

Accurate and robust face recognition from RGB-D images with a deep learning approach:

Supplementary material

Yuancheng Lee

<http://cv.cs.nthu.edu.tw/php/people/profile.php?uid=150>

Jiancong Chen

<http://cv.cs.nthu.edu.tw/php/people/profile.php?uid=153>

Ching-Wei Tseng

<http://cv.cs.nthu.edu.tw/php/people/profile.php?uid=156>

Shang-Hong Lai

<http://www.cs.nthu.edu.tw/~lai/>

Computer Vision Lab,

Department of

Computer Science,

National Tsing Hua

University,

Hsinchu, Taiwan

1 Depth Face Image Recovery and Enhancement

1.1 Problem analysis

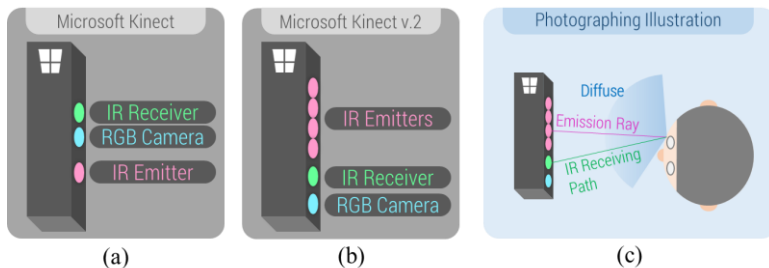


Figure 1: Simple sketch of Kinect (a), and Kinect v.2 (b).
Illustration of photographing a face (c).

All the major causes of artefacts are summarized as follows:

1. **Occlusion/Shadow:** A surface whose incoming or reflection ray path is occluded will lose its depth information. (**Fig. 2(a)**)
2. **Absorption:** The IR ray is almost absorbed, which results in weak reflection and missing depth. (**Fig. 2(a)**)
3. **Visible Specular:** If the major composition of reflection is specular reflection, and can be captured by IR sensor, the pixels may result in severe depth error.

4. **Invisible Specular:** Diffuse is weak, so the pixels will lose its depth. (**Fig. 2(b)**)
5. **Indirect Reflection:** It causes severe depth error, too. (**Fig. 2(b)**)
6. **Gaussian Noise:** Captured depth of an ordinary pixel jitters with a slight error of a Gaussian-like distribution. (**Fig. 2(b)**)
7. **Interference:** Another IR light source, such as sunlight or fluorescent lamp, interferes IR of Kinect, which results in severe artefacts. (**Fig. 2(d)**)
8. **Rolling Shutter:** Asynchronous sensing results in distortion for moving objects.

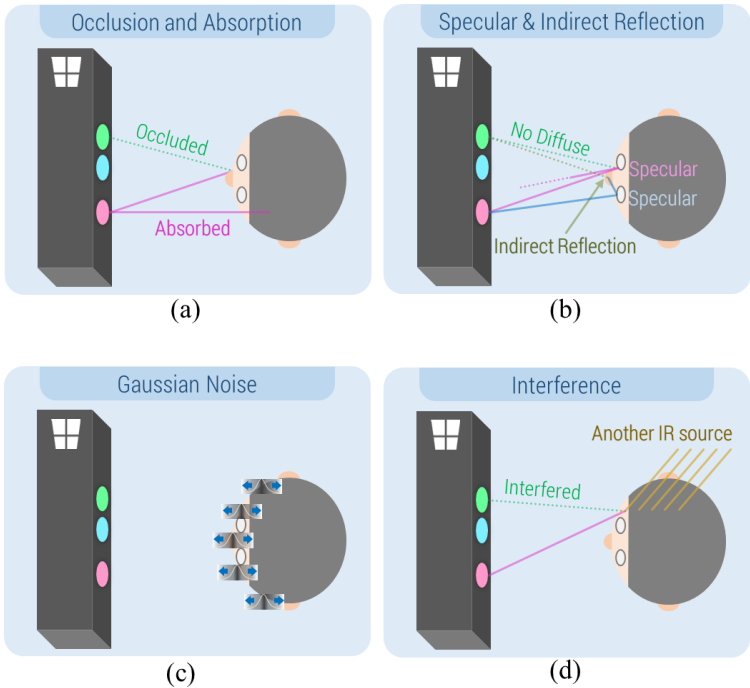


Figure 2: Illustration of all major causes of depth artefact and noise

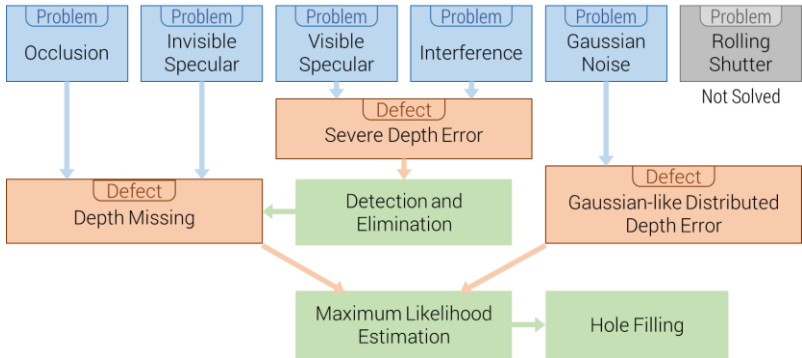


Figure 3: Problem, derivative defect, and solution flowchart for each defect

1.2 Implementation Details

Algorithm 1 contains the implementation details of 3-D face model reconstruction.

Algorithm 1. Forward Bilinear Interpolation with Depth-Buffering

- Input : 3-D point cloud (x_i, y_i, z_i)
 - Output : Rendered depth map \mathbf{D}
1. Initialize depth map \mathbf{D} , weight map \mathbf{W} , and depth-buffer \mathbf{B} as zeroes
 2. Define depth-threshold τ_d as 2 (millimeters)
 3. Transform 3-D points (x_i, y_i, z_i) to depth vectors (u_i, v_i, d_i)
 4. Sort depth vectors by their depth (d) in ascending order
 5. For each depth vector (u_i, v_i, d_i)
 6. For 4 nearest grid coordinates $\mathbf{g}_i^{(j)}, j = 1, 2, 3, 4$
 7. if $\mathbf{B}(\mathbf{g}_i^{(j)}) = 0 : \mathbf{B}(\mathbf{g}_i^{(j)}) = d_i$, jump to 10. (*First Filling*)
 8. else if $d_i \leq \mathbf{B}(\mathbf{g}_i^{(j)}) + \tau_d$: jump to 10. (*Merging*)
 9. else if $d_i > \mathbf{B}(\mathbf{g}_i^{(j)}) + \tau_d$: jump to 11. (*Neglecting*)
 10. $\mathbf{W}(\mathbf{g}_i^{(j)}) += w_i^{(j)}$, $\mathbf{D}(\mathbf{g}_i^{(j)}) += w_i^{(j)} \times d_i$
 , where $w_i^{(j)}$ is bilinear weighting coefficient
 11. End For
 12. End For
 13. For each pixel (m, n) in \mathbf{D} and $\mathbf{W} : \mathbf{D}(m, n) = \mathbf{D}(m, n) \div \mathbf{W}(m, n)$
-

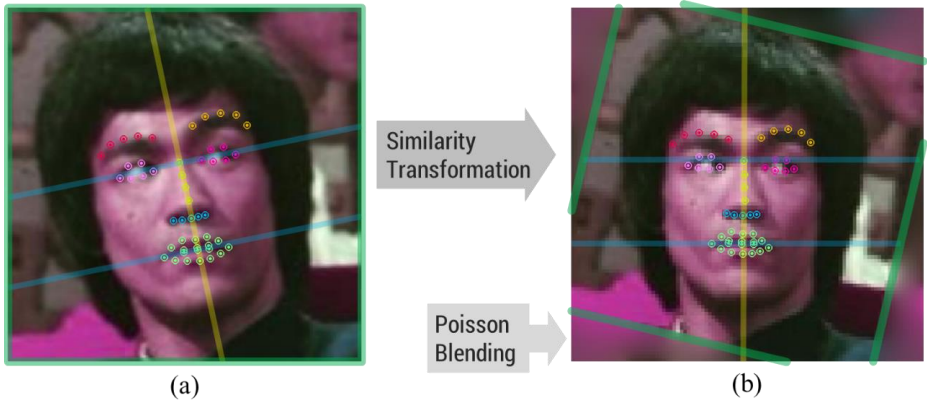


Figure 4: Example of color face alignment.

2 Distinguish Power on Images with Defects



Figure 5: Example of each artificial image defect setting
(a) original, (b) defect-1, and (c) defect-2

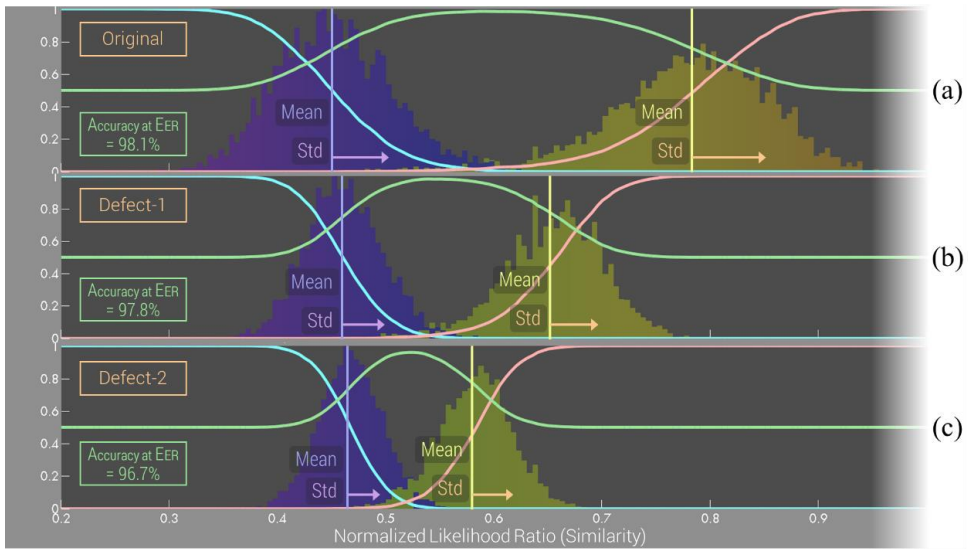


Figure 6: (1) Error rate / Accuracy – Similarity threshold chart,
(2) Histogram of estimated similarity (in different colours according to ground truth)
(3) mean and standard deviation of estimated similarity
on LFW validation set with proposed network and Joint Bayesian classifier
with artificial image defect settings (a) Original (b) Defect-1 (c) Defect-2
Purple bins are ground-truth different-subject pairs, Yellow bins are same-subject-pairs
Red lines are FAR curve. Blue lines are FRR curves. Green lines are accuracy curves.

In order to understand the impacts caused by low image quality, 3 artificial image defect settings are tested. Our artificial image defect by applying Gaussian noise of -20 dBW to an image and halving its brightness can lower the image quality. Defect-1 means for each pair, one of the images is with such a defect. (see Figure 30) Defect-2 means the quality of both images in each pair are reduced.

Similarity Statistics	Within-class : Same-subject pairs		Within-class : Different-subject pairs		Accuracy at EER
	mean	std	mean	std	
Original	0.7785	0.0743	0.4523	0.0540	98.1%
Defect-1	0.6539	0.0470	0.4612	0.0359	97.8%
Defect-2	0.5816	0.0347	0.4673	0.0291	96.7%

Table 1: Similarity statistics of each artificial defect settings

3 Data Preparation

3.1 Color Datasets



Figure 7: Examples of aligned face images of CASIA-WebFace dataset

For each class (subject) S_i , we take 3 images for testing, and the others for training. For each class (subject) S_i with $N(S_i)$ face images before augmentation, our goal is to sample $N_{target}(S_i) = \max(200, (20N(S_i)^{0.5}))$ training images for same-subject pairs, and another $N_{target}(S_i)$ images (may be duplicated with those images for same-subject pairs) for different-subject pairs, by the following algorithm.

Algorithm 2. Asymmetric Augmentation and Sampling

1. For each class (subject) S_i with $N(S_i)$ face images
 2. $N_{target}(S_i)$ is the target image number to sample
 3. if $N_{target}(S_i) \leq N(S_i)$, jump to 8
 4. else if $5N_{target}(S_i) \leq N(S_i) < N_{target}(S_i)$: jump to 7
 5. else : jump to 6
 6. Apply augmentation : multiple-cropping
 7. Apply augmentation : scale jittering
 8. Apply augmentation : mirroring
 9. Randomly sample $N_{target}(S_i)$ images from the augmented images
 10. End For
-

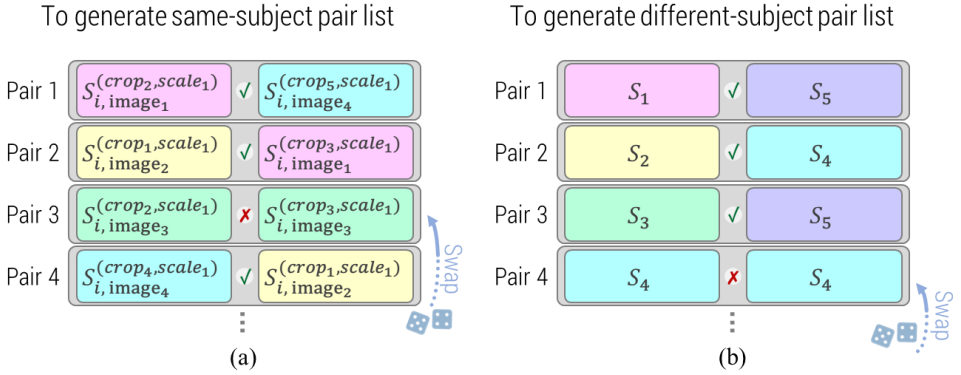


Figure 8: Illustration of the 2nd step of (a) same-subject pair list and (b) different-subject pair list generation

Besides, any image which is not sampled for same-subject pairs has higher priority to be sampled for different-subject pairs. For same-subject pairs, all the sampled images of each subject form a pool. For different-subject pairs, merging sampled images from all the subjects forms a big different-subject-pool.

To generate same-subject pair list for each subject, we randomly permute all the images in the pool and divide them into bipartite list first. Any pair with 2 image crops from the same image before augmentation is considered as illegal and need correcting. Next, we swap one of the image of each illegal pair with another (random) image in the list recursively, until there is no illegal pair. Please reference to **Fig. 8(a)** for better comprehension.

To generate different-subject pair list, we randomly permute all the images in the different-subject-pool first. Any pair with 2 image crops from the same subject is considered as illegal. Next, we correct it by swapping one of the image of each illegal pair with another (random) image in the list recursively, until there is no illegal pair. (**Fig. 8(b)**).

Last but not least, we merge and shuffle images in same-subject pair list and different-subject pair list, for better diversity within a mini-batch.

3.2 Depth / RGB-D Datasets

There is no single low quality (e.g. Kinect) depth face dataset which is large enough for our deep network, even with transfer learning, to learn a good model. For learning and evaluation, 5 depth datasets are considered. iiR3D, GavabDB: a 3D face database, Texas 3D Face Recognition Database, Eurocom Kinect Face Dataset(EKFD), Florence SuperFaces.

Since the quality of 3-D information in iiR3D, GavabDB, and Texas 3D Face Recognition Database is s better, our training dataset for learning deep representation is generated by merging these 3 datasets together. The rest 2 datasets, EKFD and SuperFaces, which are relatively small, are for evaluation.

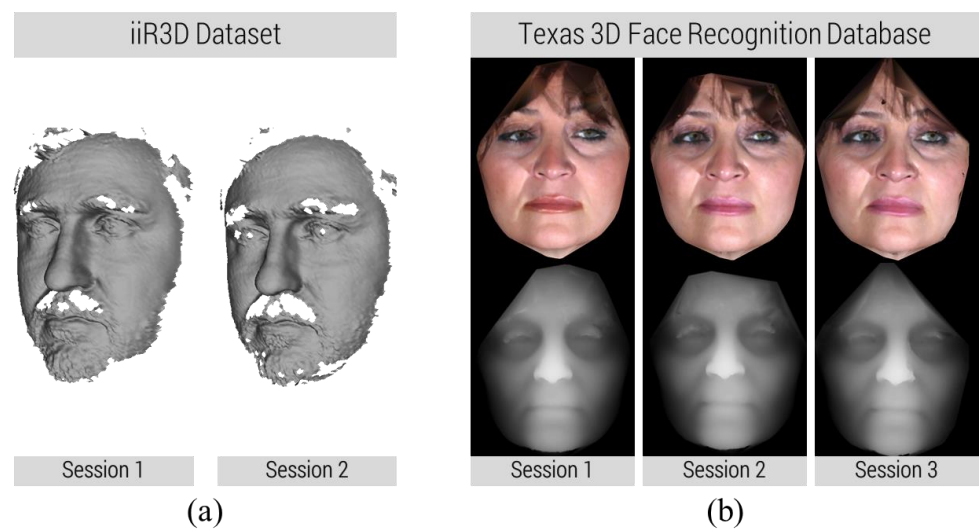


Figure 9: Visualization of example data from (a) iiR3D (3-D model) Dataset, and (b) Texas 3D Face Recognition (well aligned depth map) Database

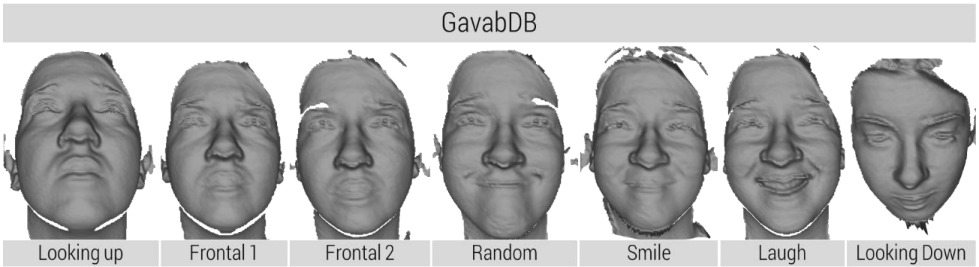


Figure 10: Visualization of example data from GavabDB (3-D model) Dataset

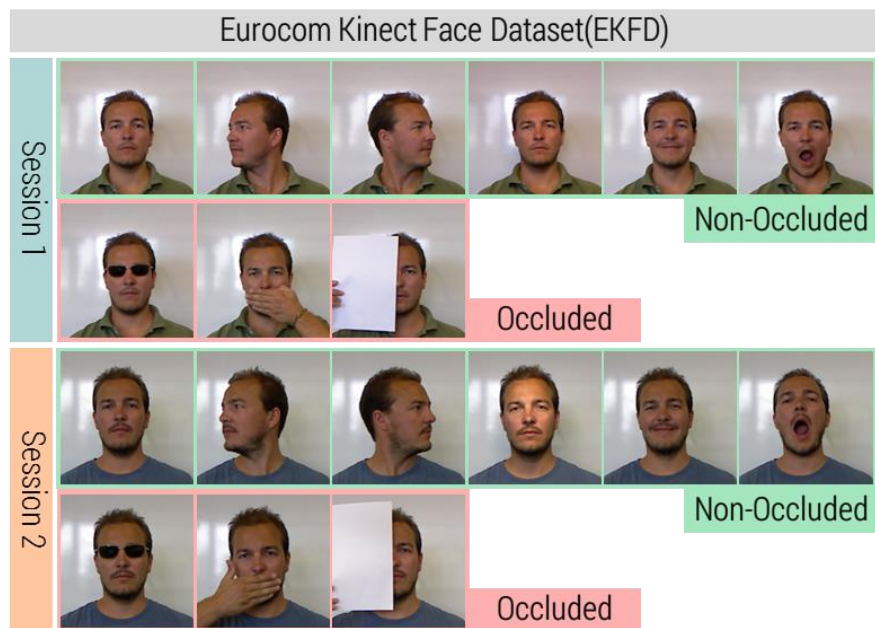


Figure 11: Example data from EKFD

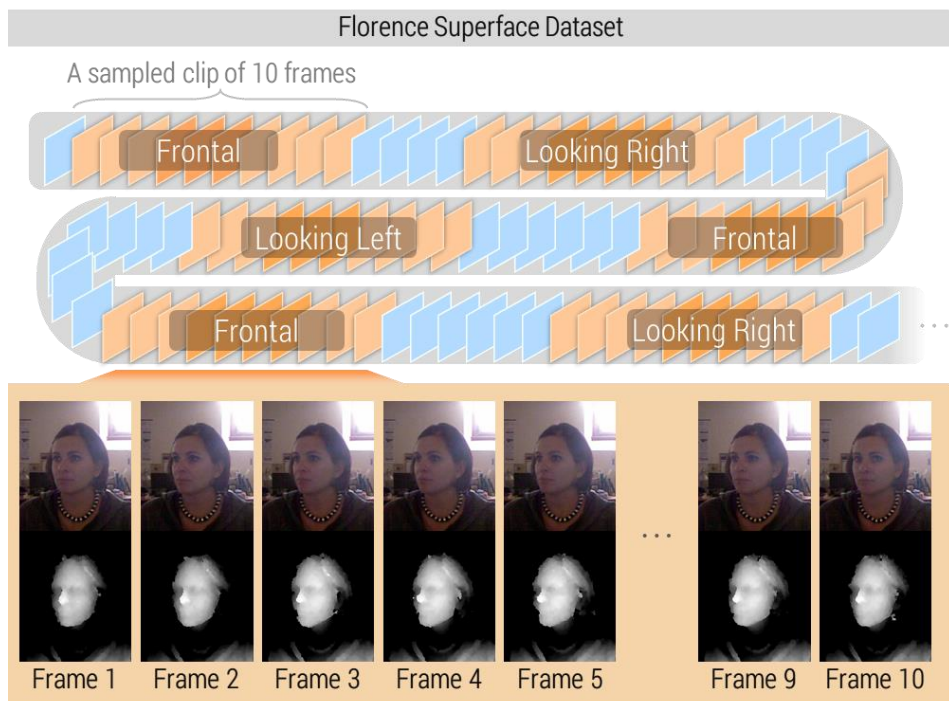


Figure 12: Visualization of sampling data from SuperFaces Dataset

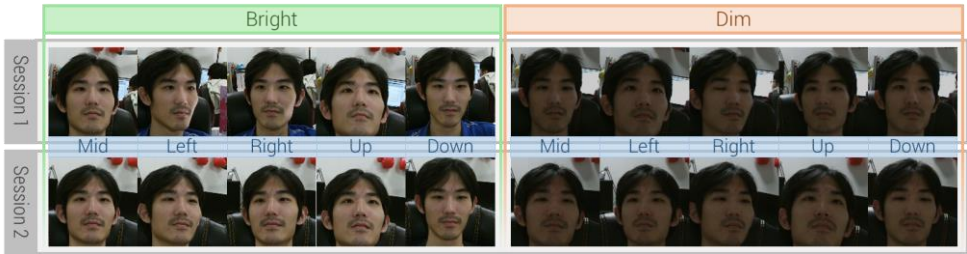


Figure 13: Example of one subject of our dataset

4 Implementation Details

4.1 Solver Parameter Configuration

Our training procedure containing 2 phases. When we start training from scratch, we are in phase 1, loss values and identification accuracy are recorded every epoch, and learning rate r_{base}^1 keeps constant. Once we find our model converged, we switch to phase 2, in which our learning rate policy is “step”, that is, with base learning rate r_{base}^2 , step size s and step ratio γ , learning rate at iteration t is given by:

$$r_t = r_{base}^2 \times \gamma^{\lfloor t/s \rfloor} \quad (11)$$

Empirically, we set $r_{base}^1 = 0.01$, $r_{base}^2 = 0.001$, $s = 10$ epochs, and $\gamma = 0.1$ for best performance. To optimizing our deep network, we use basic SGD (*stochastic gradient descent*) solver, with momentum = 0.9 and weight decay = 0.0005.

4.2 Network Details

Layer Name	Filter Size / Padding	Output Shape	Number of Parameters
Data	---	128×128×3(1)	
Conv 1A	5×5 / 2	128×128×56	4.2(1.4)K
Conv 1B	3×3 / 1	128×128×48	24K
Conv 2A	3×3 / 1	64×64×72	31K
Conv 2B	3×3 / 1	64×64×64	41K
Conv 3A	3×3 / 1	32×32×96	55K
Conv 3B	3×3 / 1	32×32×80	69K
Conv 4A	3×3 / 1	16×16×160	115K
Conv 4B	3×3 / 1	16×16×120	173K
Conv 5A	3×3 / 1	8×8×240	259K
Conv 5B	3×3 / 1	6×6×200	432K
Conv 6A	3×3 / 0	2×2×320	576K
Conv 6B	2×2 / 0	1×1×320	410K
Full Connection		1×10575 (1×266)	3384K (85k)
Total (color image)			5.20M
Total (depth image)			1.90M

Table 2: Detail of each layer (excluding loss modules) in the proposed network