Transfer CLIP for Generalizable Image Denoising

Supplementary Material

Algorithm 1 Extract dense features from CLIP ResNet.

```
# PyTorch code of CLIP ResNet image encoder
# the forward function
def forward(self, x):
    out = [] # store multi-scale dense features
    x = x.type(self.conv1.weight.dtype)
    x = self.relu1(self.bn1(self.conv1(x)))
    x = self.relu2(self.bn2(self.conv2(x)))
    x = self.relu3(self.bn3(self.conv3(x)))
    out.append(x) # scale-1 F1
    x = self.avgpool(x)
    x = self.layer1(x); out.append(x) # scale-2 F2
    x = self.layer2(x); out.append(x) # scale-3 F3
    x = self.layer3(x); out.append(x) # scale-4 F4
    x = self.layer4(x); out.append(x) # scale-5 F5
    x = self.attnpool(x)
    return out
```

7. More Analyses of CLIP ResNet Encoder

We conduct more feature analysis of CLIP frozen ResNet encoder for the image *Lena* using Poisson noise and CKA similarity measure, respectively, and report the results in Fig. 9 and 10. Besides, feature similarity analysis of CLIP ResNet encoder for the image *flowers* from Set14 is also performed and shown in Fig. 11.

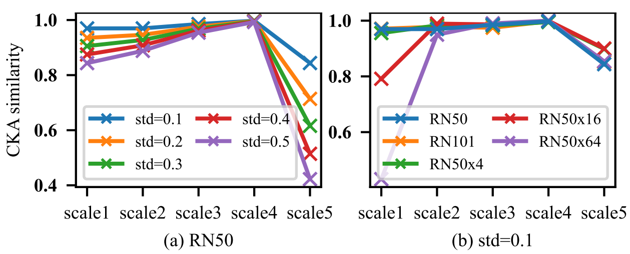

Figure 9. Feature similarity analysis of the CLIP ResNet encoder under *i.i.d.* Gaussian noise with varying levels and CKA similarity measure

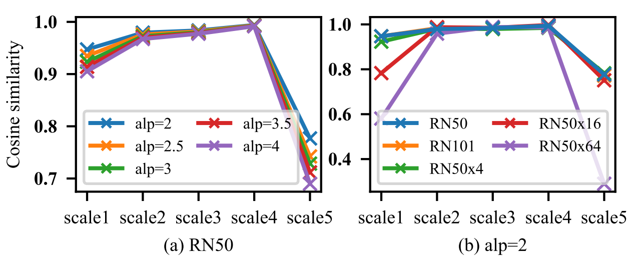

Figure 10. Feature similarity analysis of the CLIP ResNet image encoder under Poisson noise with varying levels and cosine similarity measure

8. More Implementation Details

We present the details of our learnable decoder in Fig. 12.

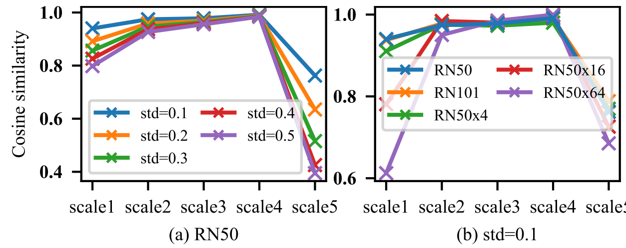

Figure 11. Feature similarity analysis for the image *flowers* from set14 under Gaussian noise and cosine similarity measure

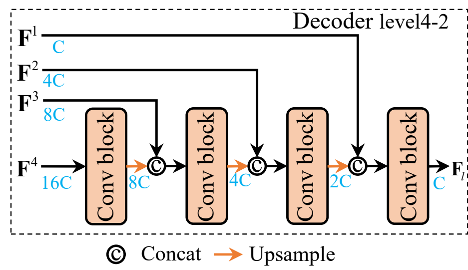

Figure 12. The detail of the learnable decoder (level-4 to -2) in our CLIPDenoising. The symbols with lightblue represent the channel number, where C is the base channel number and is 64 in RN50

9. More experimental results

9.1. Model size and inference time

We provide the model size, inference time, and FLOPs of compared methods in Table 6. Note that the frozen RN50 (excluding the last layer) in our model has 8.5M parameters, and the learnable decoder has 11M parameters. And HAT uses the DnCNN model.

Table 6. Efficiency comparisons (test on size $3 \times 512 \times 512$)

	DnCNN	Restormer	MaskDenoi.	HAT	DIL	ours
Params(M)	**0.67**	26.1	0.88	**0.67**	16.6	19.5
Infer.time(s)	0.034	0.415	2.239	0.034	0.867	**0.018**
FLOPs(G)	176.1	564.0	204.9	176.1	4360	**83.58**

9.2. Results on environmental noise

We consider image deraining task. We use Rain100L as the training set and use *Rain12*, *Rain drop (S)*, and *Rain and mist (S)* as OOD test sets. Results in Table 7 imply that our model exhibits better generalization capability than PromptIR (NeurIPS 2023), a recent image restoration model.

9.3. Results of using CLIP ViT image encoder

We incorporate CLIP ViT-B/16 image encoder into our model and report the resultant results in Table 8. Specifically, the model ex-

Table 7. Results on image deraining tasks (PSNR/SSIM)

	Rain12	*Rain drop (S)*	*Rain and mist (S)*
PromptIR	35.00/0.944	22.98/0.835	21.89/0.673
ours	**35.11/0.953**	**23.66/0.838**	**28.56/0.869**

tracts features from the middle (i.e., 7-th) layer of the frozen ViT-B/16, and then feeds it to 4 learnable ViT blocks and 4 learnable upsampling blocks that upsample the deep features back to the original image space. By comparing Table 8 and Tables 1, 9, our model with image ViT encoder shows inferior in-distribution and OOD performance.

Table 8. Results of using CLIP ViT-B/16 image encoder

	Gauss(15)	Gauss(50)	Poisson(3.5)	Speckle(0.04)
McM	33.28/0.903	21.29/0.372	24.78/0.618	27.41/0.755
Kodak24	33.98/0.914	20.59/0.338	24.02/0.529	28.08/0.741

10. Experiments on MoCo-v3 ResNet50

We conduct the feature similarity analysis of frozen MoCo-v3 ResNet50 for the image *Lena* using $i.i.d.$ Gaussian noise and cosine similarity measure, and report the result in Fig. 13. Five multi-scale features show robustness to noise. Subsequently, we substitute the RN50 of CLIP with the RN50 of MoCo-v3 in our denoiser and perform the model training and OOD experiments. As observed in Table 10, frozen MoCo-v3 RN50-powered deep denoiser exhibits a certain level of generalization ability compared with DnCNN and our method.

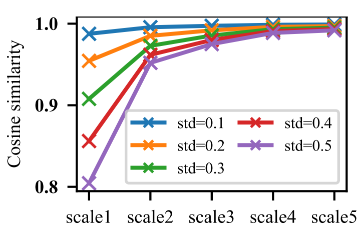

Figure 13. Feature similarity analysis of the MoCo-v3 ResNet50 under Gaussian noise. Cosine similarity is utilized here

Table 9. Additional quantitative comparison of different methods on CBSD68, McMaster, Kodak24 and Urban100 datasets with regard to diverse synthetic OOD noises. The best results are highlighted in **bold** and the second best is underlined. Note that *multiple noise levels* are required by HAT and DIL during the training to achieve generalization, while our method only needs *one noise level* for training

Noise Types	Datasets	DnCNN [62]	Restormer [60]	MaskDenoising [6]	HAT [58]	DIL [32]	Ours
Gauss $\sigma = 15$	CBSD68	33.78/0.931	**34.42/0.936**	30.99/0.888	33.22/0.912	32.50/0.906	33.97/0.930
	McMaster	34.03/0.914	**35.61/0.935**	30.85/0.832	33.04/0.883	32.45/0.853	33.86/0.910
	Kodak24	34.59/0.924	**35.49/0.931**	31.79/0.884	33.88/0.905	33.37/0.912	34.69/0.922
	Urban100	32.10/0.934	**34.57/0.955**	29.50/0.899	32.48/0.916	32.15/0.920	33.15/0.930
Gauss $\sigma = 25$	CBSD68	29.89/0.828	27.21/0.681	28.44/0.815	30.54/0.853	29.98/0.843	**31.02/0.878**
	McMaster	30.46/0.806	27.31/0.633	28.76/0.778	30.41/0.807	30.17/0.790	**31.50/0.866**
	Kodak24	30.53/0.808	27.64/0.639	29.08/0.792	31.37/0.849	30.99/0.856	**31.86/0.871**
	Urban100	28.88/0.847	27.35/0.744	27.63/0.836	29.97/0.867	29.72/0.876	**30.72/0.893**
Spatial Gauss $\sigma = 45$	CBSD68	28.05/0.785	24.51/0.668	28.19/0.815	28.33/0.791	26.33/0.703	**29.43/0.849**
	McMaster	28.24/0.740	24.01/0.539	28.27/0.762	28.44/0.742	26.47/0.646	**29.82/0.825**
	Kodak24	28.23/0.752	23.66/0.609	28.85/0.805	28.44/0.760	26.31/0.652	**30.12/0.838**
	Urban100	27.53/0.806	25.80/0.722	27.34/0.837	28.32/0.809	26.51/0.737	**29.42/0.868**
Spatial Gauss $\sigma = 50$	CBSD68	26.92/0.741	23.98/0.630	27.47/0.790	27.32/0.752	25.43/0.665	**28.51/0.825**
	McMaster	27.16/0.693	24.63/0.573	27.60/0.738	27.49/0.703	25.61/0.609	**29.10/0.803**
	Kodak24	27.04/0.702	23.29/0.569	28.09/0.778	27.37/0.715	25.39/0.611	**29.21/0.815**
	Urban100	26.49/0.766	24.93/0.690	26.74/0.815	27.36/0.776	25.61/0.704	**28.55/0.847**
Poisson $\alpha = 2.5$	CBSD68	28.70/0.806	25.63/0.693	27.80/0.806	**30.15**/0.858	29.53/0.870	29.94/**0.874**
	McMaster	29.80/0.799	25.75/0.693	28.55/0.730	30.87/0.840	30.74/**0.871**	**30.93**/0.864
	Kodak24	29.23/0.776	26.06/0.644	28.42/0.781	30.90/0.843	30.42/0.868	**30.93/0.869**
	Urban100	27.56/0.806	25.14/0.719	26.95/0.815	**29.52**/0.871	29.17/**0.893**	29.51/0.884
Poisson $\alpha = 3$	CBSD68	26.37/0.712	23.55/0.615	25.87/0.718	28.48/0.804	28.52/0.844	**28.71/0.846**
	McMaster	27.49/0.720	23.62/0.632	26.13/0.667	29.30/0.784	29.78/**0.848**	**29.82**/0.843
	Kodak24	26.68/0.660	23.95/0.561	27.04/0.683	29.16/0.779	29.45/**0.844**	**29.71**/0.841
	Urban100	25.41/0.721	22.72/0.642	25.33/0.737	27.82/0.815	28.09/**0.874**	**28.27**/0.862
Speckle $\sigma^2 = 0.02$	CBSD68	31.79/0.898	29.10/0.826	29.91/0.875	**32.50**/0.916	31.57/**0.924**	31.82/0.904
	McMaster	32.74/0.886	28.89/0.800	30.47/0.809	**33.11**/0.899	32.66/**0.907**	32.28/0.870
	Kodak24	32.82/0.895	29.96/0.814	30.80/0.874	**33.26**/0.908	32.35/**0.919**	32.91/0.908
	Urban100	30.11/0.893	28.24/0.828	28.60/0.883	**31.49**/0.917	30.90/**0.930**	30.94/0.904
Speckle $\sigma^2 = 0.03$	CBSD68	30.10/0.856	26.78/0.765	28.99/0.851	**31.10**/0.893	30.40/**0.906**	30.48/0.886
	McMaster	31.21/0.846	26.81/0.752	29.70/0.778	**31.95**/0.873	31.70/**0.904**	31.31/0.858
	Kodak24	31.12/0.852	27.50/0.740	29.90/0.848	**31.95**/0.884	31.28/**0.901**	31.64/0.891
	Urban100	28.37/0.841	25.86/0.774	27.65/0.847	**30.10**/0.892	29.72/**0.916**	29.69/0.889
Speckle $\sigma^2 = 0.04$	CBSD68	28.65/0.812	25.13/0.719	27.94/0.815	**29.97**/0.867	29.56/**0.890**	29.49/0.870
	McMaster	29.69/0.804	25.30/0.717	28.68/0.736	30.90/0.845	**30.94/0.893**	30.47/0.845
	Kodak24	29.53/0.801	25.66/0.683	28.76/0.804	**30.82**/0.856	30.49/**0.887**	30.67/0.876
	Urban100	26.92/0.795	24.17/0732	26.64/0.804	**28.86**/0.862	28.82/**0.903**	28.69/0.875
Salt&Pepper $\alpha = 0.012$	CBSD68	28.56/0.814	25.88/0.779	30.49/0.863	29.31/0.846	30.81/0.865	**31.95/0.890**
	McMaster	27.76/0.773	25.32/0.746	30.11/0.798	28.39/0.804	30.44/0.820	**31.90/0.863**
	Kodak24	29.17/0.797	26.17/0.751	31.27/0.861	29.91/0.834	31.24/0.851	**32.72/0.882**
	Urban100	27.40/0.823	25.73/0.815	29.08/0.880	28.60/0.851	30.49/0.875	**31.50/0.901**
Salt&Pepper $\alpha = 0.016$	CBSD68	27.45/0.780	24.57/0.726	30.13/0.853	28.32/0.813	30.02/0.841	**30.85/0.857**
	McMaster	26.61/0.730	24.00/0.687	29.70/0.786	27.37/0.763	29.75/0.793	**30.85/0.838**
	Kodak24	28.05/0.760	24.81/0.690	30.94/0.853	28.95/0.797	30.52/0.827	**31.67/0.863**
	Urban100	26.42/0.806	24.45/0.771	28.76/0.871	27.63/0.820	29.75/0.855	**30.30/0.889**

Table 10. Quantitative comparison of DnCNN, our model with frozen CLIP RN50, and our model with frozen MoCo-v3 RN50 on McMaster and Kodak24 datasets with regard to various synthetic OOD noises. All methods are trained under $i.i.d.$ Gaussian noise with $\sigma = 15$. Progressive feature augmentation is not used here.

McMaster	Gauss $\sigma = 50$	Spatial Gauss $\sigma = 55$	Poisson $\alpha = 3.5$	Speckle $\sigma^2 = 0.04$	S&P $d = 0.02$
DnCNN	20.18/0.312	26.18/0.649	25.50/0.651	29.69/0.804	25.72/0.691
Ours+CLIP RN50	**26.95/0.698**	**28.24/0.771**	**28.82/0.814**	**30.29/0.824**	29.62/0.795
Ours+MoCo-v3 RN50	24.85/0.625	27.36/0.748	26.92/0.747	29.68/0.823	**29.96/0.809**
Kodak24	**Gauss $\sigma = 50$**	**Spatial Gauss $\sigma = 55$**	**Poisson $\alpha = 3.5$**	**Speckle $\sigma^2 = 0.04$**	**S&P $d = 0.02$**
DnCNN	19.78/0.301	25.98/0.653	24.49/0.560	29.53/0.801	27.10/0.723
Ours+CLIP RN50	**26.87/0.692**	**28.19/0.781**	**29.74/0.840**	**30.60/0.871**	30.52/0.832
Ours+MoCo-v3 RN50	25.59/0.630	27.72/0.747	26.82/0.718	29.90/0.829	**30.95/0.834**

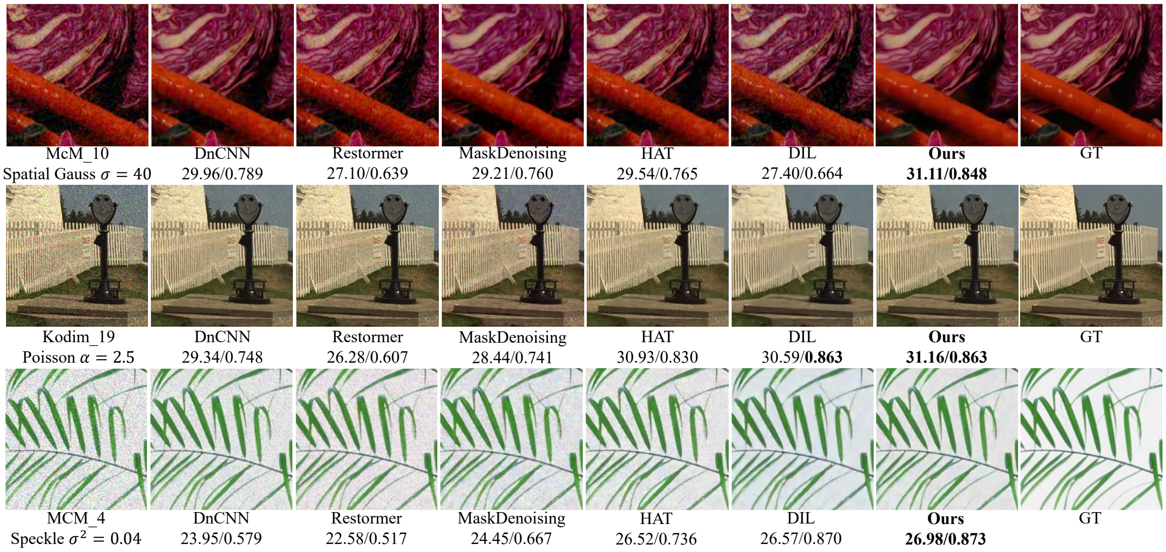

Figure 14. More qualitative denoising results on synthetic OOD noise.

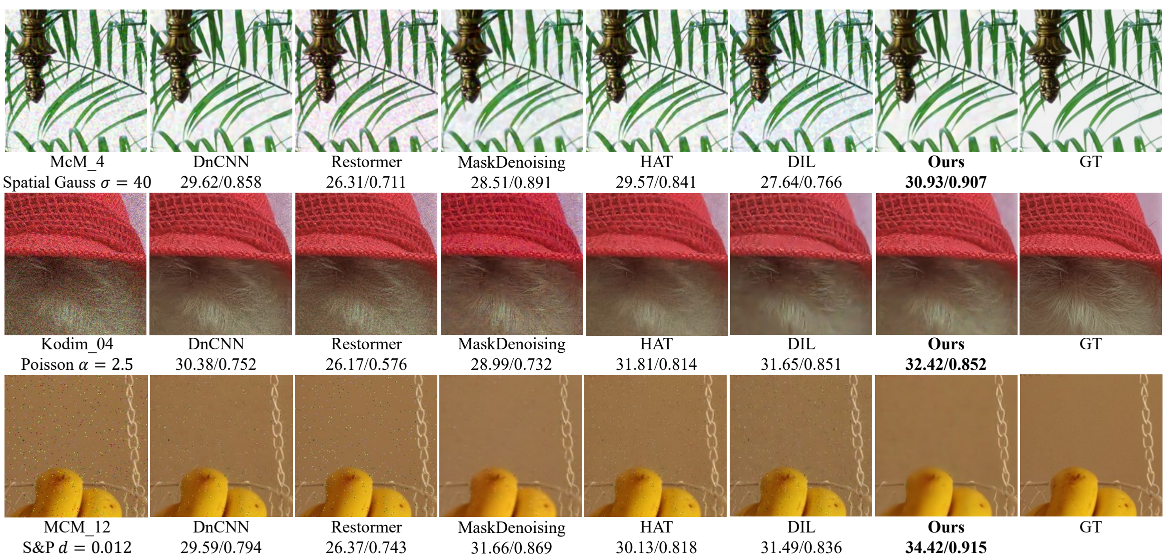

Figure 15. More qualitative denoising results on synthetic OOD noise.

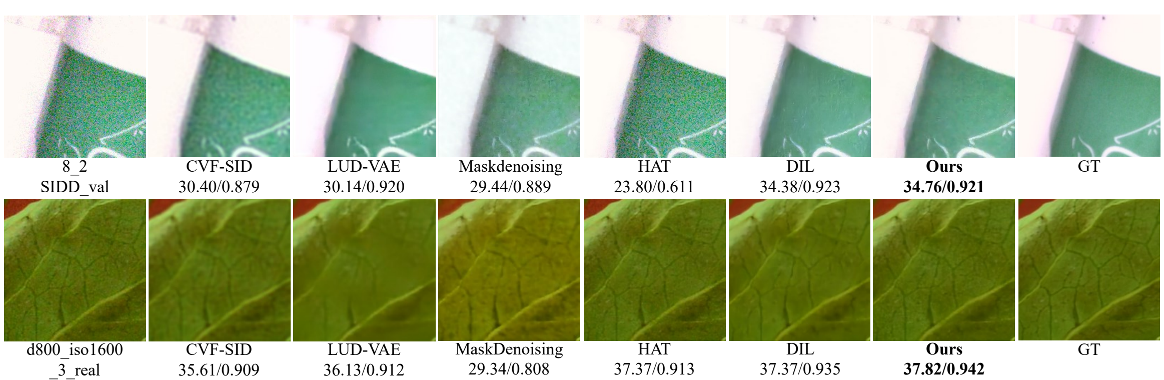

Figure 16. More qualitative denoising results on real-world sRGB noise