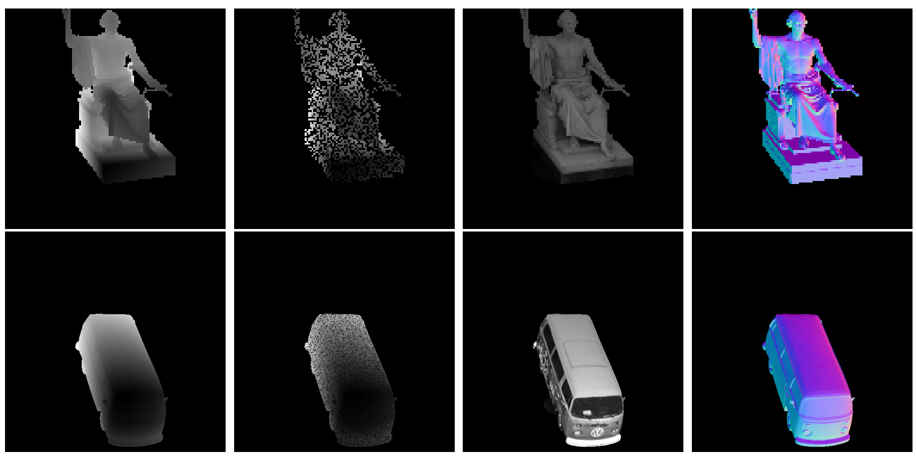
1 Training Details

The models are trained on dataset "synthetic-50-5" as mentioned in Chapter **??** with 3000 scenes. Each scene has a depth map with dimension 128×128 in height and width, 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 100 evaluation scenes that models 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 always naturally on the stage but with a random rotation in X, Y, Z axes, respectively.

For the training parameters, we set the training pipeline with batch size 8, we found that a higher batch size will reduce the final performance. 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 RTXA 6000. 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.

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 the skip connection and the gated convolution layers, we trained two extra models as a comparison. The first model, we replace all of the gated layer to standard convolution layers in the network but keeps all of the other settings same, and give it a name "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, which simply removes the skip connections in the network but keeps other settings same. Is is designed to show to which content will skip connection help the model performance. We use Reversed Huber Loss as loss function during the training, since we found it gives a better final error compare to L2 loss.

Model	**#Total**	**V-P**	**L-P**	**I-P**	**Size /MB**
CNN	17	32	0	0	25.5
NOC	32	32	0	0	30.6
GCNN	32	32	0	0	45.8

Table 1: GCNN 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.

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. When we train this model, 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-Fx, 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.

We also found that, trip-Net with four times fusion converges obviously faster than fewer fusion times model. However, model F3 with 3 times fusion converge slower than F4 but in the end it achieves a similar evaluation loss with F4(see Qualitative evaluation). The F1 and F2 models are relatively weaker than the other two models.

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 2 gives a comparison of the size and training time among different models.

Model	#Total	V-P	L-P	I-P	Size /MB
Trip-Net-F1F	88	40	24	24	106
Trip-Net-F2F	92	40	26	26	137
Trip-Net-F3F	96	40	28	28	167
Trip-Net	100	40	30	30	198

Table 2: Trip-Net 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.

2 Quantitative Evaluation

We evaluated our models on synthetic dataset with resolution 128×128. 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.

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.

From the table we can see that all the learning based approaches achieves a better result than SVD approach. In the 30 degrees error metric, GCNN based approach achieves 95 % accuracy whereas Trip-Net is even higher, some of the models like *dragon* achieves 98%. The best performance is around 90%, 75 %, 45 % in 22.5°, 11.5° and 5° degrees error metric respectively. Another notable result is the close performance of F3 and F4 models, where they achieves a very comparable performance. We also found that during the training, F4 model converges faster than F3, but F3 in the end achieves a similar loss with model F4. However, based on the30°, 22.5°, 11.5° metrics, we can still see that F4 model gives a more stable performance with less high error normals. This is reasonable since the last fusion in the original resolution provides more high resolution information to the model.

Object	#	**SVD**	**GCNN**	**NOC**	**CNN**	**F1**	**F2**	**F3**	**F4**
Baoshanlu	20	35.66	11.09	13.58	15.55	11.22	10.36	**9.77**	9.80
Bus	20	31.93	7.79	8.95	11.93	7.49	7.85	**7.30**	7.62
Dragon	20	39.57	10.60	15.29	16.03	10.47	10.23	8.16	**7.79**
Garfield	20	39.69	10.20	12.50	14.46	9.94	10.36	9.71	**9.39**
Washington	20	42.83	13.43	17.59	18.71	13.32	13.40	12.62	**12.60**

Table 3: 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	8.95	8.02	7.54	**7.50**
Bus	20	34.14	4.44	5.02	8.69	4.11	4.58	**3.65**	4.47
Dragon	20	36.43	7.62	11.10	13.26	7.60	7.12	5.87	**5.52**
Garfield	20	37.60	6.40	8.90	11.31	6.30	6.72	6.21	**6.04**
Washington	20	36.89	7.64	11.38	13.64	7.49	7.60	**7.03**	7.25

Table 4: 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	0.24	0.31	**0.33**	0.32
Bus	20	0.00	0.56	0.50	0.23	0.59	0.54	**0.63**	0.55
Dragon	20	0.00	0.31	0.17	0.10	0.31	0.34	0.43	**0.46**
Garfield	20	0.00	0.41	0.27	0.14	0.42	0.39	0.42	**0.43**
Washington	20	0.00	0.38	0.26	0.10	0.36	0.36	**0.40**	0.37

Table 5: 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	0.62	0.66	**0.69**	**0.69**
Bus	20	0.05	0.81	0.78	0.65	0.83	0.82	**0.83**	**0.83**
Dragon	20	0.02	0.69	0.51	0.40	0.70	0.71	0.79	**0.81**
Garfield	20	0.03	0.72	0.62	0.51	0.73	0.71	0.74	**0.75**
Washington	20	0.02	0.62	0.50	0.40	0.63	0.62	0.64	**0.65**

Table 6: 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	0.90	0.91	**0.92**	**0.92**
Bus	20	0.26	0.93	0.91	0.89	0.93	0.93	0.93	**0.94**
Dragon	20	0.14	0.90	0.79	0.80	0.90	0.90	0.94	**0.95**
Garfield	20	0.13	0.89	0.86	0.84	0.90	0.89	**0.91**	**0.91**
Washington	20	0.14	0.81	0.72	0.72	0.81	0.81	**0.83**	**0.83**

Table 7: Percent of error less than 22.5 degree of the evaluation dataset.

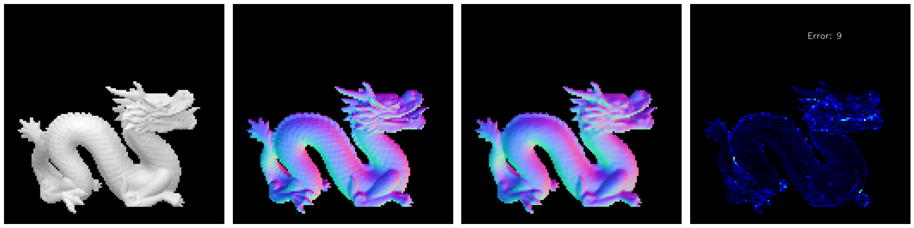
Object	**#**	**SVD**	**GCNN**	**NOC**	**CNN**	**F1**	**F2**	**F3**	**F4**
Baoshanlu	20	0.37	0.96	0.93	0.90	0.96	0.96	**0.97**	**0.97**
Bus	20	0.43	0.96	0.94	0.93	0.96	**0.96**	**0.96**	**0.96**
Dragon	20	0.30	0.95	0.88	0.90	0.95	0.95	0.97	**0.98**
Garfield	20	0.27	0.94	0.92	0.91	0.94	0.94	0.94	**0.95**
Washington	20	0.28	0.88	0.81	0.82	0.88	0.88	**0.89**	**0.89**

Table 8: Percent of error less than 30 degree of the evaluation dataset.

3 Visualization evaluation on GCNN model

A qualitative evaluation on object ”dragon” is shown in Figure 2. 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 3, the normal accuracy especially good on the smooth surface, like the body area. In the same case, NOC model as shown in Figure. 4 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).

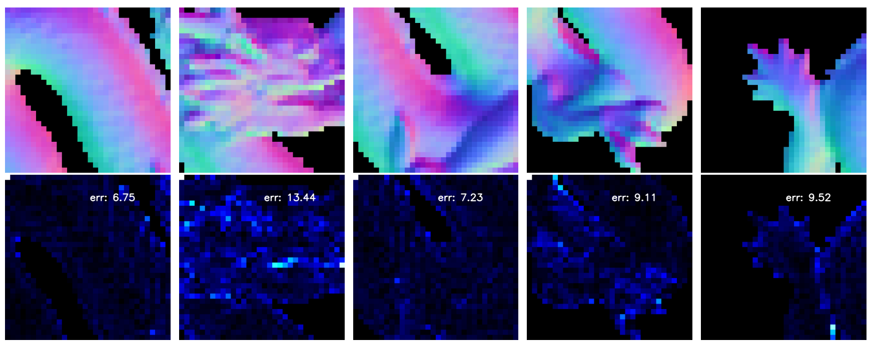
Figure 9 shows more evaluation on GCNN model. Although we can get a good result from GCNN model but from model “Washington” we still can see it lacks the sharpness in the detail area like the face and clothes area.



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

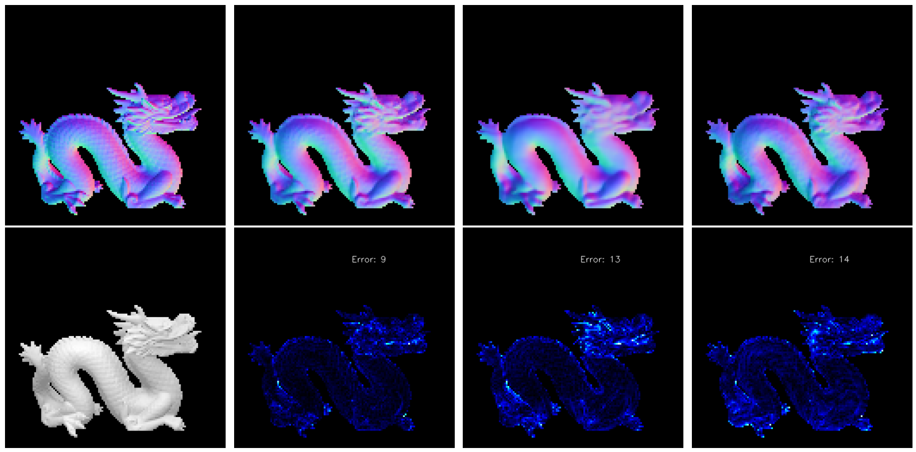
Error: 0 90

Figure 2: Normal inference based on GCNN. Test image has resolution 128×128



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

Figure 3: 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 4: 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.

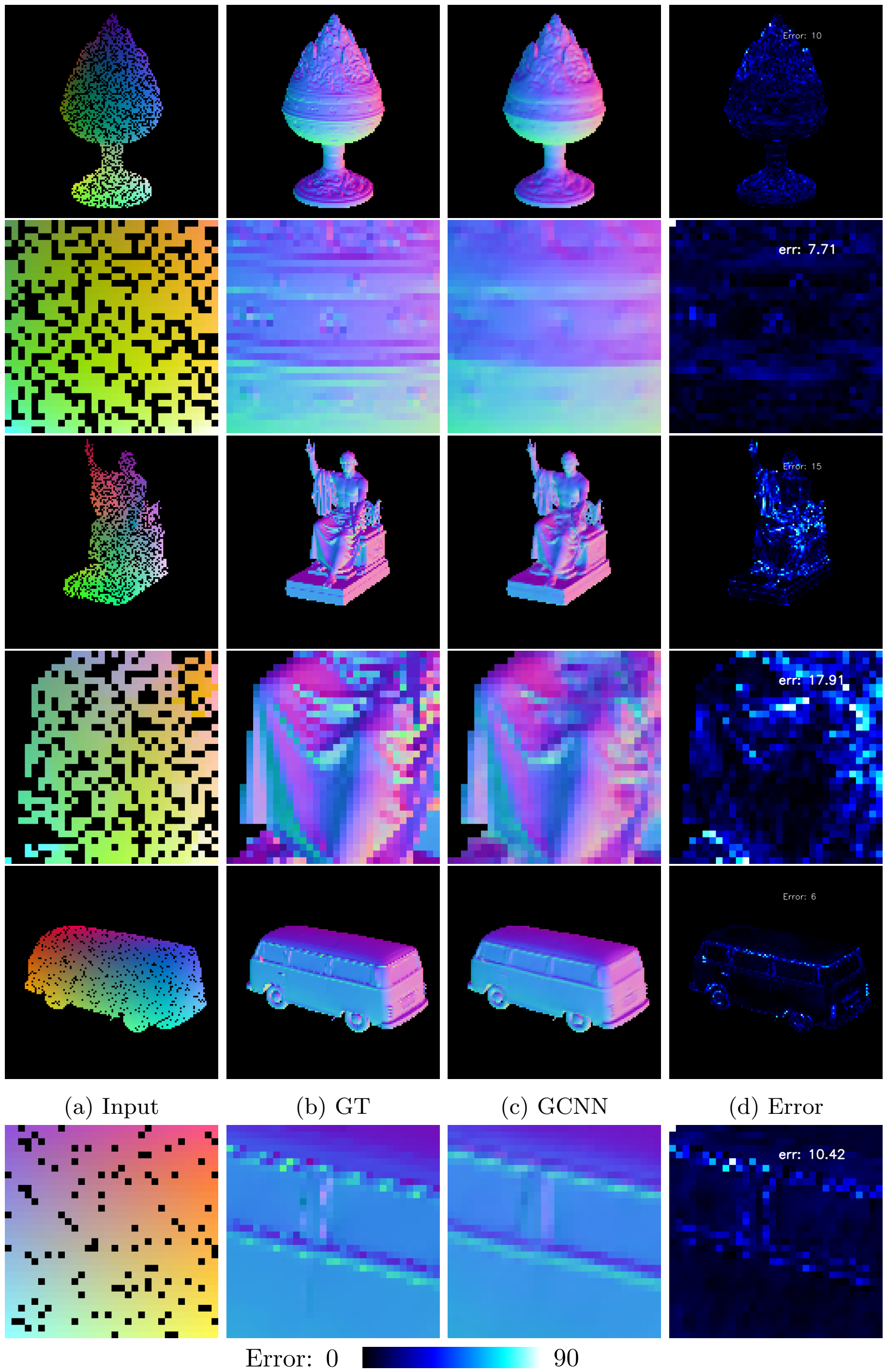
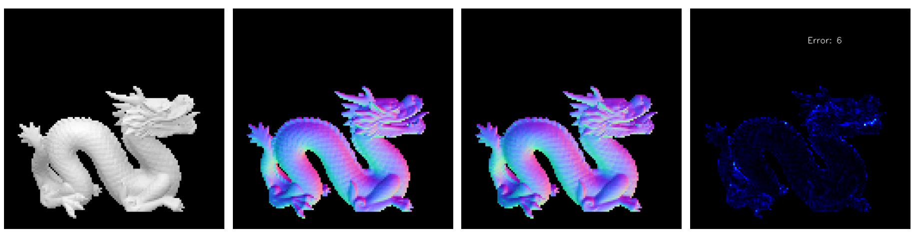


Figure 5: Evaluation on objects Baoshanlu, Washington statue, Bus(from top to bottom).

4 Visualized evaluation on Trip-Net models

For the approach using illuminated calibrated RGBD image, the task is undertaken by Trip-Net introduced in **??**. The Trip-Net uses illuminated information align with the geometry information achieves a sharper and more accurate result compare to the GCNN model. The qualitative evaluation is shown in figure 6. 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.

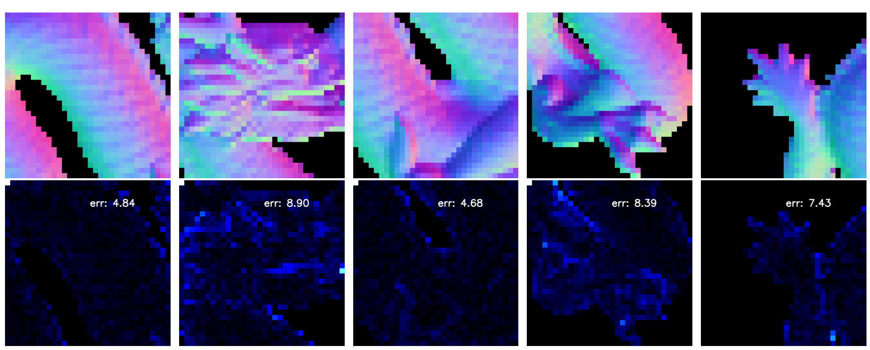
As shown in the figure 8, the scale on the dragon body is much easier to detect and also close to the ground-truth. The head region gives a sharper edge prediction. All of the five sampled zoom-in regions in the Trip-Net has a better performance than GCNN. It seems that the illumination helps the model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



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

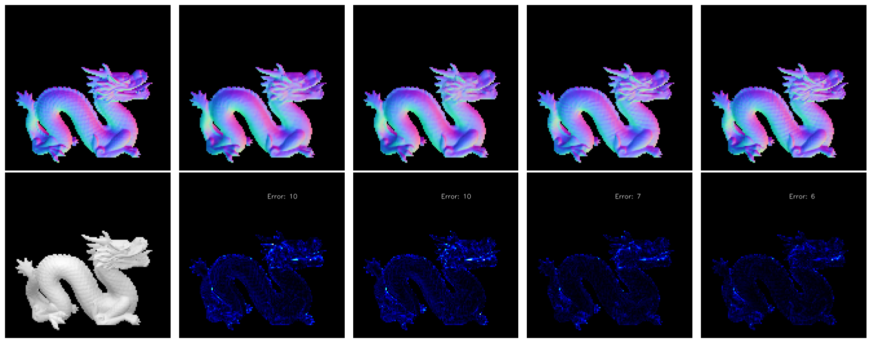
Error: 0 90

Figure 6: Normal inference based on Trip-Net. Test image has resolution 128×128



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

Figure 7: 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) F1 (c) F2 (d) F3 (e) F4

Figure 8: Comparison between different fusion times for Trip-Net. The first row is surface normal, the second row is the corresponding errors. The number in "Fx" represents the fusion times.

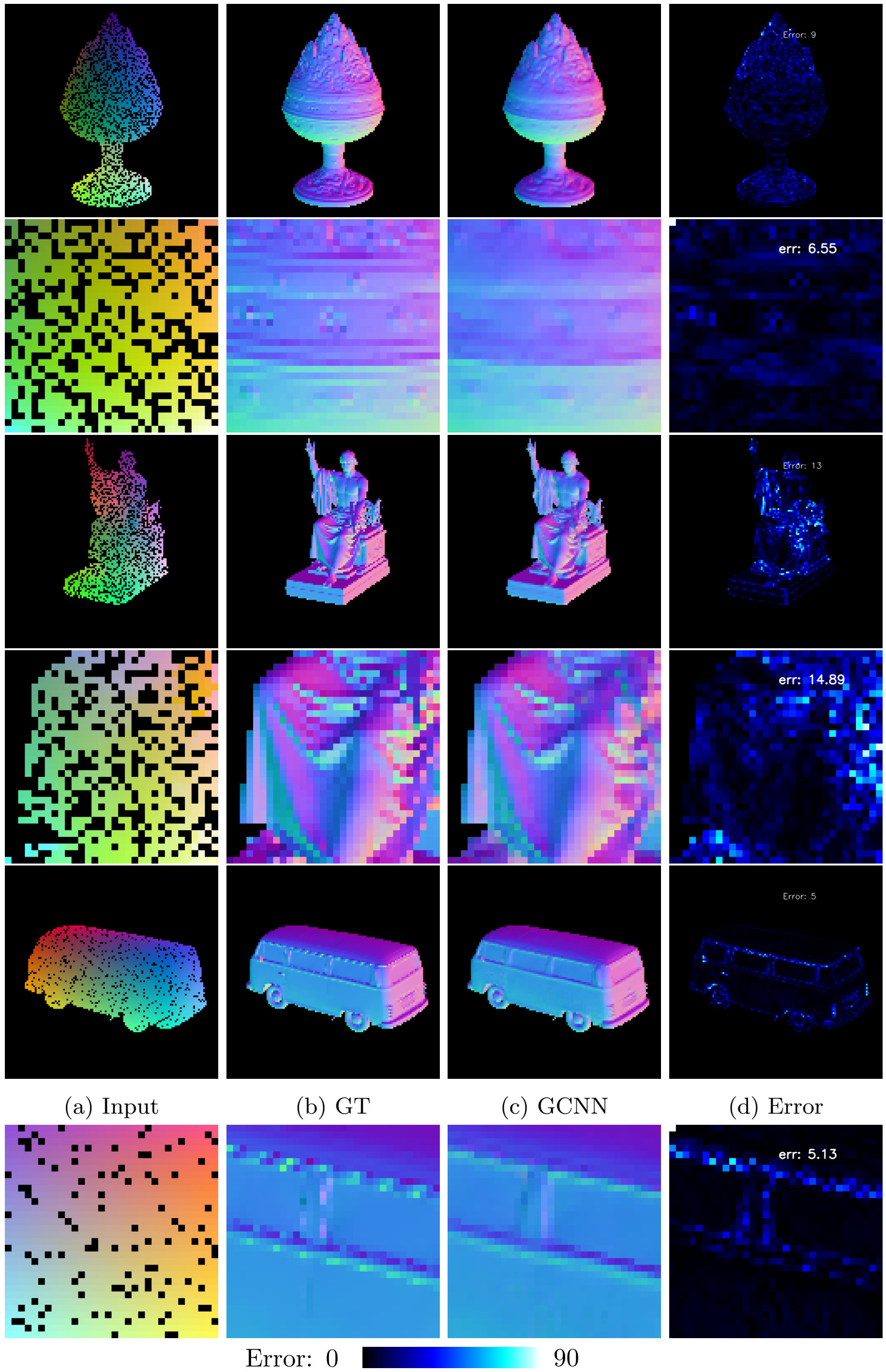


Figure 9: Evaluation on objects Baoshanlu, Washington statue, Bus(from top to bottom).

5 Trining on higher resolution

We also prepared the dataset with resolution 512×512 for model evaluation. Higher resolution gives more information about the surface feature using the same model compare to lower resolution. The benefit is, if we extract a fixed sized patch from the normal map, say 32×32, it might correspond a hindleg area of the dragon object in 128×128 resolution but maybe only a toe in 512×512 resolution. Therefore if we still use the same kernel size in the network for the same dataset, (like we did with 3×3) the higher the resolution it has, the less area it will cover. Thus the surface will more smooth and easy to calculate the surface normal. This is a good thing. Because then we might only need to use these 32×32 points to calculate a toe in 512×512 resolution image. But in 128×128 resolution image, the same area 32×32 might corresponding to entire hindleg of the dragon object. In another word, we can also say that the higher resolution "smooth" the object surface. Thus the higher resolution helps the model to calculate a more accurate normal map.

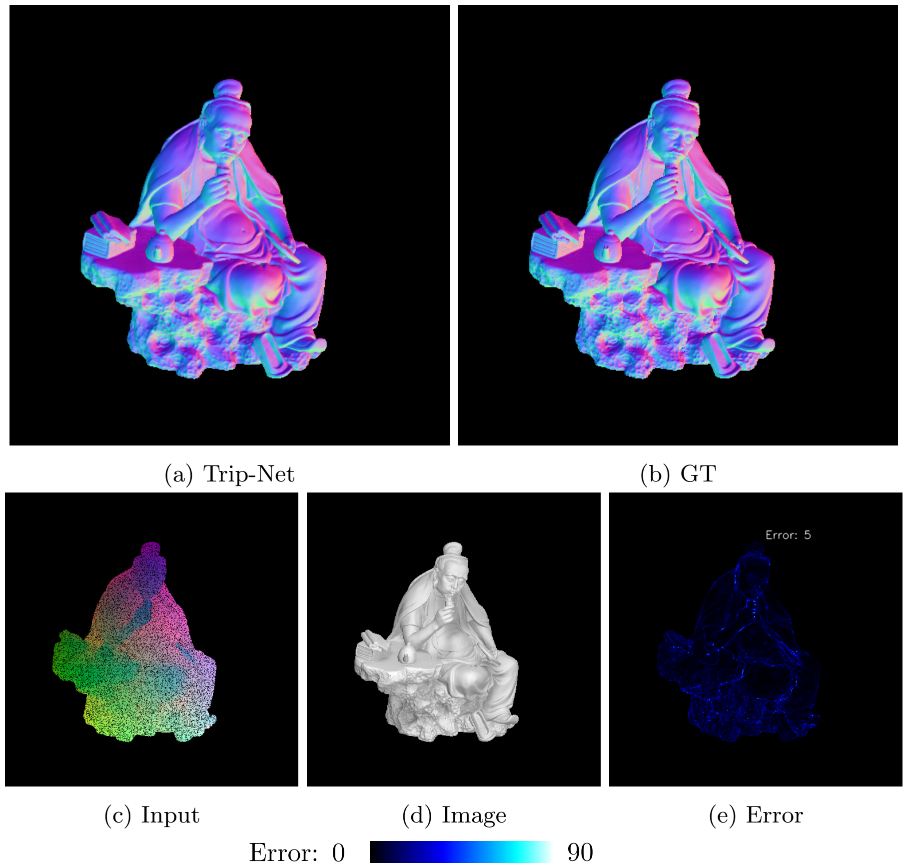
Since our model is fully convolution network, the architecture keeps exactly same on the high resolution dataset. We use the same settings and the same models for training work on dataset with 512×512 resolution. The training on high resolution network takes longer time but achieves a lower angle error.

Metrics	**SVD**	**GCNN**	**Trip-Net**
Mean	8.88	5.82	5.33
Median	3.66	3.98	3.67
5°	0.63	0.63	0.66
11.5°	0.79	0.89	0.91
22.5°	0.89	0.97	0.97
30°	0.92	0.98	0.99

Table 9: Evaluation of SVD, GCNN and Trip-Net models on 6 different metrics based on 100 test scenes.

A quantitative evaluation is shown in table 10. It follows the same performance rank compare to lower resolution, which is GCNN better than SVD approach and Trip-Net slightly better than GCNN. The accuracy is 99 % in the 30° metric for Trip-Net and 5° mean error. A qualitative evaluation is shown in figure 10, this is a figure object with both smooth area (the belly and arm) and highly detailed area with sharp surface (the uneven surface of stone table). Our models gives an average degree area at 5°. A comparison with other models is shown in figure 11.

Another remarkable result is the SVD approach, which also gives a good result (8.88° in mean degree metric). Like we discussed, the surface are relative more smooth if we keep the same window size for normal inference. In the high resolution scenes, although the percentage of missing pixels in a fixed windows are remain the same compare to small resolution scenes, but the remain valid pixels are mainly on a relatively flat plane thus they are good enough for an accurate normal inference. Consequently, we can see from the error map in Figure **??**, the sharp edges of the dragon, like the horns and the hindlegs, the SVD approach still has high errors.



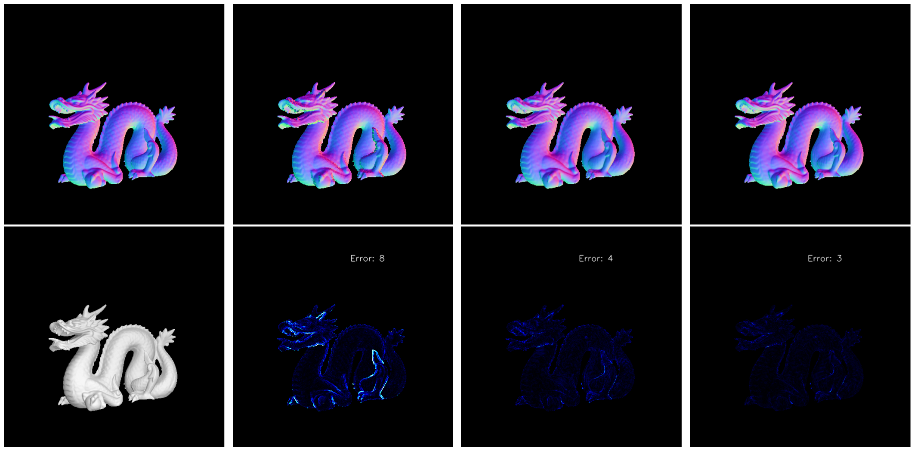
(a) Trip-Net (b) GT

(c) Input (d) Image (e) Error

Figure 10: Normal inference based on Trip-Net. Test image has resolution 128×128

6 Training on Real-Dataset

We also applied our model on a small dataset that captured by a light scanner in our laboratory in order to see the applicability of our approaches. For the geometry information based approach, we use the GCNN model that trained on high resolution synthetic dataset directly, since it only require the depth map as input, the scenarios of two dataset are the same. For the illuminated calibrated RGB-D image based approach, the model has to be refined based on the real-dataset, since the light intensity, camera matrix are different. We



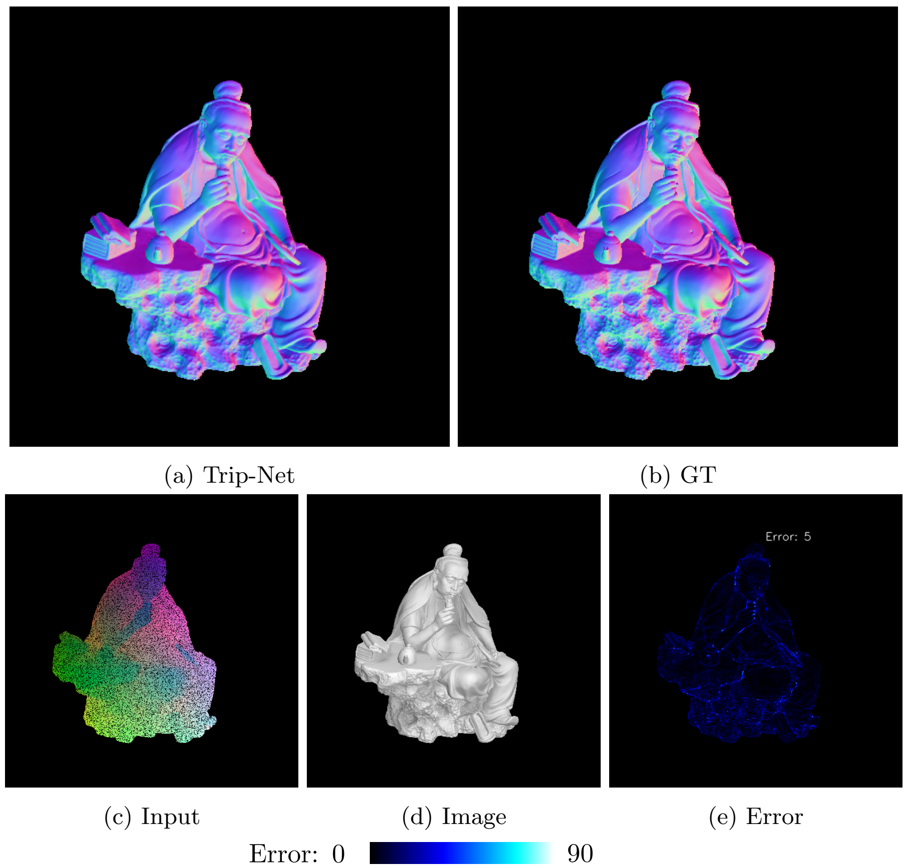
(a) GT (b) SVD (c) GCNN (d) Trip-Net

Figure 11: Comparison between different SVD, GCNN and Trip-Net model on 512×512 dataset. The first row is surface normal, the second row is the corresponding errors.

refined the Trip-Net based on a pre-trained GCNN model with the same settings in previous experiments, and observe the results. As shown in Figure 12

Metrics	**SVD**	**GCNN**	**Trip-Net**
Mean	8.20	8.74	**8.09**
Median	**4.87**	5.70	5.00
$5°$	**0.51**	0.44	0.50
$11.5°$	0.79	0.79	**0.81**
$22.5°$	0.93	0.93	**0.94**
$30°$	0.96	0.96	0.96

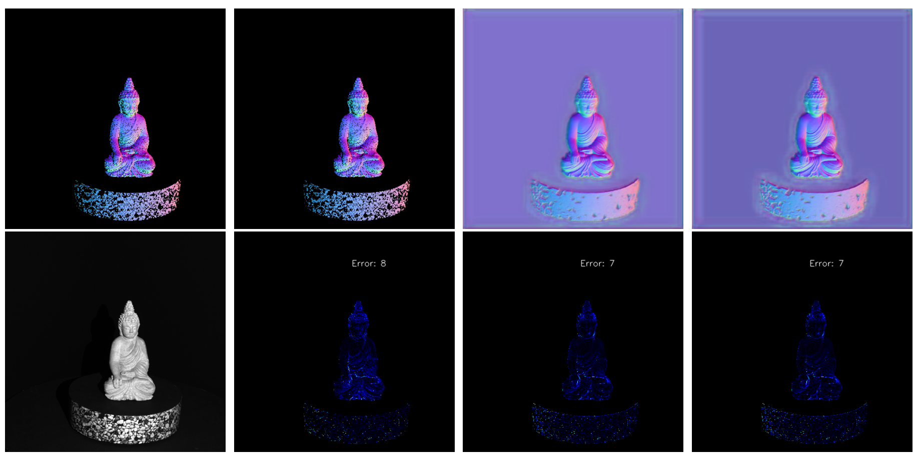
Table 10: Evaluation of SVD, GCNN and Trip-Net models on 6 different metrics based on 100 test scenes in real dataset.



(a) Trip-Net (b) GT

(c) Input (d) Image (e) Error

Figure 12: Normal inference based on Trip-Net. Test image has resolution 128×128



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

Figure 13: Comparison between different SVD, GCNN and Trip-Net model on 512×512 real dataset. The first row is surface normal, the second row is the corresponding errors.