Creative Integrated Design 2, 4190.413A
Seoul National University

Development of deep learning-based facial normal restoration techniques

Junhyeok Park, Daun Lee, Jiwon Lee

1 Goal

- Customizing SDPS which is a deep learning model that obtains a normal map to produce results of the same or similar quality while reducing the number of inputs.
- Develop a Deep Learning Network that acquires normal information from human faces from a single portrait (or multiple photos reflecting multiple light source information).
  - Version 1 : Inference normal map from 2 stages(LCNet, NENet)
  - Version 2 : Inference normal map from a single stage(LCNet + NENet)

2 Introduction

Normal maps are usually created using Photometric Stereo techniques, but require acquisition in a limited location, which is challenging and highly constrained to acquire. Therefore, it is difficult to create a normal map using information acquired in a wild environment, so there is a demand for technology that can predict a normal map even from photos taken freely using Deep Learning. If the technology is developed, normal maps can be utilized in a variety of photos, making the texture creation for rendering much easier, which can be very useful.

3 About SDPS

SDPS[1] is a base model which we are modifying. This model receives 64 pictures of one object as input, and get the final normal map through two stages.

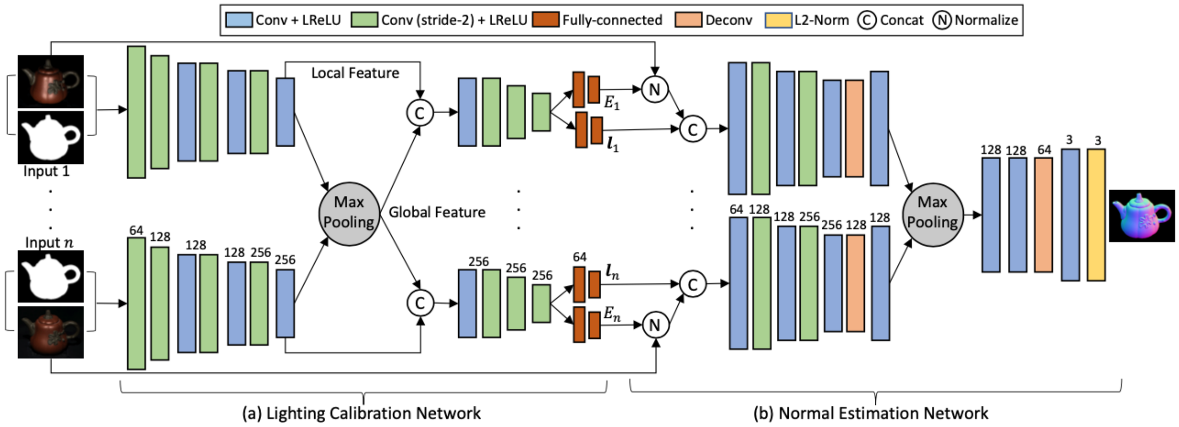


Figure 1. The network architecture of SDPS-Net is composed of (a) Lighting Calibration Network and (b) Normal Estimation Network. Kernel sizes for all convolutional layers are 3 × 3, and values above the layers indicate the number of feature channels.

Figure 1: Architecture of SDPS

3.1 LCNet

LCNet, the first stage of the SDPS, specifies the location of the light source through input images. To estimate lightings from the images, an intuitive approach would be directly regressing the light direction vectors and intensity values. However, SDPS proposes that formulating the lighting estimation can be considered as a classification problem. Classifying a light direction into a certain range is easier than regressing the exact values, and this will reduce the learning difficulty. Taking discretized light directions as input may allow NENet to better tolerate small errors in the estimated light directions.

Data that is finished learning on LCNet is concatenated through the Local-global feature fusion. The feature map extracted from a single observation obviously does not provide sufficient information for resolving the shape-light ambiguity. Photometric stereo uses multiple observations of an object. So using a local-global feature fusion strategy to extract more comprehensive information from multiple observations. the feature map extracted from each image is local feature, and global feature map is produced by max-pooling using all local feature map. This local, global feature map is concatenated and become input of NENet.

3.2 NENet

The second stage, NENet, performs a normal estimation with the location and image of a specific light source on LCNet. Specifically, using discretized lighting extracted from LCNet as input to the convolutional layer. while using this, NENet ensures noise tolerance over existing methods.

3.3 Perfromance validation of SDPS

The disadvantages of this model are that it is heavy, has many parameters, and has a long learning time. Before making modifications to the model, we experimented with the performance of the existing model for performance comparison with the modified model. Below table is a mean error value of performance according to the number of images. The validation image number is 1-10. The number of images used by SDPS model is 64. Looking at the rest, there is no significant difference except for the number of images. So we focused on simultaneous performance and model lightening.

	1	2	3	4	5	6	7	8	9	10	avg
in_img_1	15.038	20.26	15.096	17.685	16.749	18.208	20.245	16.407	18.536	14.215	17.244
in_img_2	6.07	6.238	11.324	4.833	6.868	12.021	9.164	11.226	7.793	19.548	9.509
in_img_4	3.194	7.919	4.379	4.539	6.07	5.141	8.329	9.566	4.833	6.875	6.085
in_img_8	3.178	7.435	3.697	4.128	5.795	6.644	9.962	7.272	3.966	6.649	5.873
in_img_16	3.387	4.691	4.26	3.421	3.74	4.703	9.677	5.955	4.887	6.716	5.144
in_img_32	2.465	4.879	2.726	3.404	4.179	4.76	7.556	6.4	4.362	6.757	4.749
in_img_64	3.271	4.077	5.436	3.471	2.866	4.344	10.357	4.501	4.516	6.322	4.916

Table 1: MAE of LCNet

	1	2	3	4	5	6	7	8	9	10	avg
in_img_1	12.991	22.226	23.991	15.22	19.2	29.436	28.675	29.879	18.21	27.108	22.693
in_img_2	5.994	12.857	14.172	14.2	11.482	17.135	20.286	21.25	10.59	22.161	15.013
in_img_4	4.564	10.233	8.786	10.303	8.963	12.248	12.95	19.204	7.669	19.153	11.407
in_img_8	3.221	9.404	7.516	6.643	8.173	10.52	11.803	17.442	8.224	17.996	10.094
in_img_16	2.318	8.73	7.823	4.783	7.192	9.614	11.074	16.495	8.925	18.537	9.549
in_img_32	2.698	9.846	8.304	7.52	8.552	9.407	10.518	15.917	8.036	18.56	9.936
in_img_64	2.765	8.06	8.141	6.892	7.499	8.975	11.91	14.905	8.48	17.43	9.506

Table 2: MAE of NENet

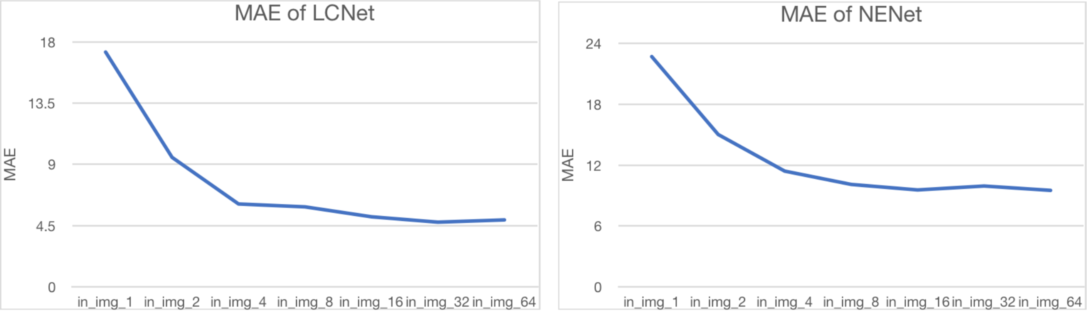


Figure 2: Performance of SDPS according to image num

4 EffectiveNet

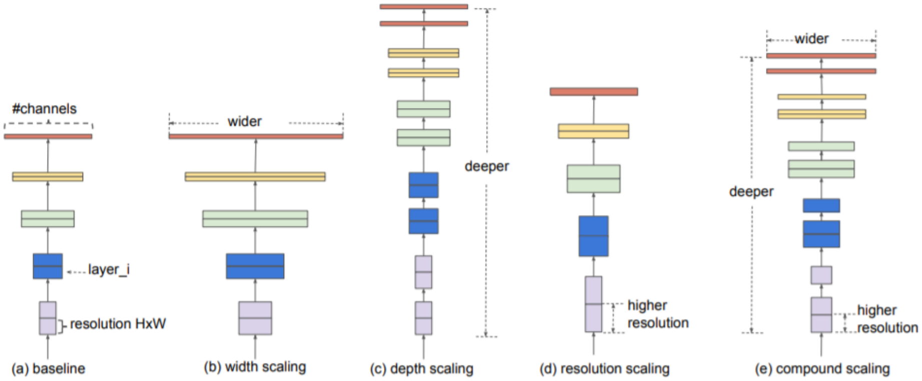


Figure 2. **Model Scaling.** (a) is a baseline network example; (b)-(d) are conventional scaling that only increases one dimension of network width, depth, or resolution. (e) is our proposed compound scaling method that uniformly scales all three dimensions with a fixed ratio.

Figure 3: Architecture of EffecitveNet

As explained earlier, LCNet regarded the source of light as a classification problem. So, we tried to use EfficientNet, a SOTA model in the field of image classification. we will briefly explain what EfficientNet is

and move on.
EfficientNet[3] is a model for image classification published in 2020, and it has performed well with small parameters. The key to this model is the scaling of the input data and the convolutional layer. In common sense, it has been widely known that scaling can improve the performance of the model, such as increasing the resolution of the input or increasing the number of channels in the layer. However, there was no model that attempted this quantitatively and mixedly, and EfficentNet proposed compound scaling that attempted this.

If performance is improved by scaling, a certain degree of superior performance of the existing model should be assumed. So, the author of this paper composed EfficientNet based on the MobileNet model. MobileNet is a model that focuses on weight reduction, and has the feature of delivering excellent performance with few parameters. So we also implemented the EfficientNet model based on MobileNet and applied it to LCNet as it is. The result is shown in the graph below. The parameters increased by nearly four times, but the accuracy was rather poor. So, we decided that it was not meaningful to simply import and apply a good model as it is, and we decided to try to implement the model by importing only the part that was judged to be useful among the features of the model.

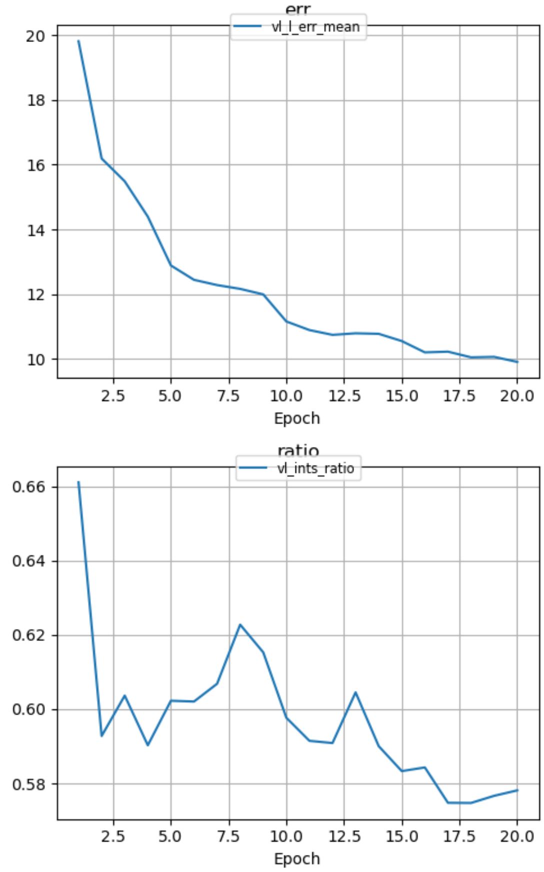


Figure 4: Performance result of LCNet

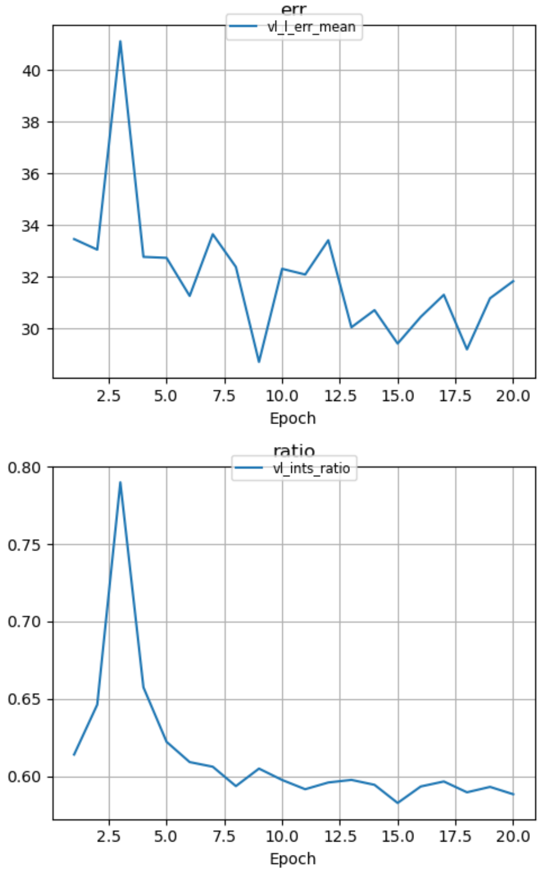


Figure 5: Performance result of EffectiveNet

5 Depthwise Seperable Convolution

We decided to apply only some part of the EfficientNet to the SDPS model. We first applied depthwise separable convolution.

Depthwise separable convolution is a twisted version of normal convolution. It is composed of 2 operations-depthwise convolution and pointwise convolution. Depthwise convolution, as you can guess from its name, is a way of convolution that is applied only among elements inside the same channel. Thus the number of channels of inputs should be equal to the number of channels of outputs.

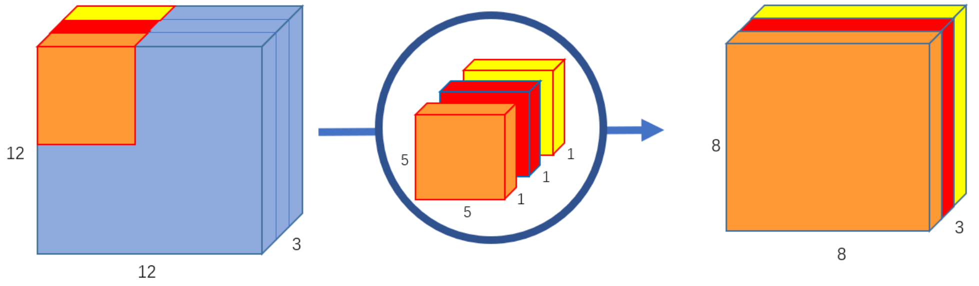


Figure 6: Example of Depthwise Convolution

Pointwise convolution, in contrast, is a way of convolution that is applied only among elements in different channels, but sharing the same point in the channel. Thus, unlike depthwise convolution, you can change the number of channels between inputs and outputs.

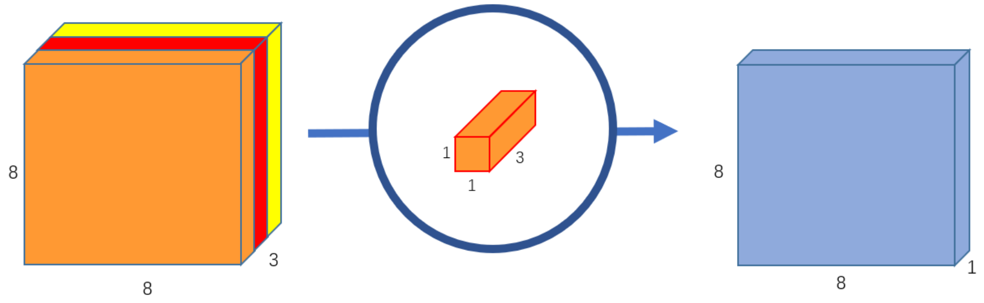


Figure 7: Example of Pointwise Convolution

Both depthwise convolution and pointwise convolution have less amount of operations to be done, resulting in a decrease in total parameters throughout the whole operation.

We replaced some convolutional layers with the depthwise separable convolution layers by replacing a single convolution layer with a single depthwise convolution layer followed by a single pointwise convolution layer. Other than applying depthwise separable convolution to the LCNet, we also added some residual blocks to create direct connections to some deep convolutional layers, which enhance the ability of gradients to be updated well.

6 Modifying NENet

We also tried to apply different methods to enlighten NENet, too. One of the methods we tried is "attention augmented convolution", which tries to apply transformers' way of applying attention. Since applying attention increases graphic memory, we failed to successfully train data due to lack of memory of our GPU. After putting some effort on reducing GPU memory, for instance, reducing the batch size, we successfully trained the model, but the result was poorer than the original model by far. Instead, we applied depthwise separable convolution on NENet, and got a better result. Compared to the model with LCNet modified with depthwise separable convolution, it showed significantly less drop in accuracy while maintaining same amount of decrease rate of parameters.

7 Modifying Light Intensity

Changing the light intensity attempted two ways. The first is the light intensity of the existing [1.0, 2.0) to the light intensity of [1.0, 2.0), which consists of images with brighter intensity than the original because the original picture is too dark. The second is that in the light intensity of the whole [0,2, 2.0), the interval is divided to have the same number of images in that interval. For example, given eight image inputs, we divide the intervals into four to produce images with two random light intensities for each interval. Below are the results the two versions and original with input image 16 mentioned above.

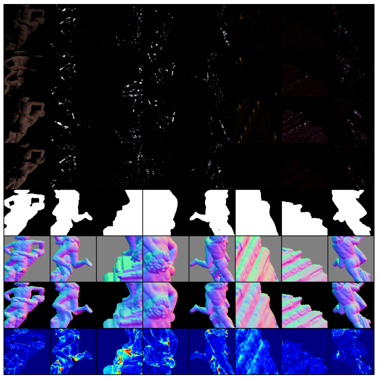


Figure 8: First version with [1.0, 2.0) light intensity

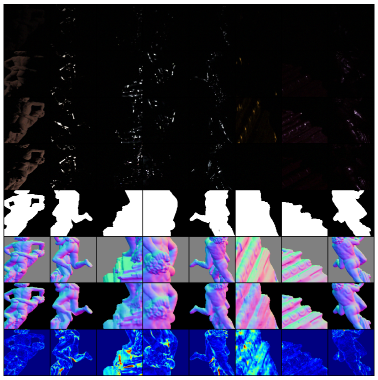


Figure 9: Second version with intervals

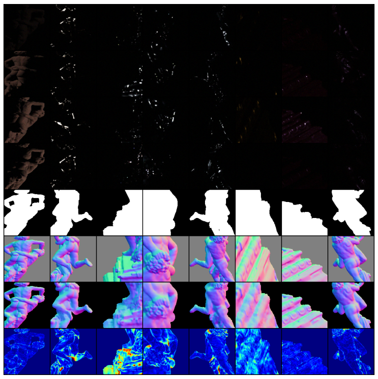


Figure 10: Original version

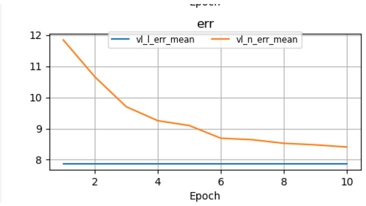


Figure 11: First version with [1.0, 2.0) light intensity

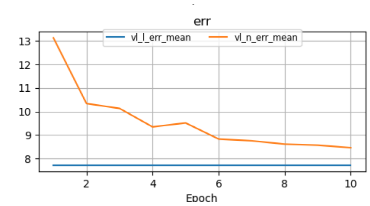


Figure 12: Second version with intervals

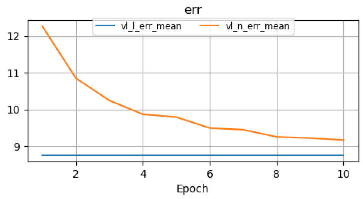


Figure 13: Original version

8 FusedNet

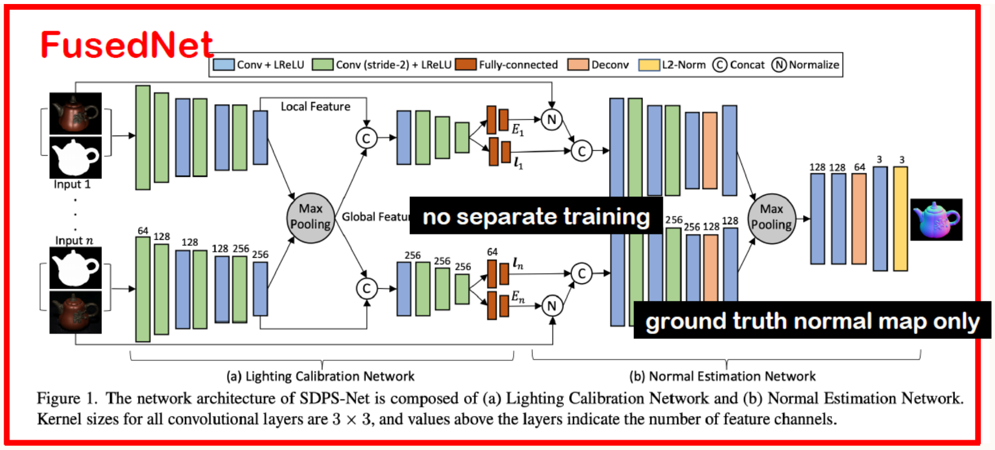


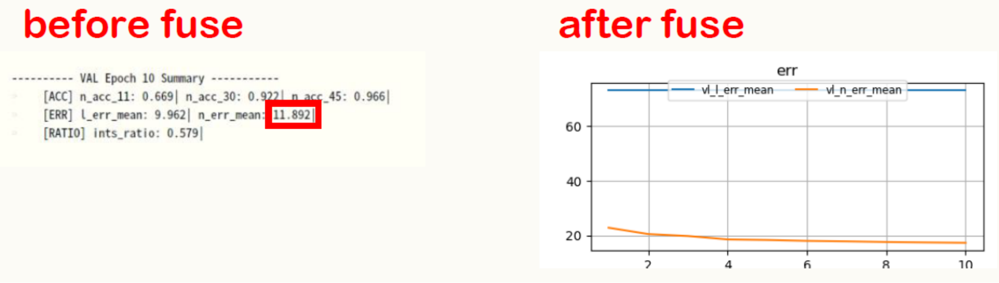
Figure 1. The network architecture of SDPS-Net is composed of (a) Lighting Calibration Network and (b) Normal Estimation Network. Kernel sizes for all convolutional layers are 3 × 3, and values above the layers indicate the number of feature channels.

Figure 14: FusedNet

After achieving the goal of reducing the parameters of LCNet and NENet, the next goal was to integrate the two models into one model. Since it is difficult to obtain accurate information about the light source from a photograph obtained from nature, it may be difficult to learn when using LCNet in practice. So, LCnet and NeNet are integrated so that the model can be trained without knowing the exact information of the light source. The code of the joint part of the two models was modified, and the code was modified so that both models were trained during training.

The Figure 15. is the result of simply merging the two models and training them. As you can see, the average error increased by about 7 points. Here, a residual block is used to further reduce errors. A residual block is a block that adds an input to the output of the convolution. In addition to residual blocks, depthwise separable convolution, the previously used method, was also applied to reduce parameters.

The Figure 16. is the result of applying residual block and depthwise separable convolution in fusedNet. It can be seen that the average error point increased by 4 points compared to the existing fusedNet. The parameters were reduced to 1/8 level. In order to recover the lost accuracy, we tried to increase the layers of the model more or increase the channels, but it did not recover. So, we tried to implement fusedNet, but it did not show the performance enough for practical use.[2]



```
---------- VAL Epoch 10 Summary -----------
    [ACC] n_acc_11: 0.669| n_acc_30: 0.922| n_acc_45: 0.966|
    [ERR] l_err_mean: 9.962| n_err_mean: 11.892|
    [RATIO] ints_ratio: 0.579|
```

Figure 15: Result of applying FusedNet

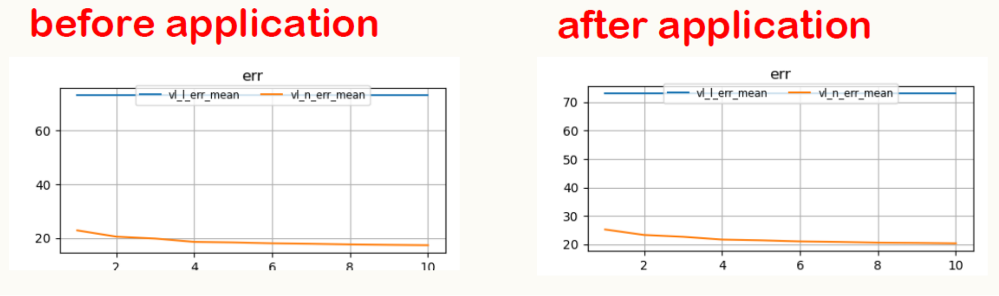


Figure 16: Result of applying depthwise separable convolution and residual block

9 Real Life Application

In order to train through real-world photographs, an `ICT-3DFRE` dataset containing a normal map was required. However, it was not obtained. Instead, `BaselFaceModel` dataset was obtained, but it also did not contain a normal map and could not be used by trainig.
Instead, the test was tried as a human photograph, and the original model used `DPR`[4] model to make multiple photographs using lighting from different angles. It is a single photo, but it roles like several different angle pictures. The results obtained from this are as follows.

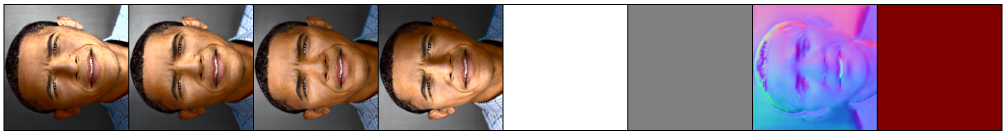


Figure 17: Result of real portrait picture with original SDPS

10 Conclusion & Future Work

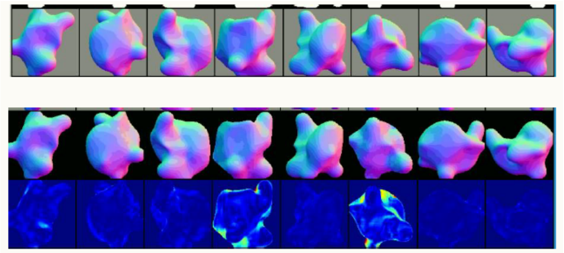


Figure 18: Original model

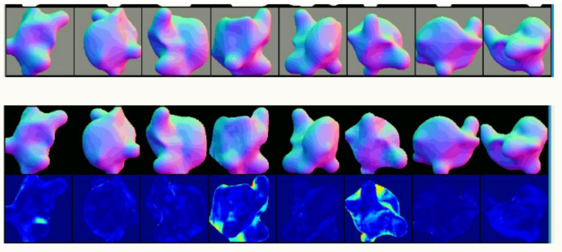


Figure 19: Our model

We have successfully reduced significant amount of the number of parameters, while maintaining a small drop in accuracy by modifying the original model by applying depthwise separable convolution and residual block on NENet. Furthermore, we successfully attached LCNet and NENet to a single model, FusedNet, that can solely be trained. Although FuseNet shows poor performance, especially in terms of accuracy, but given that it doesn't need any light direction information for training, it can be said that the accuracy drop is quite worth the tradeoff. However, the accuracy is still not tolerable for use in real life, thus additional future work on improving accuracy is still required.

11 Datasets & Library

- DiLiGenT (https://sites.google.com/site/photometricstereodata/single?authuser=0)
- Gourd & Apple Dataset (http://vision.ucsd.edu/ nalldrin/research/cvpr08/datasets/)
- Light Stage Data Gallery Dataset (https://vgl.ict.usc.edu/Data/LightStage/)
- Basel Face Model(3D face dataset) (https://faces.dmi.unibas.ch/bfm/)
- Library
  - cuda : v11.2
  - pytorch : v1.8.0
  - numpy : v1.19.2
- Running Environments
  - OS : Ubuntu 18.04
  - CPU : Intel(R) Core(TM) i7-9700K @3.6GHz
  - GPU : Titan RTX

References

[1] Guanying Chen, Kai Han, Boxin Shi, Yasuyuki Matsushita, and Kwan-Yee K Wong. Self-calibrating deep photometric stereo networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8739–8747, 2019.

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[3] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[4] Hao Zhou, Sunil Hadap, Kalyan Sunkavalli, and David W. Jacobs. Deep single portrait image relighting. In *International Conference on Computer Vision (ICCV)*, 2019.