ENGN8501 Project
DeepLens: Shallow Depth Of Field From A Single Image

Yiding Qiu u7175591
Haixu Liu u7215510
Wenjia Cheng u7234723

Abstract

The shallow depth-of-field (DoF) effect can improve the visual effect by blurring out the less important part of images thus highlighting the region of focus. In this project, we aim to implement a novel neural network model which consists of a depth prediction module, a lens blur module, and a guided upsampling module to generate high-resolution shallow depth-of-field (DoF) images from a single all-in-focus image with controllable distance and aperture size. All these modules are differentiable. We also capture some images using our own mobile phones and apply the model to them. The results show that the model is quite effective and produces few artifacts.

1. Introduction

The shallow depth-of-field (DoF) effect is quite important in photography because it highlights the region of focus of the image by blurring out the rest of it thus improving the visual effect. However, single-lens reflex (SLR) cameras which are used to capture these images are too expensive for casual photographers. Besides, these images are hard to refocus onto other regions or change the amount of defocus once captured.

To deal with these, we aim to implement a novel neural network that consists of three modules: a depth prediction module that estimates the image depth, a lens blur module that predicts the spatially-varying blur kernels, and a guided upsampling module that generates high-resolution shallow DoF images. Since all the modules are differentiable, the network allows end-to-end training.

We capture some images by our own mobile phones to apply the network to and we find that it is effective and produces few artifacts.

In summary, we present the following contributions:

(1) Implementing a neural network which consists of a depth prediction module, a lens blur module, and a depth prediction module.

(2) Applying the network to images that are captured by our own mobile-phones and reflecting on the results.

2. Problem Statement

As [2] said , it's possible to cast the shallow DoF rendering problem as a black-box input-to-output regression problem. In this work, we aim to solve three main problems: depth prediction, lens blur rendering, and guided upsampling.

2.1. Depth Prediction

In this module, we aim to predict depth map by minimizing the loss functions:

$$J_d(\theta_d) = \| D - D^g \|_1 + \gamma \times \| S - S^g \|_1, \qquad (1)$$

where D and S denote the predicted depth map and foreground segmentation map; D^g and S^g are the corresponding ground truth; γ represents a trade-off parameter to balance the two tasks.

2.2. Lens Blur Rendering

According to [3], [5], and [6], given the predicted depth map together with the original image, existing image space or object space rendering methods could be utilized to approximate lens blur effects. However, the results of their methods often contain artifacts around object boundaries. In this work, we aim to overcome this by a differentiable neural network to approximate the spatially varying lens blur kernel. There is one problem, if the input image is of $H \times W$ resolution and a maximum blur kernel size is of $k \times k$, it would require computing an $H \times W \times k^2$ kernel tensor, which would take too much memory. In this work, we aim to downsample images first for computation and then upsample them to generate high-resolution results. In a word, in this part, we aim to predict kernel given the depth map, focal depth, and aperture radius.

2.3. Guided Upsampling

As mentioned in the last section, we downsample images to reduce computational cost. Therefore, we need to upsample them to generate the final high-resolution results. Since we already have original high-resolution images, the problem is how we could extract information both from the low-resolution space and high-resolution space.

3. Method description

The model in this paper could be divided into 3 modules, the depth prediction module, the lens blur rendering module, and the guided upsampling module. The overall architecture is shown in Fig 1.

3.1. Depth Prediction

This module aims to predict depth. It consists of an encoder and a decoder followed by multi-task heads. The encoder extracted features using the first 14 residual blocks in a pre-trained ResNet-10 network. At the end of this sequence of residual blocks, there is an atrous convolution layer used for preserving features. After extracting feature maps, a pooling pyramid with four different sizes of kernels is applied.

As for the decoder, it begins with a set of upsampling layers in order to upsample the feature maps. Then the decoder divides the upsampled feature into two multi-task heads, which aim to predict the depth and segment foreground separately, both of them have three convolutional layers.

Besides, the depth of field rendering process does not need the exact values on the depth map, thus, we normalized these depth maps to [0, 1] as it's convenient for computation.

3.2. Lens Blur Rendering

Within the lens blur rendering process, the computational complexity of it could be huge as we are applying a varying blur kernel. The tensor we need to calculate will have the shape of $H \cdot W \cdot k^2$, where H and W denote the height and width of the input image, and k represents the blurring kernel size. If the kernel size is a bit large and the input image has multiple channels, then the network will take a huge amount of memories. In order to handle this issue, we decided to apply a downsampling operation before feeding into the blur rendering module, which could get a scaled image that is smaller than before. Besides, after rendering the lens blur, the scaled image would be restored by a guided upsampling module.

The lens blur module consists of a feature extracting network and a kernel predicting network. After feeding low-resolution images to the feature extracting network, the network calculates the multi-scale feature maps through several convolutional layers, with ReLU between these layers. In the end, these feature maps are concatenated into a set that contains different extracted features in multiple scales.

Then, the kernel prediction networks take the downscaled depth map and the focus parameters as inputs, where focus parameters include focal depth and aperture radius. The predicted kernel tensor K is calculated by these parameters and a $1 * 1$ kernel, the shape of it should be $h * w * c$, h, w, and c denotes the height, width and channel of the input image. Besides, the final network in kernel prediction also takes the original sized image as input, in order to fix the errors in reshaping images.

After calculating feature maps and kernel tensor, the the shallow DoF image is rendered by the equation below:

$$L_i(x, y) = \sum_{j=1}^{c} K(x, y, j) \cdot F_i(x, y, j) \tag{2}$$

where $L_i(x, y)$ represents the color value of the rendered DoF image, at the location of (x, y).

3.3. Guided Upsampling

The guided upsampling module is introduced by [1], which is well designed but not appropriate for our model. Thus we implemented a guided upsampling module which takes the low-resolution feature maps and the original high-resolution image as inputs, then predicts two high-resolution spatial weight maps M_A and M_L, and concatenates these maps together with the original image and the shallow Dof image. Finally it would get our result, which is a high-resolution shallow D0F image H:

$$H = M_A \odot A + M_L \odot \hat{L}, \tag{3}$$

where $\odot$ denotes element-wise multiplication. A denotes the high-resolution all-in-focus image, and $\hat{L}$ denotes the shallow DoF image upsampled from the low-resolution L, as shown in **Equation (2)**, using bilinear interpolation.

4. Experiments

Firstly, we drew a simplified flow chart Fig 2 for our model in order to implement the model conveniently. In our work, we took several pre-trained networks such as ResNet.

After building the network structure, we tested the model with our own images. Some of the results are shown as Fig 3. The images of our result from left to right are the original image, image with aperture radius as 5, image with aperture radius as 10, depth map.

As can be seen in the images, the depth maps given when working with outdoor images are believable, even for intricate scenes, such as plants, and have better results. However, the shallow depth of field results based on this rather messy depth map is not quite as good as they could be, for which there is no good solution at the moment. What's more, the shallow depth of field effect is also rarely used in actual photography for scenes such as this. However, this model becomes less effective when applied to examples of flat ranges but with large colour differences. For example, in Figure 8, the hand and the wood are on almost the same plane, yet a depth gap is recognized. What's more, this pre-trained ResNet model will tend to set the depth closer to the lower edge of the image lower. And because there is no real reference, in some cases where there is no subject, the ground

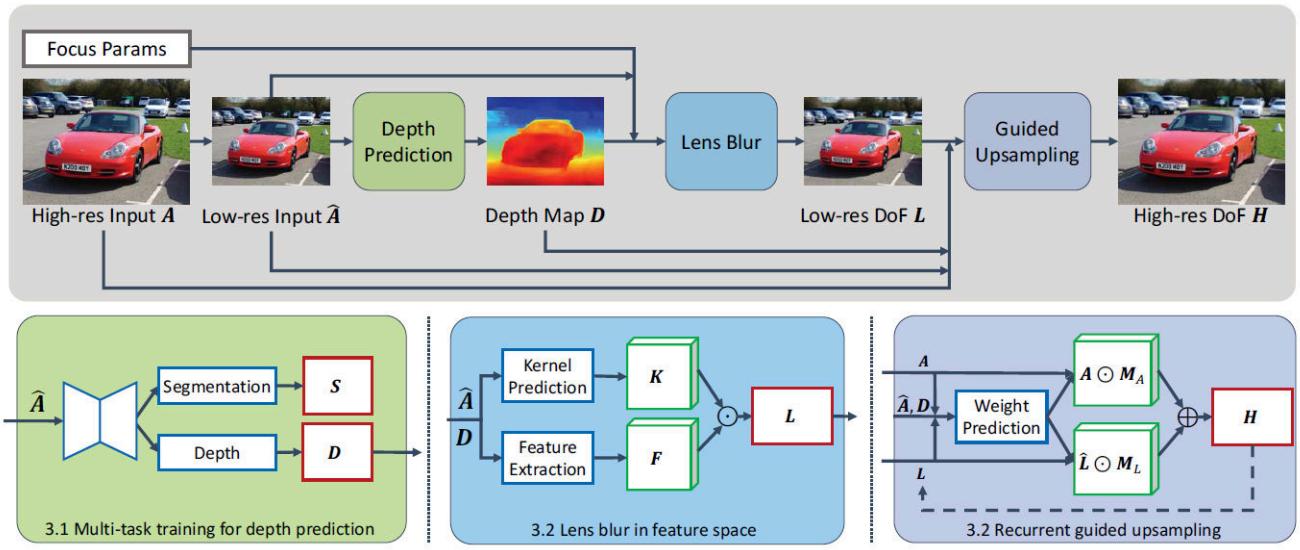

Figure 1: Overall model architecture

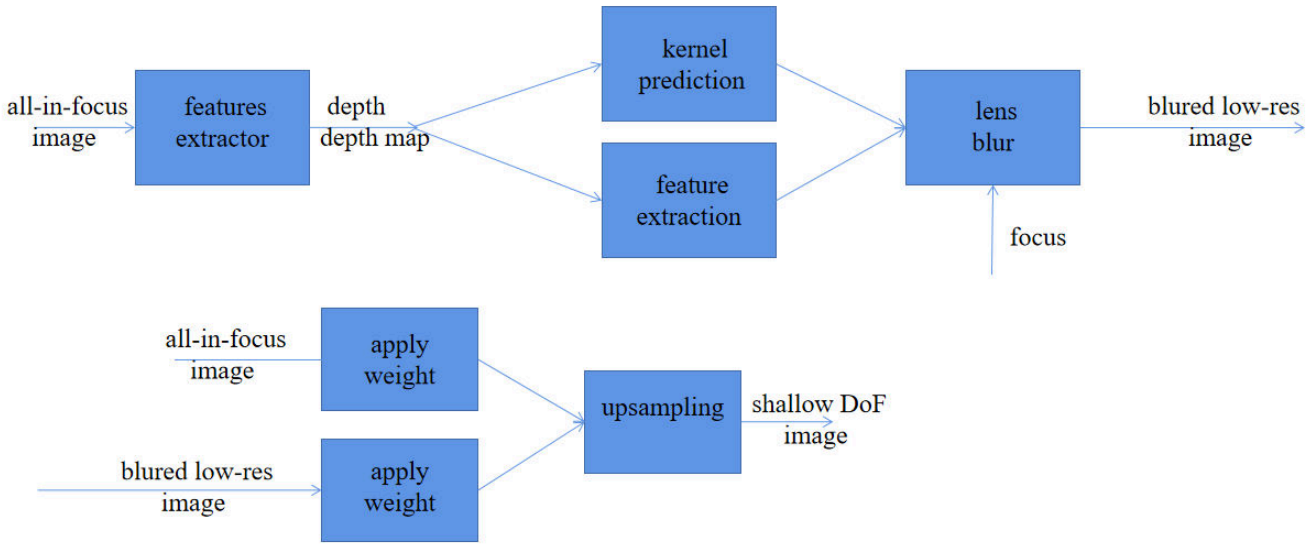

Figure 2: Flowchart image of our model

at close range is shown as if it were only a few centimeters from the camera. If it were possible to get a reference, for example, an object with a definite distance. Marking it in the frame might solve this problem, but it would require introducing additional input variables when generating the depth map. This should solve the situation where the blurred effect generated in Figures 5 and 8 does not match the reality (it looks like macro photography, especially in Figure 5.). But overall, the shallow depth of field effect derived from this depth map is good. For example, in Fig. 2, when the focus is set on the tree and on the lantern. Two different blurring effects are exhibited. Also, the degree of blur changes when different aperture sizes are entered. In a follow-up study, it might be useful to consider introducing semantic analysis. The first step would be to add clustering analysis to all of the feature extraction, first to determine the form of the scene and then to analyze the content of the subject. This might improve the robustness in the face of different environments.

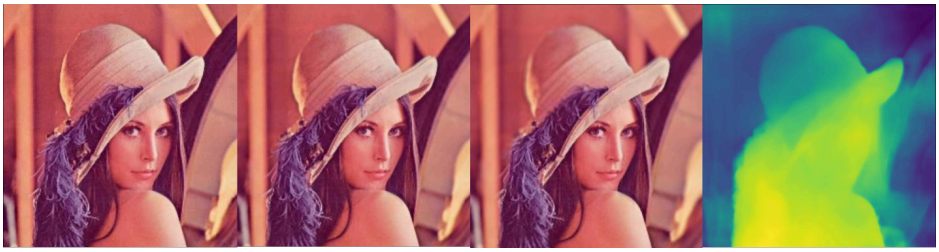

Figure 3: Test result 1 of our model.

Figure 4: Test result 2 of our model.

5. Conclusion

In conclusion, we have introduced a deep neural network, which combines different techniques. The model is composed of three modules, which are depth prediction, lens blur rendering and guided upsampling. We tested some images on our model and got high-resolution shallow DoF images with different aperture parameters.

6. Confidential Peer Review

Wenjia Cheng In doing this project, to the best of my judgement, I confirm that Haixu Liu mainly contributed to code modifying, result analysis and report writing. and his overall contribution is about 33%, and Yiding Qiu mainly worked on applying experiment, code modifying, preparing slides and resource finding, and his contribution is about 33%. We not only focus on our own part but also discuss with each other to come up with personal ideas.

References

[1] Kaiming He, Jian Sun, and Xiaoou Tang. "Guided image filtering". In: *European conference on computer vision*. Springer. 2010, pp. 1–14.

[2] Phillip Isola et al. "Image-to-Image Translation with Conditional Adversarial Networks". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 5967–5976.

[3] Sungkil Lee, Elmar Eisemann, and Hans-Peter Seidel. "Real-Time Lens Blur Effects and Focus Control". In: *ACM Trans. Graph.* 29.4 (July 2010). ISSN: 0730-0301.

[4] Wang Lijun et al. "DeepLens: Shallow Depth of Field from a Single Image". In: *ACM Trans. Graph. (Proc. SIGGRAPH Asia)* 37.6 (2018), 6:1–6:11.

[5] Pratul P. Srinivasan et al. "Aperture Supervision for Monocular Depth Estimation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.

[6] Yang Yang et al. "Virtual DSLR: High Quality Dynamic Depth-of-Field Synthesis on Mobile Platforms". In: *Digital Photography and Mobile Imaging*. 2016.

Figure 5: Test result 3 of our model.

Figure 6: Test result 4 of our model.

Figure 7: Test result 5 of our model.

Figure 8: Test result 6 of our model.

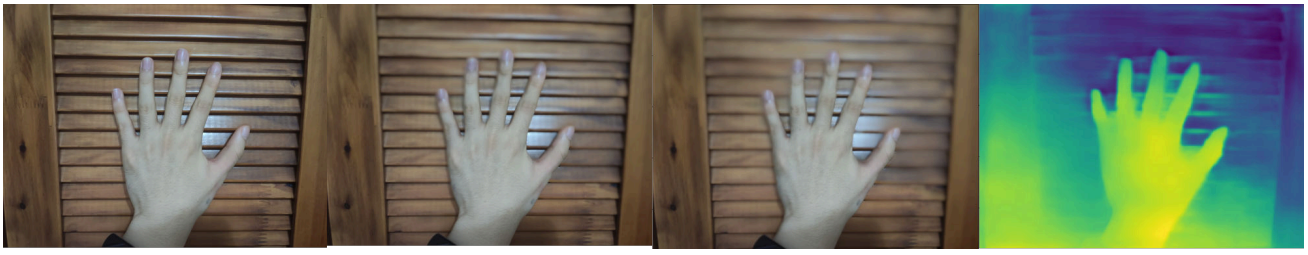

Figure 9: Test result 6 of our model.