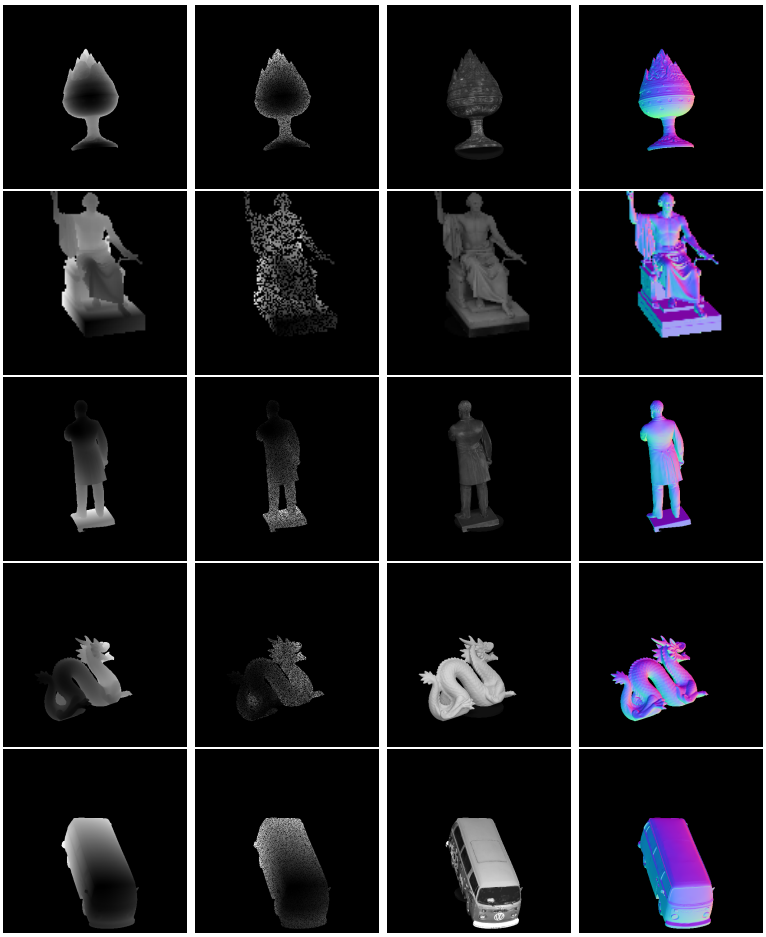
1 Training Details

The approaches are trained on dataset "synthetic-50-5" as mentioned in Chapter **??** with 3000 scenes. Each scene has a depth map with dimension $128 \times 128 \times$ in height, width and channel, an image with dimension $128 \times 128 \times 1$. The depth map is converted to 3D vertex map as introduced in Chapter **??**. The light map is calculated based on vertex map and the known light position. We create a tensor in PyTorch that includes vertex map, image and the light direction for each scene and considered it as one training case. Thus 3000 scenes has corresponding 3000 training cases. Each scene has a corresponding ground-truth normal map for loss calculation and the evaluation.



(a) Depth Map (b) Add noise (c) Image (d) Normal

Figure 1: Some of the test scenes during the training. From top to bottom, baoshanlu, Washington, Garfield, Dragon, Bus

The training processes are evaluated in every epochs with 29 evaluation scenes that model never seen before, which contains the 5 different objects in the "synthetic-50-5" test set. Figure 1 shows some of the test scenes during the training work. Note that the position of objects are not placed naturally on the stage but with a random rotation in X, Y, Z axes, respectively.

The training pipeline use batch size 8, Adam optimizer (**adam**), learning rate start from 1×10^{-3}, learning schedule [8,1000], learning decay factor 0.5. The model is trained with PyTorch 1.10.0a0, CUDA 11.4.1, GPU with single NVIDIA GEFORCE RTX 3090. It takes 14 hours to train GCNN and 35 hours to train the Trip-Net. We terminate the training when the evaluation on the test dataset converged.

2 GCNN model based on Geometry Information

The GCNN model is the base model of the whole thesis. The architecture is described in **??**. We use a single GCNN to estimate the surface normal based on geometry information. It uses vertex map as input to estimate the corresponding tangent surface normal map.

In order to verify the applicability of UNet architecture and Gated Convolution layer for the normal inference task, two similar models are created. We replace all of the gated layer to standard convolution layers in the network but keeps all of the other settings same in model "CNN". It is used to verify the performance the gated convolution layers. As mentioned in chapter **??**, the gated layer is designed to deal with noised input. Since all of the vertex map in the dataset has been added noise, the GCNN is supposed to over-perform "CNN". Another model called "NOC" is designed to verify the skip connection in the UNet, which simply removes the skip connections in the network but keeps other settings same. Is is designed to show the validation of skip connections. Figure 2 shows the training history on BerHu Loss. The GCNN approach achieves a lower loss from start to the end of the training.

3 Trip-Net model based on Calibrated Illuminated RGB-D Image

The Trip-Net model uses three times GCNN architecture with 4 times fusions, which is more difficult to train. It takes the calibrated illuminated RGB-D images as input to estimate the surface normal map. For the sake of comparison, we take the GCNN model as a baseline, to observe the beneficial of illuminated information using with Trip-Net architecture. Since it is more complicate than GCNN, we also explored the optimum fusion times of the Trip-Net to see any possibility for the model simplification. A set of similar models have been trained with same settings but different fusion times, denotes by Trip-Net-FxF,

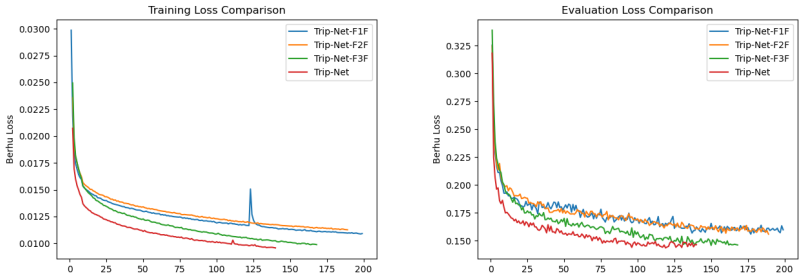


Figure 2: The training history of GCNN model. The line chart records the training BerHu loss of the model GCNN, NOC and CNN. The left shows the training loss history whereas the right one shows the evaluation loss history.

where x denotes the fusion times. We evaluate the fusion times from 1 to 4. For the learning rate. we set $1e-3$. It goes well with GCNN model but lead to loss explosion in Trip-Net. Thus we set a learning rate schedule with an extra decay step at epoch 8. The decay factor is 0.5. The batch size is chosen as 8.

Figure 3 shows the training history of these models on BerHu Loss. As shown in the loss figure, all of the four models has a reasonable learning rate. Trip-Net with four times fusion converges faster than others and also achieve a lower loss. Trip-Net-F3F converges slower than Trip-Net but achieved to a similar evaluation loss. The evaluation loss in Trip-Net-F1F and Trip-Net-F2F are relative higher than 3 or 4 times fusions model.

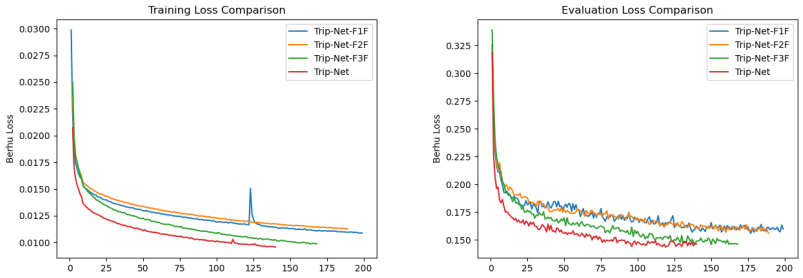


Figure 3: The training history of Trip-Net model. The line chart records the training BerHu loss of the model Trip-Net, Trip-Net-F1, Trip-Net-F2, Trip-Net-F3, GCNN.

However, the sacrifice of accuracy gives a relatively lighter model. Since we remove the fusions between different pipes, the corresponding upsampling layers in the image and light pipes can also be removed. The model can be trained faster and the size is reduced as well. Table **??** gives a comparison of the size and training time among different models.

Model	#Total	V-P	L-P	I-P	Size /MB	Time /h
Trip-Net-F1F	88	40	24	24	106	9.82
Trip-Net-F2F	92	40	26	26	137	7.35
Trip-Net-F3F	96	40	28	28	167	7.35
Trip-Net	100	40	30	30	198	9.15
GCNN	32	32	0	0	46	2.18

Table 1: Model information. Columns V-P, L-P and I-P represent the number of convolution layers in vertex pipe, light pipe and image pipe respectively. Note that one gated convolution layer is constructed with 2 standard layers, thus it is counted as 2.

4 Evaluation

Based on metrics proposed by **geometry˙based˙solution**, 6 different metrics are used for evaluation. Note that the input vertex map is only semi-dense. One of the benefit of GCNN architecture is the robust to the noisy input, thus in the evaluation, all the points including missing points in the input vertex map are taken into account.

Average Angle Error Metric The metric calculate the average angle error for each point between the inferred normal and ground-truth normal.

Median Angle Error Metric The metric calculate the median angle error of all the point in the normal map.

5 Degree Error Metric The metric calculate the percentage of the predicted normals that has error less than 5 degrees comparing to ground-truth.

11.5 Degree Error Metric The metric calculate the percentage of the predicted normals that has error less than 11.5 degrees comparing to ground-truth.

22.5 Degree Error Metric The metric calculate the percentage of the predicted normals that has error less than 22.5 degrees comparing to ground-truth.

30 Degree Error Metric The metric calculate the percentage of the predicted normals that has error less than 30 degrees comparing to ground-truth.

We evaluate the trained models on "synthetic-50-5". 5 objects are considered in the test dataset. They are: $Baoshanlu$, Bus, $Dragon$, $Garfield$, and $Washington$. Each object has 20 scenes with total 100 scenes for 5 objects. The test objects do not exist in the training dataset. We evaluate all the presented models on the test dataset, in order to fit them in one table, the name of each models are simplified. SVD model use SVD optimization method, NOC

model is the no skip connection version of $GCNN$, CNN is the CNN version of $GCNN$. $F1$, $F2$, $F3$,$F4$ means the fusion times in the Trip-Net.

Among all the columns, the most important models are $GCNN$ (depth map based) and $F4$ (calibrated illuminated RGB-D image based), which is the core result of this thesis. When evaluate the $GCNN$ models, we can take SVD model as baseline, NOC and CNN are used to verify the performance of $GCNN$ model. When evaluate the $F1-F4$ models, we can take $GCNN$ model as baseline.

Object	#	**SVD**	**GCNN**	**NOC**	**CNN**	**F1**	**F2**	**F3**	**F4**
Baoshanlu	20	35.66	11.09	13.58	15.55				9.82
Bus	20	31.93	7.79	8.95	11.93				7.32
Dragon	20	39.57	10.60	15.29	16.03				7.35
Garfield	20	39.69	10.20	12.50	14.46				9.15
Washington	20	42.83	13.43	17.59	18.71				12.13

Table 2: Average Angle Error of the evaluation dataset.

Object	#	**SVD**	**GCNN**	**NOC**	**CNN**	**F1**	**F2**	**F3**	**F4**
Baoshanlu	20	34.06	8.86	10.82	13.25				7.62
Bus	20	34.14	4.44	5.02	8.69				4.01
Dragon	20	36.43	7.62	11.10	13.26				5.06
Garfield	20	37.60	6.40	8.90	11.31				5.81
Washington	20	36.89	7.64	11.38	13.64				6.76

Table 3: Median Angle Error of the evaluation dataset.

Object	#	**SVD**	**GCNN**	**NOC**	**CNN**	**F1**	**F2**	**F3**	**F4**
Baoshanlu	20	0.01	0.25	0.18	0.11				0.31
Bus	20	0.00	0.56	0.50	0.23				0.59
Dragon	20	0.00	0.31	0.17	0.10				0.50
Garfield	20	0.00	0.41	0.27	0.14				0.45
Washington	20	0.00	0.38	0.26	0.10				0.41

Table 4: Percent of error less than 5 degree of the evaluation dataset.

Object	#	**SVD**	**GCNN**	**NOC**	**CNN**	**F1**	**F2**	**F3**	**F4**
Baoshanlu	20	0.03	0.62	0.52	0.41				0.69
Bus	20	0.05	0.81	0.78	0.65				0.83
Dragon	20	0.02	0.69	0.51	0.40				0.82
Garfield	20	0.03	0.72	0.62	0.51				0.75
Washington	20	0.02	0.62	0.50	0.40				0.66

Table 5: Percent of error less than 11.5 degree of the evaluation dataset.

Object	#	**SVD**	**GCNN**	**NOC**	**CNN**	**F1**	**F2**	**F3**	**F4**
Baoshanlu	20	0.18	0.90	0.84	0.79				0.92
Bus	20	0.26	0.93	0.91	0.89				0.94
Dragon	20	0.14	0.90	0.79	0.80				0.95
Garfield	20	0.13	0.89	0.86	0.84				0.91
Washington	20	0.14	0.81	0.72	0.72				0.84

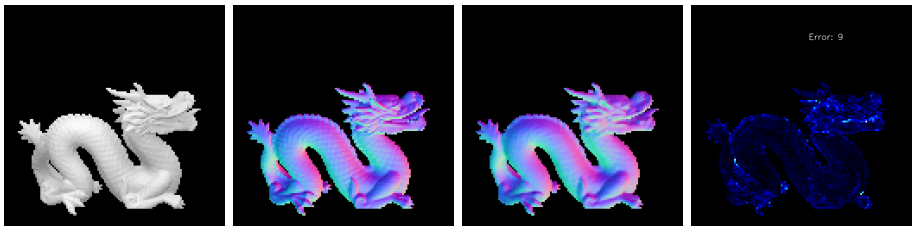
Table 6: Percent of error less than 22.5 degree of the evaluation dataset.

Object	#	**SVD**	**GCNN**	**NOC**	**CNN**	**F1**	**F2**	**F3**	**F4**
Baoshanlu	20	0.37	0.96	0.93	0.90				0.97
Bus	20	0.43	0.96	0.94	0.93				0.96
Dragon	20	0.30	0.95	0.88	0.90				0.98
Garfield	20	0.27	0.94	0.92	0.91				0.95
Washington	20	0.28	0.88	0.81	0.82				0.90

Table 7: Percent of error less than 30 degree of the evaluation dataset.

5 Visualization evaluation on GCNN model

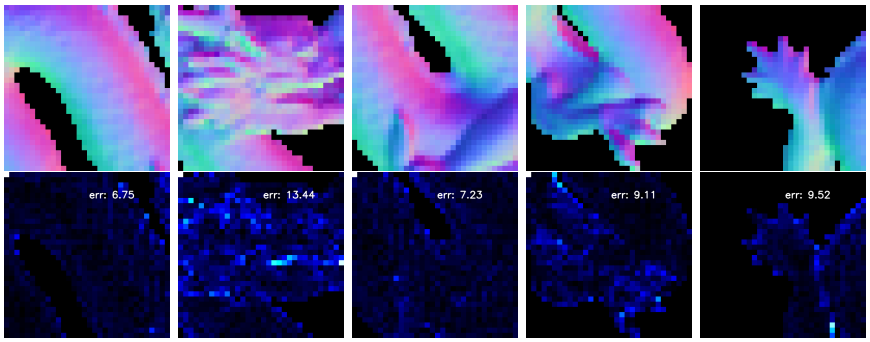
A qualitative evaluation on object "dragon" is shown in Figure 4. As shown in the figure, GCNN model achieved a mean angle error in 9 degrees on this dragon object. The image has an overall good performance on the whole object. A closer evaluation is shown in Figure 5, the normal accuracy especially good on the smooth surface, like the body area. In the same case, NOC model as shown in Figure 6 has a overall worse normal than GCNN model in the smooth area. CNN model keeps the skip connection thus gives a sharper result than NOC model, however, the overall smooth part of the model is still worse than GCNN. Besides, the sharp area like the hindleg, the head of dragon object, CNN model gives a much brighter error map (which lead to a higher angle error).



(a) Image (b) GT (c) Predict (d) Error

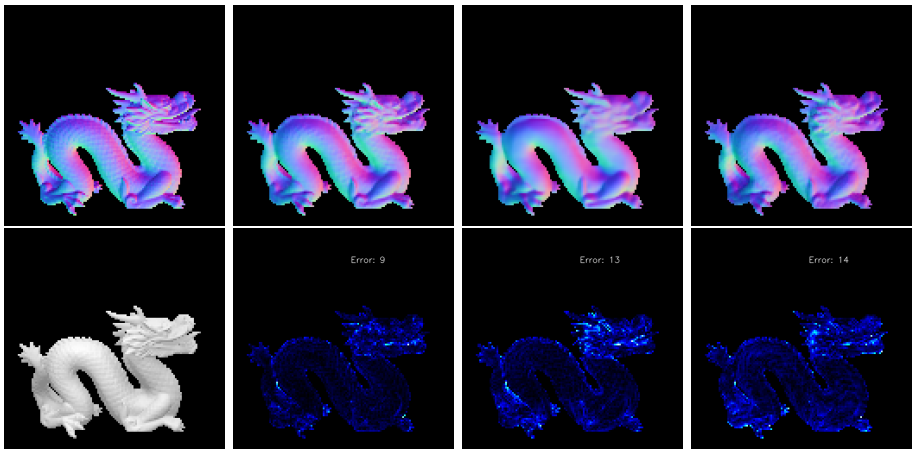
Error: 0 90

Figure 4: Normal inference based on GCNN. Test image has resolution 128×128



(a) Scale (b) Head (c) Foreleg (d) Hindleg (e) Tail

Figure 5: Zoom in of some regions of Dragon object. The first row is surface normal, the second row is the corresponding errors. Zoom-in normal map corresponding 32×32 points in the original matrix.



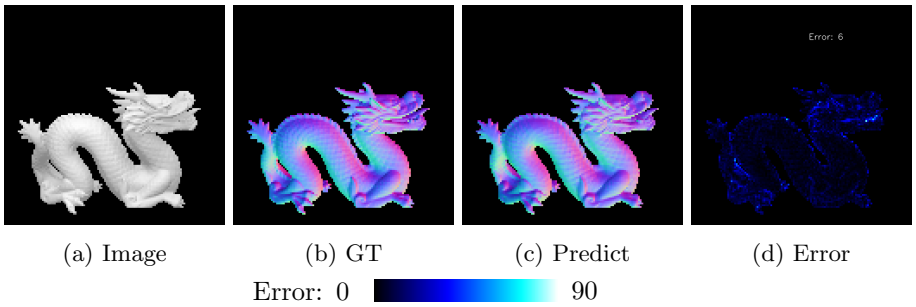
(a) GT (b) GCNN (c) NOC (d) CNN

Figure 6: Comparison of GCNN model with no skip connection version(NOC) and standard convolution layer only version (CNN). The second row is the corresponding mean average degree error.

6 Surface Normal Inference based on Calibrated Illuminated RGBD images

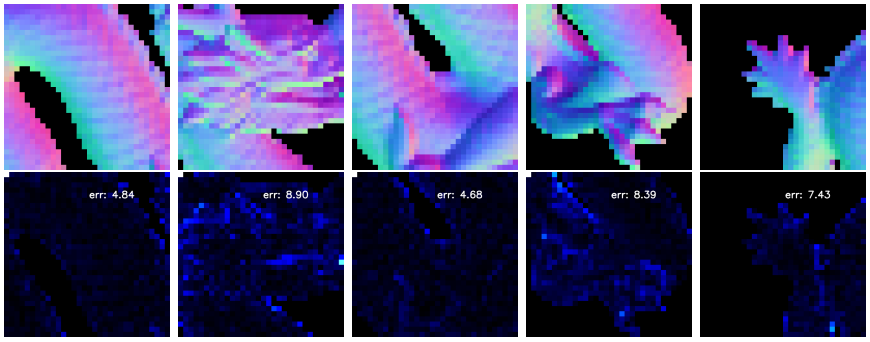
For the approach using illuminated calibrated RGBD image, the task is undertaken by Trip-Net introduced in **??**. The qualitative evaluation is shown in figure 7. In order to show the effectiveness of added illuminated information, the training settings for all the models are exact the same. We also use the same input as we did in GCNN evaluation. The error of Trip-Net is 6 degree whereas the GCNN is 9.

As shown in the figure 8, the scale on the dragon body is much easier to detect and close to the ground-truth. The head region gives a sharper edge prediction. All of the five sampled zoom-in regions in the Trip-Net has a better performance than GCNN.



(a) Image (b) GT (c) Predict (d) Error

Figure 7: Normal inference based on Trip-Net. Test image has resolution 128×128

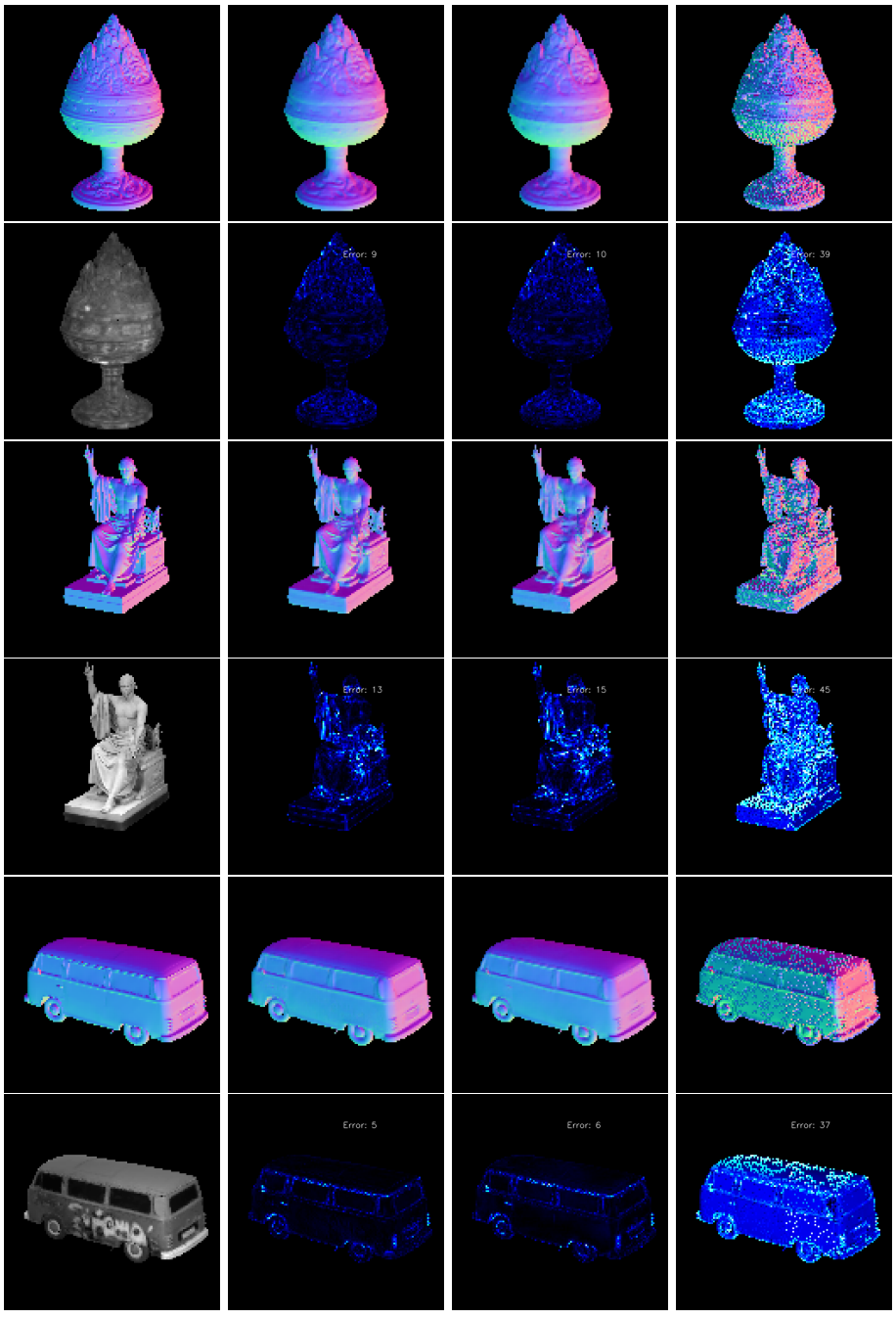


(a) Scale (b) Head (c) Foreleg (d) Hindleg (e) Tail

Figure 8: Zoom in of some regions of Dragon object. The first row is surface normal, the second row is the corresponding errors. Zoom-in normal map corresponding 32×32 points in the original matrix.

7 More Comparison

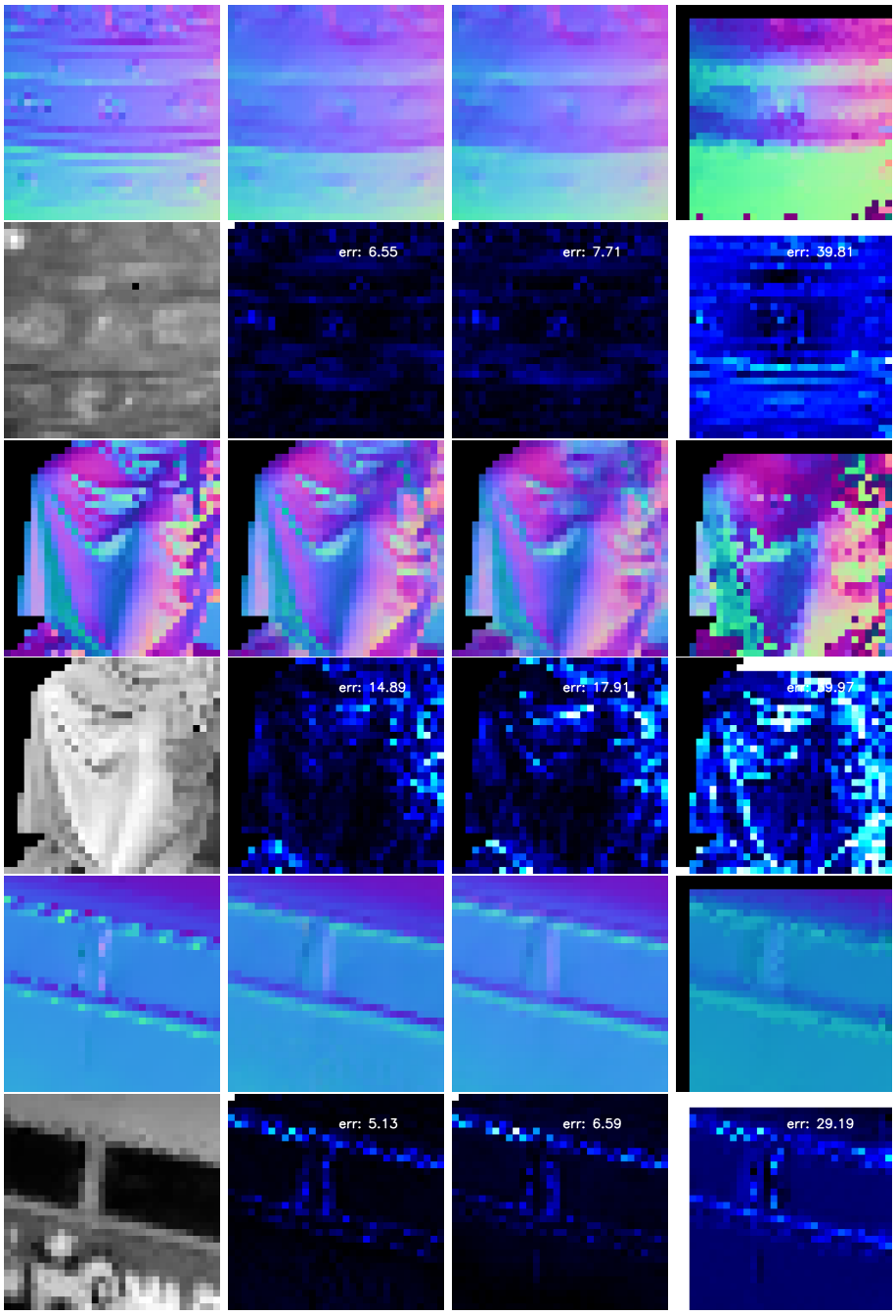
Figure 9 gives more visualization on model SVD, GCNN and Trip-Net. Figure 10 is the corresponding zoom-in visualization.



(a) GT (b) TripNet (c) GCNN (d) SVD

Error: 0 90

Figure 9: Evaluation on objects Baoshanlu, Washington statue, Bus(from top to bottom).



(a) GT (b) Trip-Net (c) GCNN (d) SVD

Figure 10: Zoom in of the center region of the objects in Figure 9