Label Denoising Adversarial Network (LDAN) for Inverse Lighting of Faces Supplementary Material

Hao Zhou * Jin Sun* Yaser Yacoob David W. Jacobs
University of Maryland, College Park, MD, USA
{hzhou, jinsun, yaser, djacobs}@cs.umd.edu

1. Spherical Harmonics

The 9 dimensional spherical harmonics in Cartesian coordinates of the surface normal $\vec{n} = (x, y, z)$ are:

$$
\begin{aligned}
Y_{00} &= \tfrac{1}{\sqrt{4\pi}} & Y_{10} &= \sqrt{\frac{3}{4\pi}}z \\
Y_{11}^e &= \sqrt{\tfrac{3}{4\pi}}x & Y_{11}^o &= \sqrt{\frac{3}{4\pi}}y \\
Y_{20} &= \tfrac{1}{2}\sqrt{\tfrac{5}{4\pi}}(3z^2-1) & Y_{21}^e &= 3\sqrt{\frac{5}{12\pi}}xz \\
Y_{21}^o &= 3\sqrt{\tfrac{5}{12\pi}}yz & Y_{22}^e &= \frac{3}{2}\sqrt{\frac{5}{12\pi}}(x^2-y^2) \\
Y_{22}^0 &= 3\sqrt{\tfrac{5}{12\pi}}xy
\end{aligned}
\tag{1}
$$

2. Quantitative Evaluation of LDAN on different lighting conditions

In this section, we show the quantitative evaluation of LDAN under different lighting conditions. We take the lighting condition shown in Figure 4 in our paper as example. In classification we expect extreme lighting to be differentiated more easily than normal lighting because equal changes in the angle of lighting directions affect frontal lighting less than side lighting under the Lambertian model. Table 1 shows the top-1 classification results of LDAN and SIRFS for faces under four illumination conditions corresponding to Figure 4 (a) to (d) in the original submission. As expected, under extreme side lighting (i.e., (b) and (d)) or dark lighting (i.e., (a)), LDAN predicts very consistent lightings that lead to good performance. Classifications in normal lighting (c) are harder for both methods, but LDAN is far superior to SIRFS.

3. Details of Model B and Model C

In this section, we show the details of Model B and Model C. Figure 1 (a) and (b) illustrates the structure of

*means equal contribution.

Table 1. Top-1 classification results of different lightings.

	Dark (a)	Side (b)	Norm (c)	Side (d)
LDAN %	**99.80**	73.90	**61.37**	**71.81**
SIRFS %	96.40	**76.31**	34.14	50.60

Model B and Model C respectively. The objective function of training Model B and Model C is the same with that of training the LDAN:

$$
\begin{aligned}
&\min_{\mathcal{R},\mathcal{S},\mathcal{L}} \max_{\mathcal{D}} \underbrace{\sum_i (\mathcal{L}(\mathcal{R}(\mathbf{r}_i)) - \hat{\mathbf{y}}_{ri})^2}_{\text{regression loss for real}} \\
&+ \underbrace{\mu\, \mathbb{E}_{\mathcal{S}(\mathbf{s})\sim\mathbb{P}_s}[\mathcal{D}(\mathcal{S}(\mathbf{s}))] - \mathbb{E}_{\mathcal{R}(\mathbf{r})\sim\mathbb{P}_r}[\mathcal{D}(\mathcal{R}(\mathbf{r}))]}_{\text{adversarial loss}} \\
&+ \sum_{(i,j)\in\Omega} \underbrace{(\nu[(\mathcal{L}(\mathcal{S}(\mathbf{s}_i)) - \mathbf{y}_{si}^*)^2 + (\mathcal{L}(\mathcal{S}(\mathbf{s}_j)) - \mathbf{y}_{si}^*)^2]}_{\text{regression loss for synthetic}} \\
&+ \underbrace{\lambda(\mathcal{S}(\mathbf{s}_i) - \mathcal{S}(\mathbf{s}_j))^2)}_{\text{feature loss}},
\end{aligned}
\tag{2}
$$

where $\mu = 0.01$, $\lambda = 0.01$, and $\nu = 1$. Model B is inspired by [1]: real and synthetic data share the same feature net. Model C, on the other hand, is inspired by [6]: it defines two different feature nets for real and synthetic data. In a high level view, the difference between Model B/C and LDAN is that Model B/C tries to map lighting related features for synthetic and real data to a common space, which might be different from that learned with synthetic data alone, whereas LDAN tries to directly map lighting related features of real data to the space of synthetic data. This difference is illustrated in Figure 2.

Algorithm 1 shows the details of how to train Model B and C.

Algorithm 1 Training procedure for Model B/C

for number of iterations in one epoch **do**
  for k= 1 to 1 iterations **do**
    Sample 128 $\mathbf{s}$ and $\mathbf{r}$, train discriminator $\mathcal{D}$ through the following loss using RMSProp[3]:

$$\max_{\mathcal{D}} \mathbb{E}_{\mathcal{S}(\mathbf{s})\sim\mathbb{P}_s}[\mathcal{D}(\mathcal{S}(\mathbf{s}))] - \mathbb{E}_{\mathcal{R}(\mathbf{r})\sim\mathbb{P}_r}[\mathcal{D}(\mathcal{R}(\mathbf{r}))]$$

  end for
  for k=1 to 4 iterations **do**
    Sample 128 $\mathbf{s}$ and $\mathbf{r}$, train $\mathcal{L}$, $\mathcal{R}$ and $\mathcal{S}$ through Equation 2 by Adadelta.
  end for
end for

4. Network Structures For Lighting Regression

We show the structure of our networks in this section. As mentioned in the paper, we borrow the structure of ResNet [2] to define our feature net. Figure 4 (a) shows the details. A block like "Conv 3×3, 16" means a convolutional layer with 16 filters, the size of each filter is $3 \times 3 \times n$ where n is the number of input channels. This convolutional layer is followed by a batch normalization layer and a ReLU layer. A block like "Residual 3×3 32" means a residual block of two 3×3 convolutional layers with skip connections: each of the two convolutional layers has 32 filters and is followed by batch normalization and ReLU layer. ",/2" means the stride of the first convolutional layer in residual block is 2. The output of the feature net is a 128 dimensional feature.

Figure 4 (b) shows the structure of the lighting net. "FC ReLU 128" means a fully connected layer whose number of outputs is 128 followed by a ReLU layer. "Dropout" means a dropout layer with dropout ratio being 0.5. "FC, 18" means a fully connected layer with 18 outputs.

Figure 4 (c) shows the structure of the discriminator. "FC tanh, 1" means a fully connected layer with one output followed by a tanh layer.

5. Details of Keypoints Regression

We resize each of the images from the dataset to be 256×256. Following [5], we normalize the keypoint location by the width and height of the image, so that the (x, y) coordinates of each 2D keypoint are within $[0, 1]$. As illustrated in Figure 3, similar to the lighting regression network, our keypoints regression network contains two parts: a feature net and a regression net. Inspired by [7], we define a separate regression network for each keypoint, resulting in 14 different regression networks. Our feature net takes a 256×256 image as input and outputs a $16 \times 16 \times 512$ tensor as the feature vector. Our regression network takes this feature as input and predicts the 2D location of the corresponding keypoint. Figure 5 (a), (b), and (c) shows the structure of the feature net, regression net and discriminator separately. The notion of each block is the same as those in Figure 4.

Similar to the lighting regression formulation, let us denote $\mathcal{S}$ as the feature network for synthetic data, $\mathcal{R}$ as the feature network for real data, and $\mathcal{L}$ as the regression network. Specifically, we use $\mathcal{L}_j$ to represent the regression network for the j-th keypoint, where $j = 1, 2, ..., 14$. Using $(\mathbf{s}_i, \mathbf{y}^*_{si})$ as (data, ground truth label) pair for synthetic data, and y^{j*}_{si} to represent the ground truth location of the j-th keypoint. We train the feature net $\mathcal{S}$ and regression net $\mathcal{L}$ using the following loss function:

$$\min_{\mathcal{S},\mathcal{L}} \sum_{i,j} (\mathcal{L}_j(\mathcal{S}(\mathbf{s}_i)) - \mathbf{y}^{j*}_{si})^2 \tag{3}$$

After training $\mathcal{S}$ and $\mathcal{L}$, we fix $\mathcal{S}$ and $\mathcal{L}$ and train the feature net $\mathcal{R}$ with real data together with a discriminator $\mathcal{D}$. Suppose $\mathbf{r}_i$ represents the i-th real data and $\hat{\mathbf{y}}_{ri}$ represents its corresponding noisy label. More specifically, letting $\hat{\mathbf{y}}^j_{ri}$ represents the location of j-th keypoint, we train $\mathcal{R}$ and $\mathcal{D}$ using the following loss function:

$$\min_{\mathcal{R}} \max_{\mathcal{D}} \underbrace{\sum_{i,j} (\mathcal{L}_j(\mathcal{R}(\mathbf{r}_i)) - \hat{\mathbf{y}}^j_{ri})^2}_{\text{regression loss for real}} + \underbrace{\mu\,(\mathbb{E}_{\mathcal{S}(\mathbf{s})\sim\mathbb{P}_s}[\mathcal{D}(\mathcal{S}(\mathbf{s})] - \mathbb{E}_{\mathcal{R}(\mathbf{r})\sim\mathbb{P}_r}[\mathcal{D}(\mathcal{R}(\mathbf{r}))])}_{\text{adversarial loss}} \tag{4}$$

where $\mu = 0.05$ and $\mathbb{P}_s$ and $\mathbb{P}_r$ are the distributions of keypoints related features for synthetic and real images respectively.

We train $\mathcal{S}$ and $\mathcal{L}$ for 50 epochs using SGD. We set the learning rate to 0.05, momentum to 0.9, and batch size to 64. While training $\mathcal{R}$ and $\mathcal{D}$, we train $\mathcal{D}$ using RMSProp [3] and $\mathcal{R}$ using ADAM [4] alternatively. $\mathcal{D}$ is trained for one mini batch while $\mathcal{R}$ is trained for three mini batches in one iteration with batch size of 64. We train $\mathcal{D}$ and $\mathcal{R}$ for 75 epochs.

References

[1] Y. Ganin and V. Lempitsky. Unsupervised domain adaptation by backpropagation. In *ICML*, pages 1180–1189, 2015.

[2] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[3] G. Hinton, N. Srivastava, and K. Swersky. Lecture 6a, overview of mini-batch gradient descent. `http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf`, 2012.

[4] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[5] C. Li, Z. Zia, Q. huy Tran, X. Yu, G. D. Hager, and M. Chandraker. Deep supervision with shape concepts for occlusion-aware 3d object parsing. In *CVPR*, 2017.

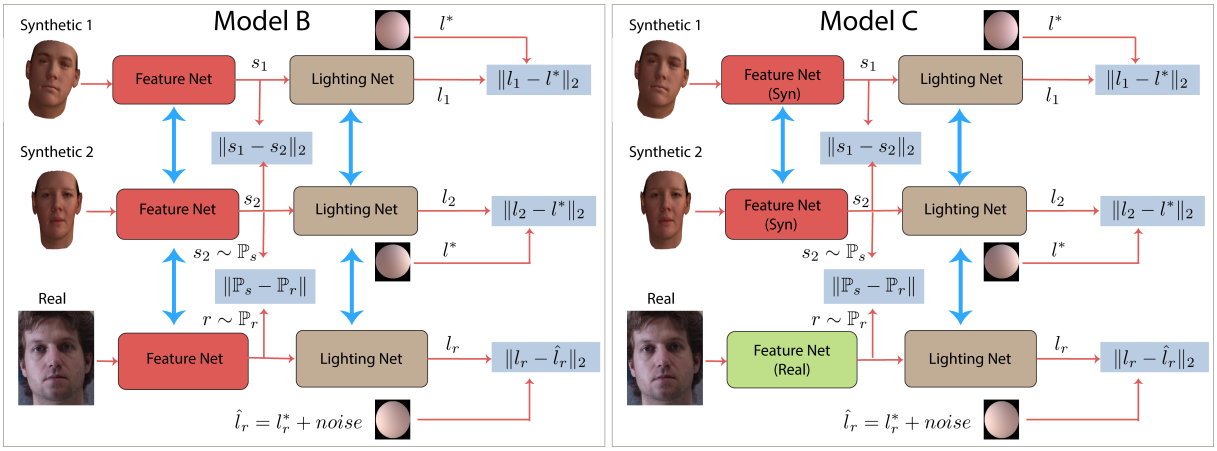
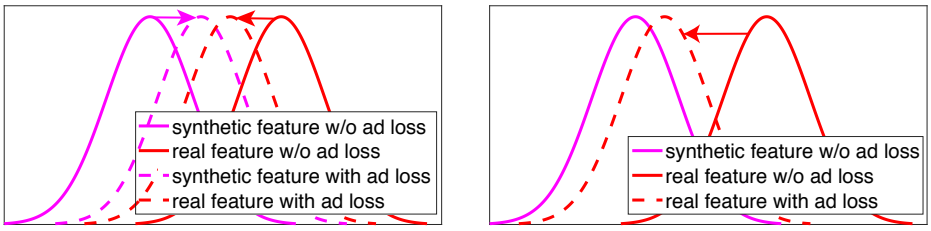


Figure 1. Two models we use to compare with the proposed LDAN. Different from LDAN, Model B uses the same feature net for synthetic and real data; Model C trains feature net for synthetic and real data together.



(a) Model B and C (b) LDAN

Figure 2. (a) illustrates the distribution of lighting related features of real and synthetic data before and after applying adversarial loss for Model B and Model C. With the adversarial loss, the distribution of lighting related features for real and synthetic data will move from the distribution illustrated by solid lines to that illustrated by dashed lines. (b) illustrates the distribution of lighting related features of real and synthetic data before and after applying adversarial loss for LDAN. Since the synthetic networks is fixed, the adversarial loss will drag the distribution of lighting related features of real data from the solid line to the dashed line. Note: "ad loss" means adversarial loss.

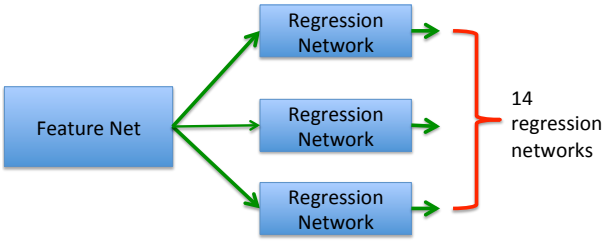
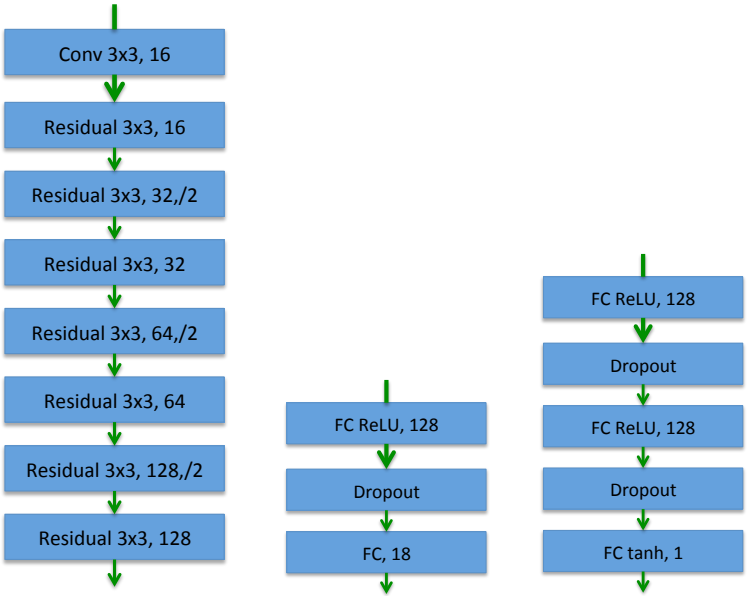


Figure 3. Illustration of the network for keypoints regression.

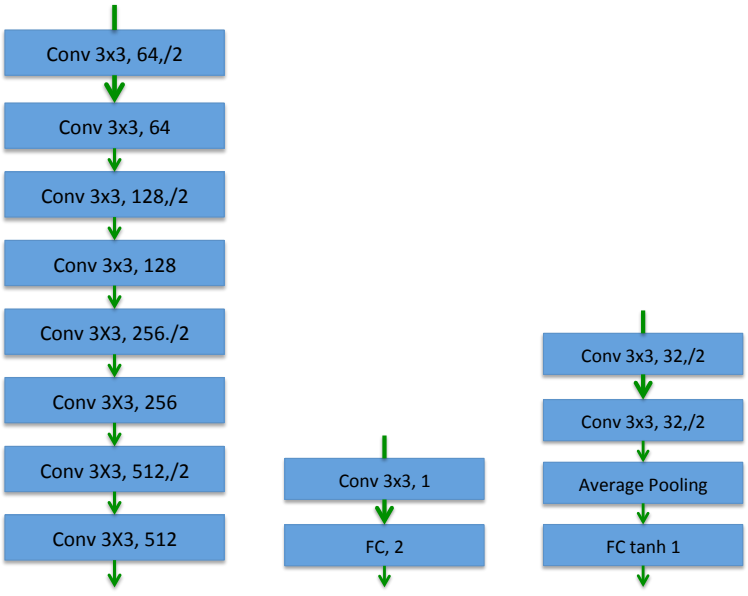
[6] K. Saito, Y. Mukuta, Y. Ushiku, and T. Harada. Demian: Deep modality invariant adversarial network. *ArXiv e-prints*, abs/1612.07976, 2016.

[7] J. Wu, T. Xue, J. J. Lim, Y. Tian, J. B. Tenenbaum, A. Torralba, and W. T. Freeman. Single image 3d interpreter network. In *ECCV*, 2016.



(a) Feature net (b) Lighting net (c) Discriminator

Figure 4. (a), (b) and (c) show the structure of feature net, lighting net, and discriminator used in our paper.



(a) Feature net (b) Regression net (c) Discriminator

Figure 5. (a), (b) and (c) show the structure of feature net, keypoint regression net, and discriminator used in our paper.