smaller amount of distortion that is present in each patch of the document.

2.3. DewarpNet

The DewarpNet [2] team contribute the dataset Doc3D and use a different end-to-end deep learning network to unwrap the document. They use Doc3D renderer to create a dataset containing distorted document images with ground truth. And their network architecture consists of two networks: shape network and texture mapping network. The shape network is designed for regressing the 3D shape of the input image. A U-Net style encode-decoder with skip connections is used here. And the texture mapping network is to transform from the 3D coordinate map to backward mapping. Then through bilinear sampling, the network generates the final unwrapped image.

The dataset required by DewarpNet is relatively simpler than [3], and the network is more efficient than [1]. So after reviewing these previous pieces of literature, we decided to study [2]. Their renderer was used to generate our own dataset of distorted-painting images, and we use DewarpNet to train our dataset.

3. Dataset

3.1. Data Generation

We utilize the Doc3D renderer aided by a secondary team, which replaces the document images with painting images which have been collected from previous efforts. This is then complemented with the creation of basic distortion meshes which represent binder curl along with fold distortions, augmenting the complicated distortions that have been provided by the Doc3D authors beforehand. The paintings are then texture mapped onto said meshes and placed in high-dynamic range images to simulate real-life photographic situations.

3.2. Load Dataset

File Name Correspondence. Use Python scripts to grab all image filenames and create text files for both the training set and the validation set. This helps the data loader to identify correct inputs.

Loading data and preprocessing. Labels, which are in terms of 3D coordinates, are generated in *.exr* format. After transforming images to 8-bit unsigned integer array and labels to floating type, we crop the images to approximately fit in the labels' size.

4. Method

4.1. Introduction to DewarpNet

Network Structure. We use two subnetworks: a shape network and a texture network. Because the texture network tends to generalize well and requires less training, in this report, we will dig into the shape network.

As shown in Figure 1, the U-Net performs sequential down-sampling and up-sampling, taking advantage of (transposed) convolutional layers. DewarpNet avoids max-pooling by carefully choosing the kernel size, stride and padding numbers to get the same effect. Figure 2 describes a typical submodule in the shape network of DewarpNet.

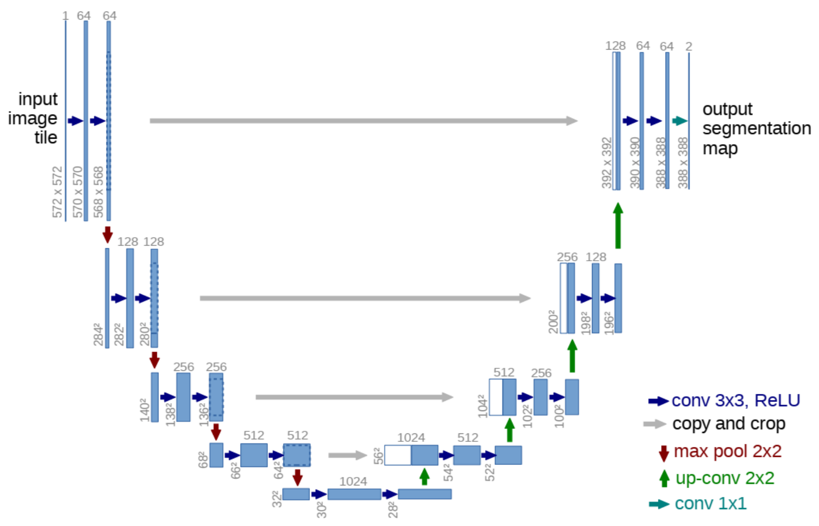

Figure 1. The original U-Net design. Notice the DewarpNet does not use max pooling for down-sampling. Instead, a 4x4 kernel with stride 2 and 1 zero-padding achieves the same purpose.

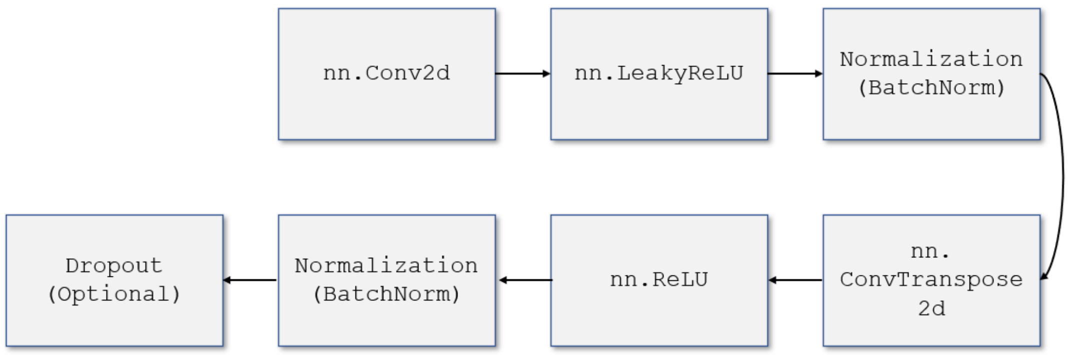

Figure 2. A typical layer of the shape network of DewarpNet, consisting of a complete down-sampling to up-sampling process.

4.2. Experiment

Hyperparameters. We run the training for 100 epochs, with an initial learning rate of 1e-5. We have a learning rate scheduler which decreases the learning rate dynamically.

Loss function. Loss is calculated using: 1. The value difference between the predicted 3-D coordinate map (the "prediction") and the ground truth 3-D coordinate map (the "label"); 2. The weighted gradient difference between the prediction and the label. Notice we use L1 distance for these two criteria.

Training Process and Loss Graphs. As shown in Figure 3, Our hyperparameter is chosen without loss explosion or unstable loss decrease.

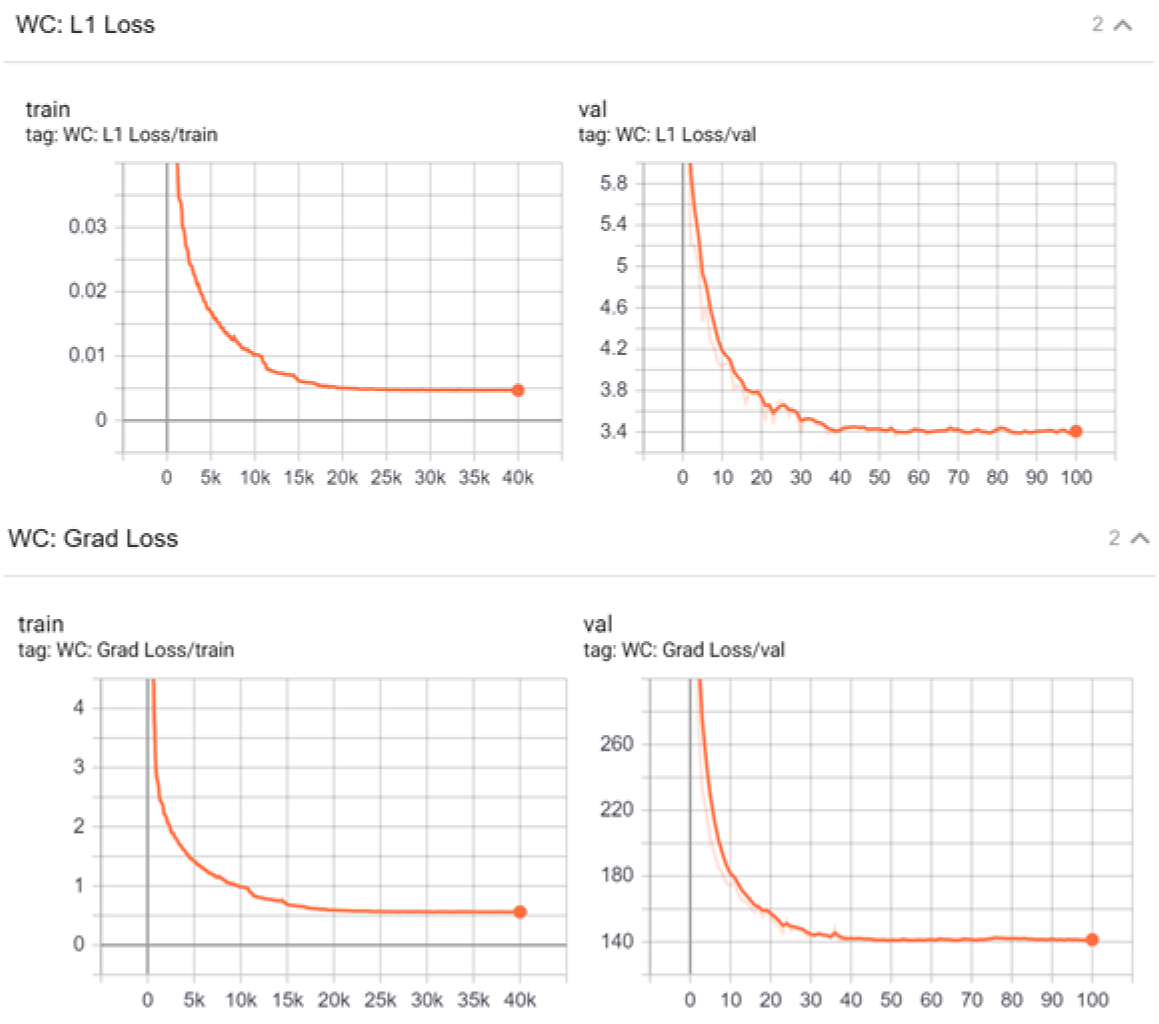

Figure 3. Up: L1 loss (value difference). Down: Gradient Loss (weighted gradient difference).

Experiment Results. As described in Figure 5, the experiment rectified the geometric distortion to some extent. However, the model failed to predict some input images. We reasonably guess it's because of arbitrary distortions in our dataset is difficult for the model to identify a common mapping.

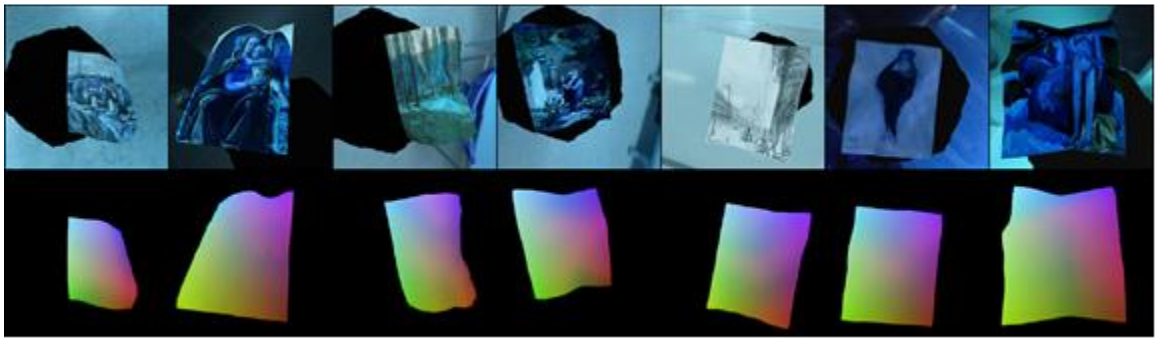

Figure 4. Up: An example of training images. Down: An example of the 3D coordinate maps (the "labels").

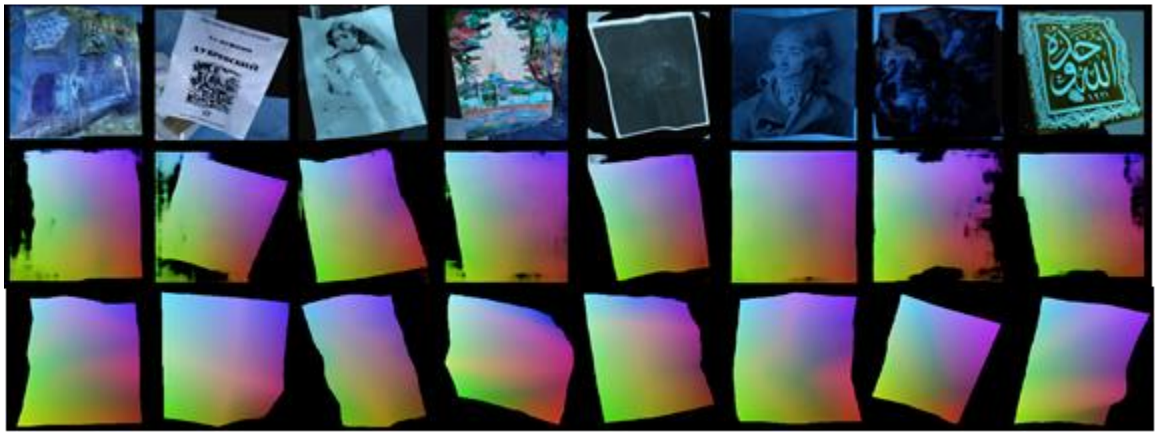

Figure 5. Top: An example of validation set images. Middle: Model prediction of the 3D coordinate maps. Bottom: The ground truth coordinate maps.

5. Limitations and Future Extension

As we have seen from Figure 5, some preliminary predictions are grainy and have uneven shapes. This results in a large disparity between predicted results and the ground truths of these images. Severe lighting disparities along with colors that blend with the background result in the network not being able to properly segment the images from their backgrounds, resulting in severe distortions in the end predicted coordinate maps.

Future improvements can be made on making a more robust model, incorporating a smoothing function that coerces the predicted 3D coordinate map into a shape that a user may specify. Furthermore, this model should also be able to be utilized through a mobile device, as to be able to make use of high-quality digital cameras directly.

6. Conclusion

In this UROP, we reviewed several previous literatures and discussed our method to unwrap the distorted painting images. After several trials, we decided to apply DewarpNet to the dataset of images containing distorted paintings. We use the Doc3D meshes which contain arbitrary distortions to train the deep learning network. However, the output is not satisfactory, and the model failed to predict some complicated distortions. So in the future improvements, we may start with generating data with simple distortion like bind or curl, so that the unwrap function will be easier to train. Then we will gradually come up with something more complicated and even random distortion to build a more robust model.

References

[1] K. Ma, Z. Shu, X. Bai, W. Jue and S. Dimitris, "DocUNet: Document Image Unwarping via A Stacked U-Net," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 06 2018.

[2] S. Das, K. Ma, Z. Shu, D. Samaras and R. Shilkrot, "DewarpNet: Single-Image Document Unwarping With Stacked 3D and 2D Regression Networks," *IEEE International Conference on Computer Vision (ICCV),* 10 2019.

[3] X. Li, Z. Bo, J. Liao and P. V. Sander, "Document Rectification and Illumination Correction using a Patch-based CNN," *ACM Transactions on Graphics (TOG),* 11 2019.