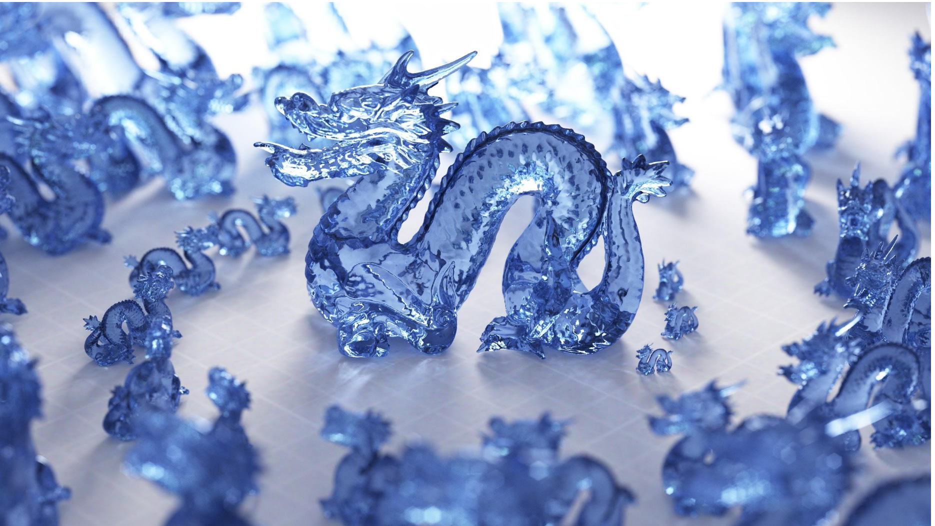
NR-IQA using Fully Convolutional NNs

Joss Whittle






The Problem

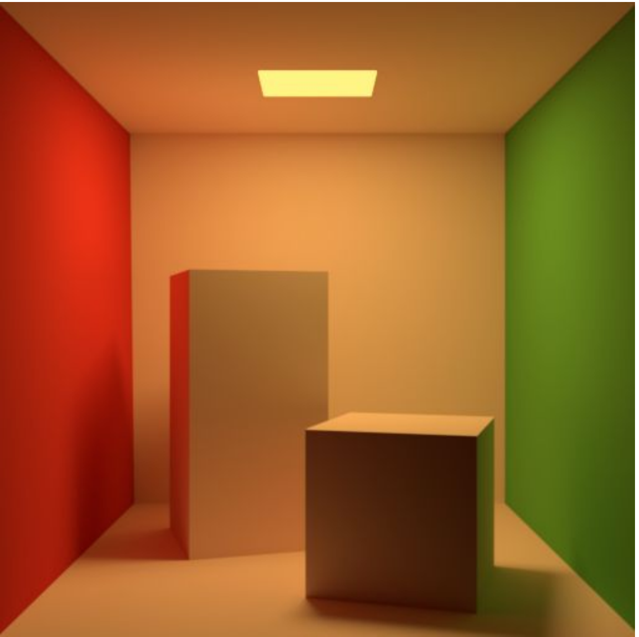
Monte Carlo Rendering produces noise in images at low sample counts.

To what degree are these distortions affecting image quality?

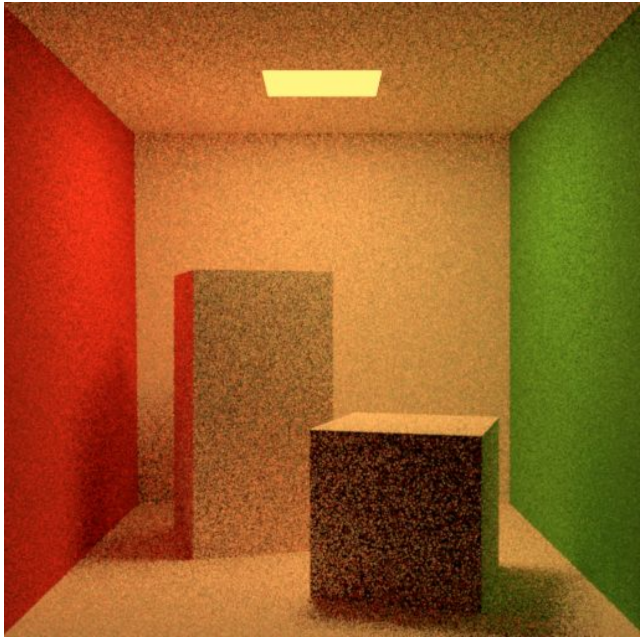
Largely subjective and hard to model well across all conditions.

Full-Reference IQA - Compare to a known Truth

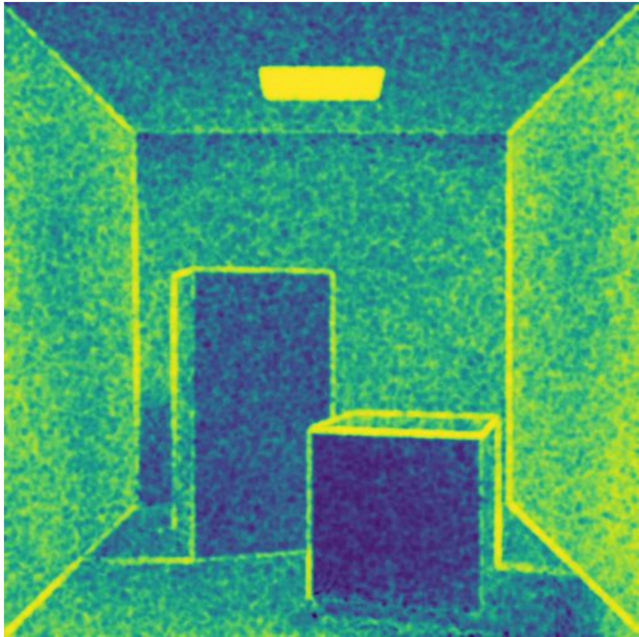
Reference



Test



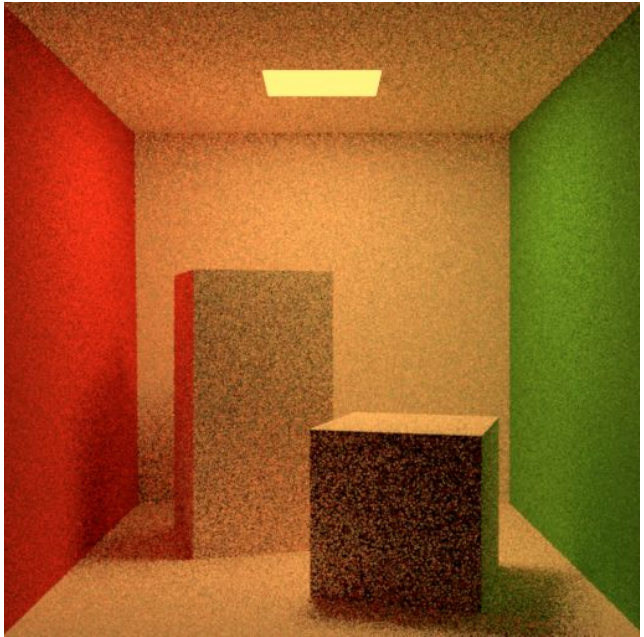
SSIM(Ref, Test)



During Rendering No Ground Truth Available

Can we still evaluate image quality?

?



We do this by comparing to a learned distribution of images we consider to be plausible.

?

Existing No-Reference IQA Methods

Trained on Public Datasets such as TID2008/2013, Live, ect.

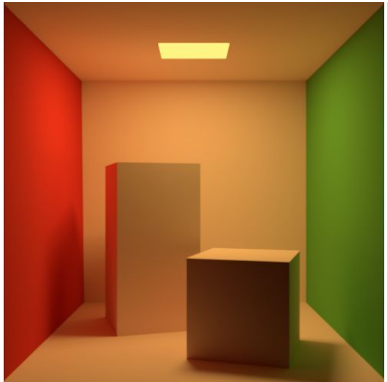
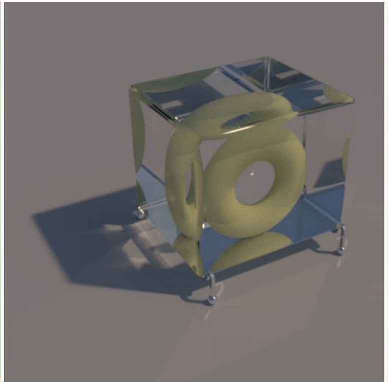
Images are natural photographs corrupted synthetically using distortions such as Gaussian Blur, random noise, masking, fast-fading, quantization, ect.

Are these distortions representative of the natural distortions we will encounter during inference?

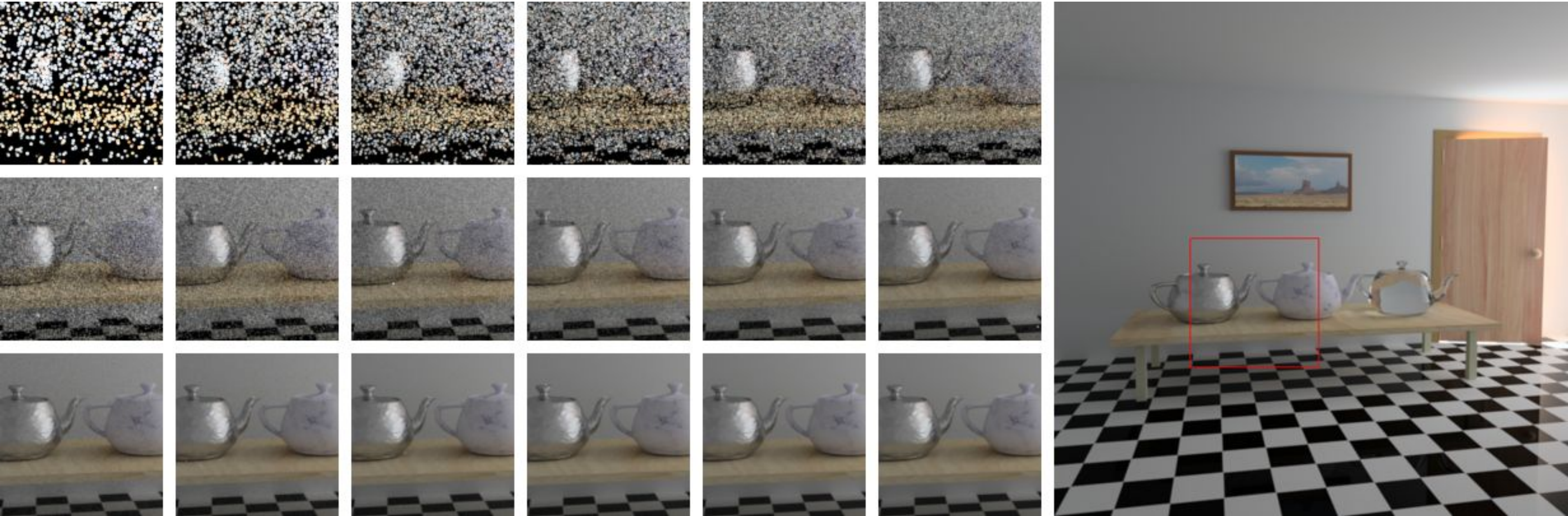
When evaluating Monte Carlo Rendered Images, no.

Compute a new Dataset of Monte Carlo Rendered Images containing naturally occurring distortions.

Several scenes, lighting conditions, materials.


Render each scene independently to an increasing quality generating a large amount of data about how different image regions may look under varying distortions.

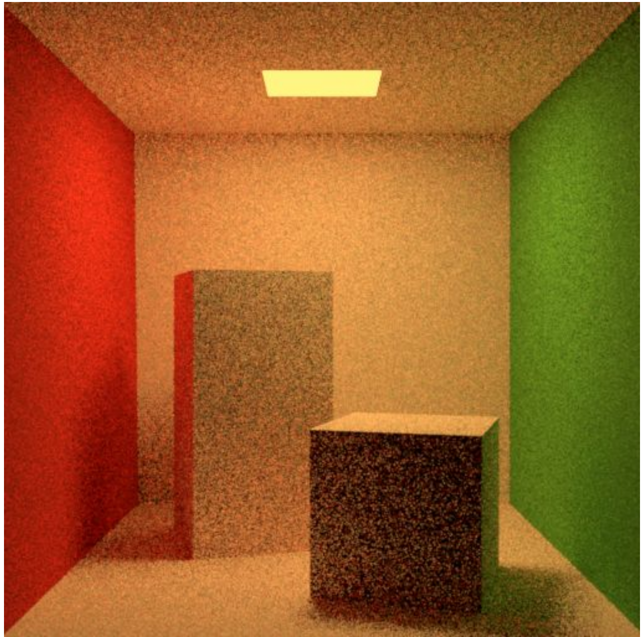


Goal: Predict per-pixel quality score given only noisy image

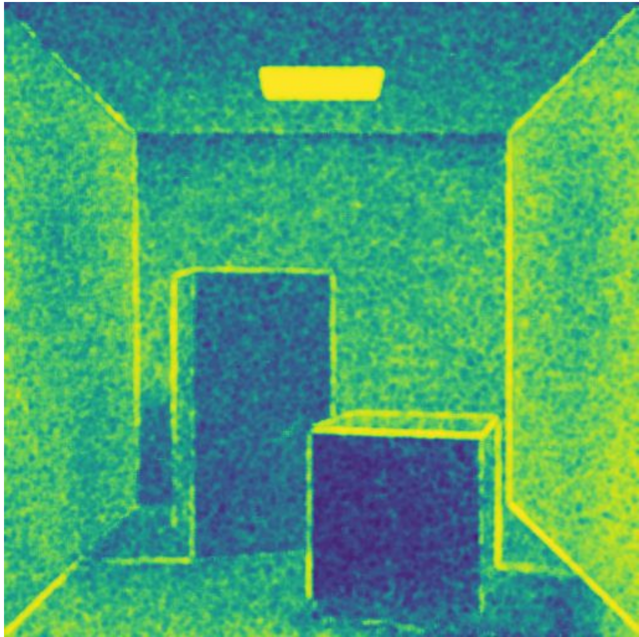
Use dataset of naturally distorted images to learn distribution of distortions.

Apply learned distribution during inference to new images we have not trained on.

Test



Predicted_SSIM(Test)

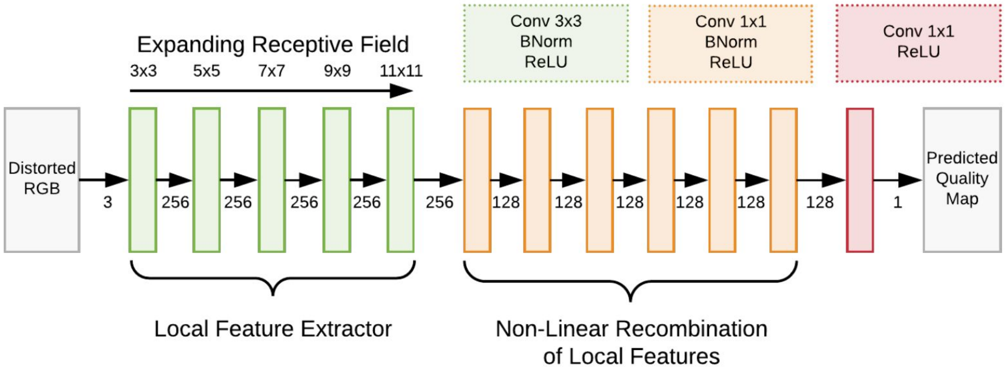


The model

Fully Convolutional Neural Network: Output Size (H,W,1) is the same as the Input Size (H,W,3) giving a per-pixel quality prediction.

First few blocks use 3x3 convs to bring the receptive field of the network up to 11x11 (same as target metric - SSIM)

Remaining blocks use 1x1 convs to perform per-pixel non-linear combination of local features.



Training

Train on randomly selected 64x64 patches each batch.

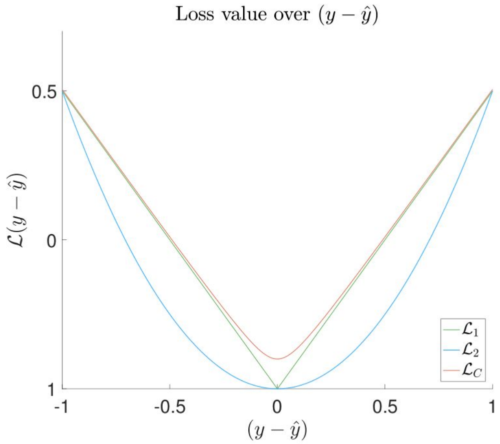
Use per-pixel batchwise Loss between

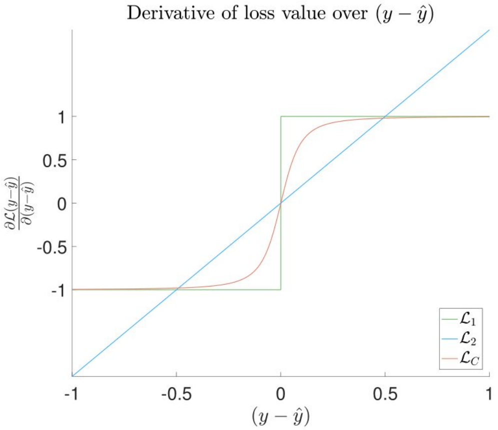
y_pred = Predicted_SSIM(Test)
y = SSIM(Ref, Test)

Loss function based on Charbonnier Loss with added Pearson's Correlation Coefficient term to regularize learning.

$$\mathcal{L}_{\mathcal{C}} = \sqrt{(y-\hat{y})^2 + \epsilon^2}$$

$$\mathcal{L}_{\text{Joint}} = \delta(1 - |PCC(y, \hat{y})|) + \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}_{\mathcal{C}}(y_i, \hat{y}_i)$$





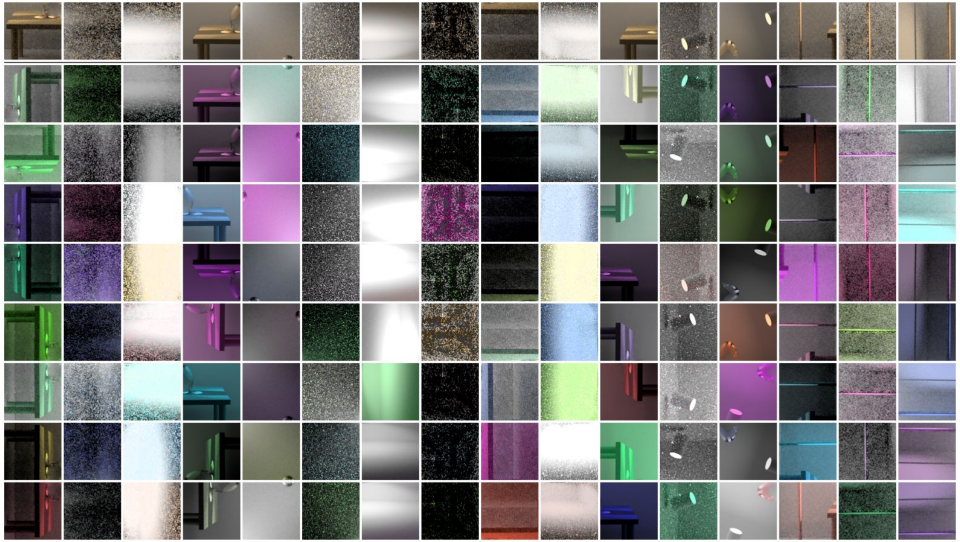
Data Augmentation

Model is susceptible to memorizing colour pallette information as a cue for predicting quality.

We want to discourage this behaviour.

Augment the training patches with changes in rotation, flip, and HSV colour space. No augmentation is applied at inference time.

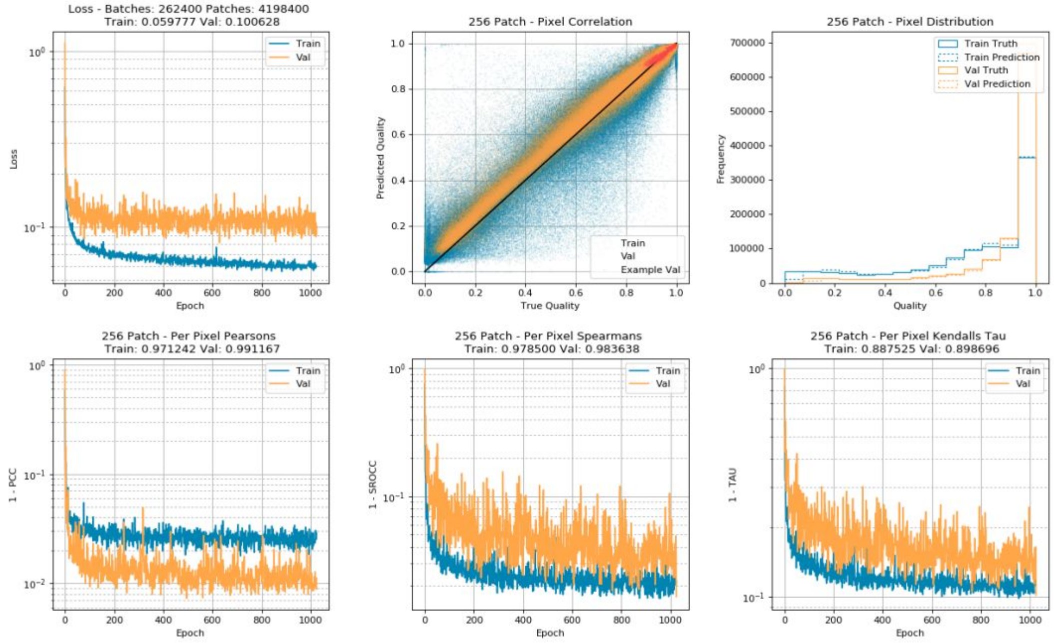
$$
\begin{aligned}
H, S, V &= \mathrm{RGBtoHSV}(R, G, B) \\
H' &= (H + \xi_H) \ mod \ 1 \\
S' &= \mathrm{clip}(S + \xi_S, 0, 1) \\
V' &= \mathrm{clip}(V + \xi_V, 0, 1) \\
R', G', B' &= \mathrm{HSVtoRGB}(H', S', V') \\
\text{where} \quad \xi_H \in \mathcal{U}(0,1) &\quad \text{and} \quad \xi_S, \xi_V \in \mathcal{U}(-0.3, 0.3)
\end{aligned}
$$



Training Progression

Validation (orange) Loss and PCC begin to plateau while Validation SROCC and Tau continue to improve.

Good correlation between per-pixel predictions and ground truth quality scores.

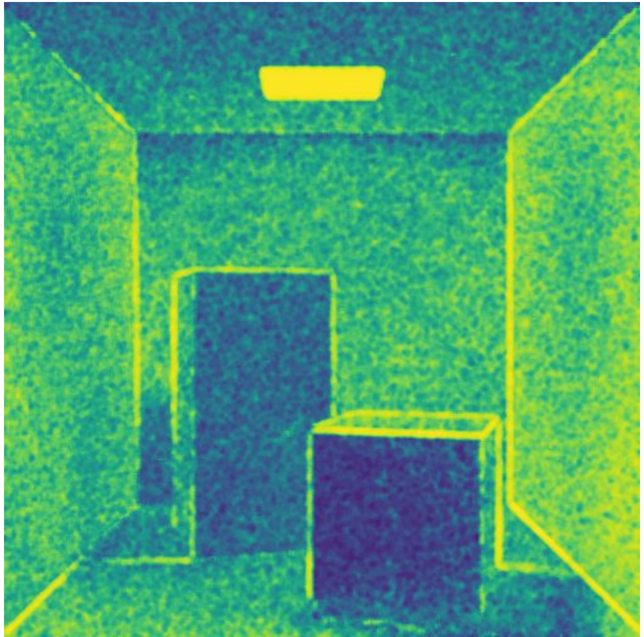


(a) Scene: Cornell Box

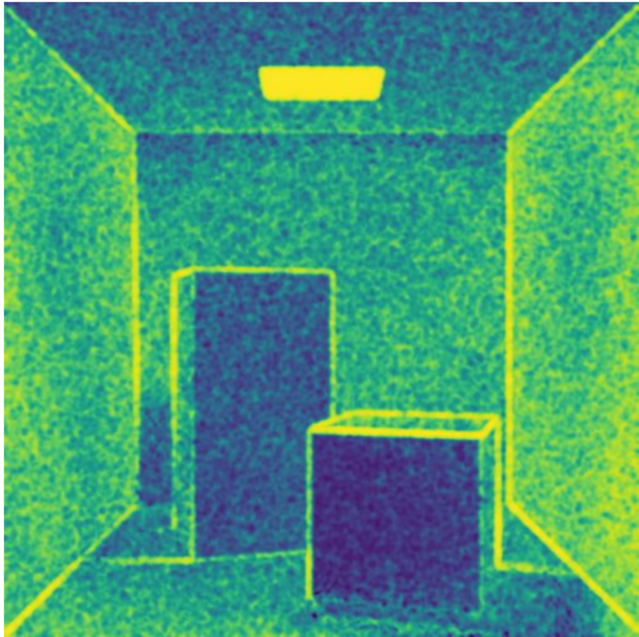
Results

Test

Predicted_SSIM(Test)



SSIM(Ref, Test)



Results

Our model achieves state of the art results on NR-IQA for Monte Carlo Rendered Images containing natural distortions.

Other NR-IQA designed for synthetic distortions and are not able to perform inference correctly on natural Monte Carlo distortions.

	Cornell Box			Veach Bidir			Veach Door			Sponza		
Metric	PCC	SROCC	Tau	PCC	SROCC	Tau	PCC	SROCC	Tau	PCC	SROCC	Tau
Ours - Experiment II	**0.9996**	**0.9959**	**0.9688**	**0.9982**	**0.9921**	**0.9393**	**0.9928**	**0.9916**	**0.9331**	**0.9989**	**0.9964**	**0.9686**
BLIINDS [Saa+10]	0.9840	0.9858	0.9287	0.9619	0.9544	0.8650	0.9640	0.9870	0.9144	0.9066	0.9668	0.8723
BIQI [MB10]	-0.2150	-0.0472	-0.0145	-0.0365	-0.3395	-0.2411	-0.7170	-0.3449	-0.2462	-0.4219	-0.5132	-0.3693
BRISQUE [Mit+12a]	0.3596	0.6854	0.5414	0.2211	0.3402	0.2172	0.1214	0.4154	0.3546	-0.5622	-0.3427	-0.2879
HIGRADE 1 [Kun+16a]	0.2321	-0.1136	-0.0779	-0.7522	-0.6113	-0.4465	0.3415	-0.3348	-0.2901	0.2163	-0.0028	0.0246
HIGRADE 2 [Kun+16a]	0.4008	0.4236	0.3167	-0.7526	-0.8591	-0.6761	0.4064	0.1528	0.1009	0.4042	0.2663	0.1867
JP2K-NR [She+05b]	-0.3977	-0.9352	-0.8264	-0.5410	-0.9139	-0.7517	-0.0234	0.2408	0.1091	-0.6635	-0.8156	-0.6524
NIQE [Mit+13b]	0.6013	0.9490	0.8283	0.6573	0.8651	0.7089	-0.0037	0.4418	0.3581	-0.4974	-0.2841	-0.2371
OG-IQA [Liu+16]	0.2863	0.4868	0.3278	-0.2013	-0.1947	-0.1759	-0.5076	-0.2150	-0.1673	-0.7798	-0.6177	-0.4556

Tab. 5.8.: Validation set NR-IQA performance for **leave one scene out** cross validation. Each correlation is computed relative to the full reference SSIM of each 512×512 image in the validation set compared to its ground truth image.