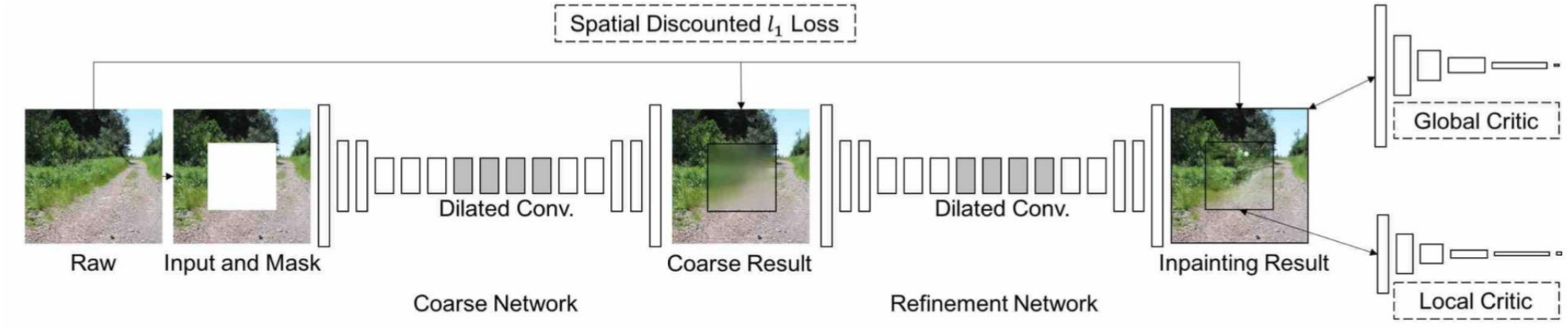
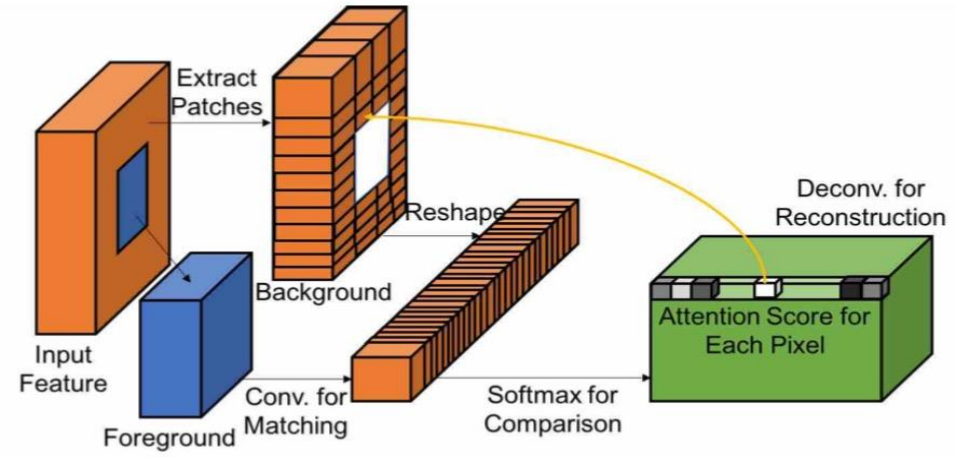
PEPSI : Fast Image Inpainting with Parallel Decoding Network (CVPR 2019)

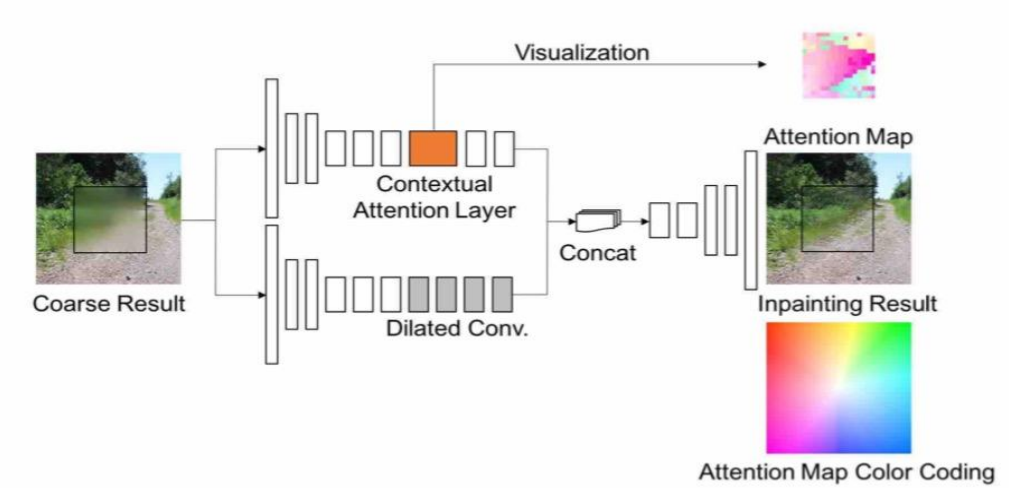
PEPSI++: Fast and Lightweight Network for Image Inpainting(arxiv 1905.09010)

https://arxiv.org/abs/1904.09925
https://arxiv.org/abs/1905.09010

Generative Image Inpainting with Contextual Attention (CVPR 2018)







PEPSI : Fast Image Inpainting with Parallel Decoding Network (CVPR 2019)

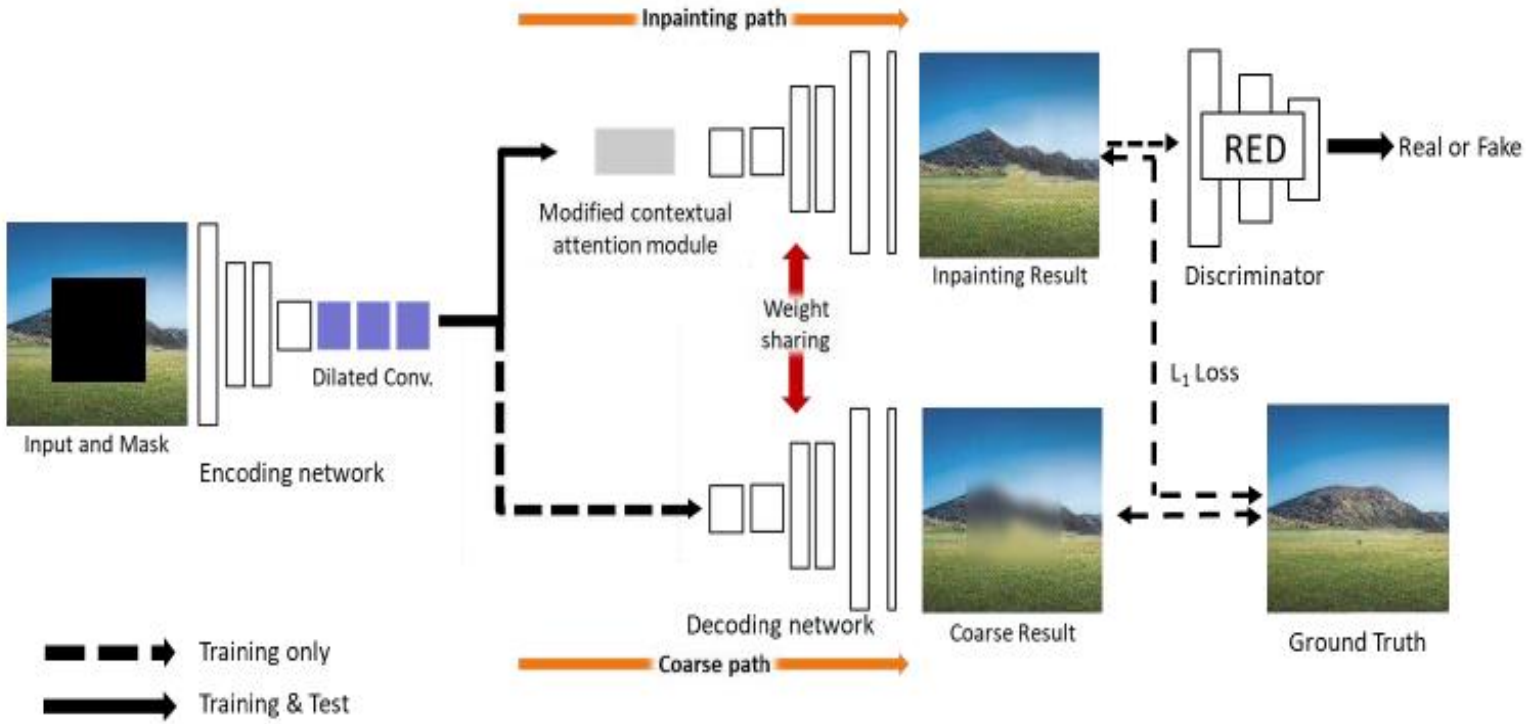
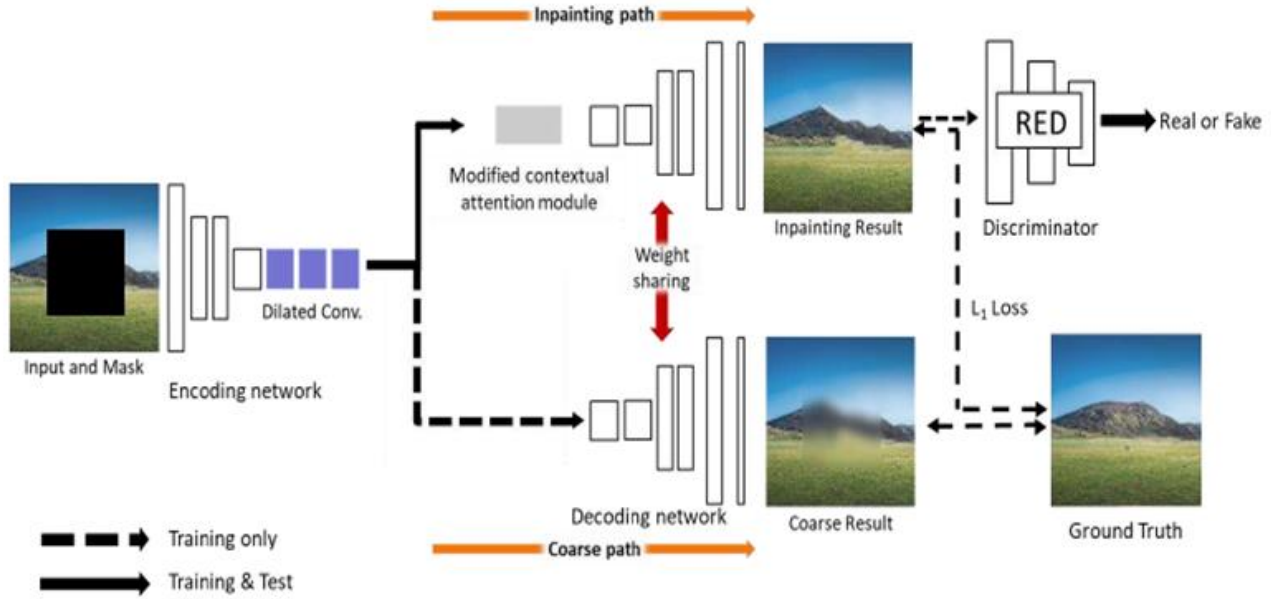
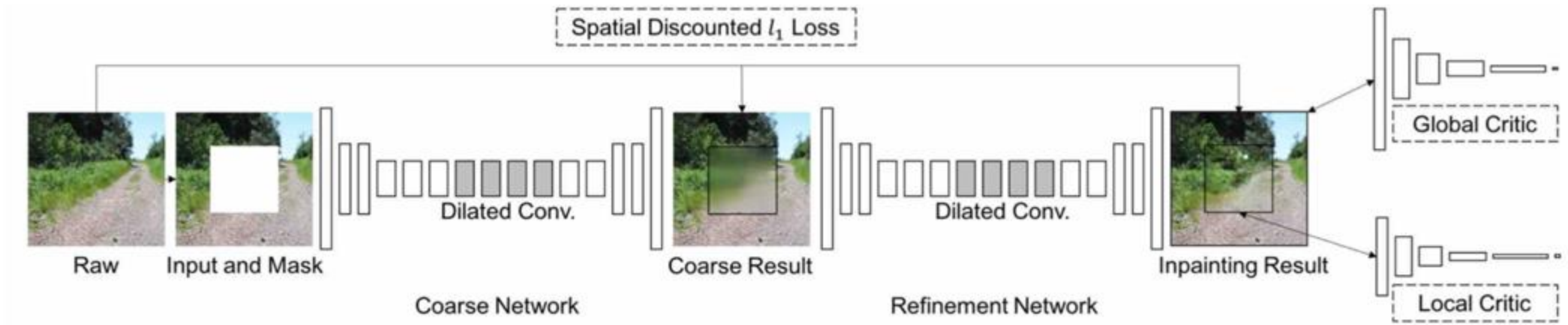


Figure 1. An architecture of PEPSI. The coarse path and inpainting path share their weights to improve each other. The coarse path is trained only with the ℓ_1 reconstruction loss while the inpaiting path is trained with both of ℓ_1 and adversarial loss

PEPSI (CVPR 2019)



Contextual Attention (CVPR 2018)

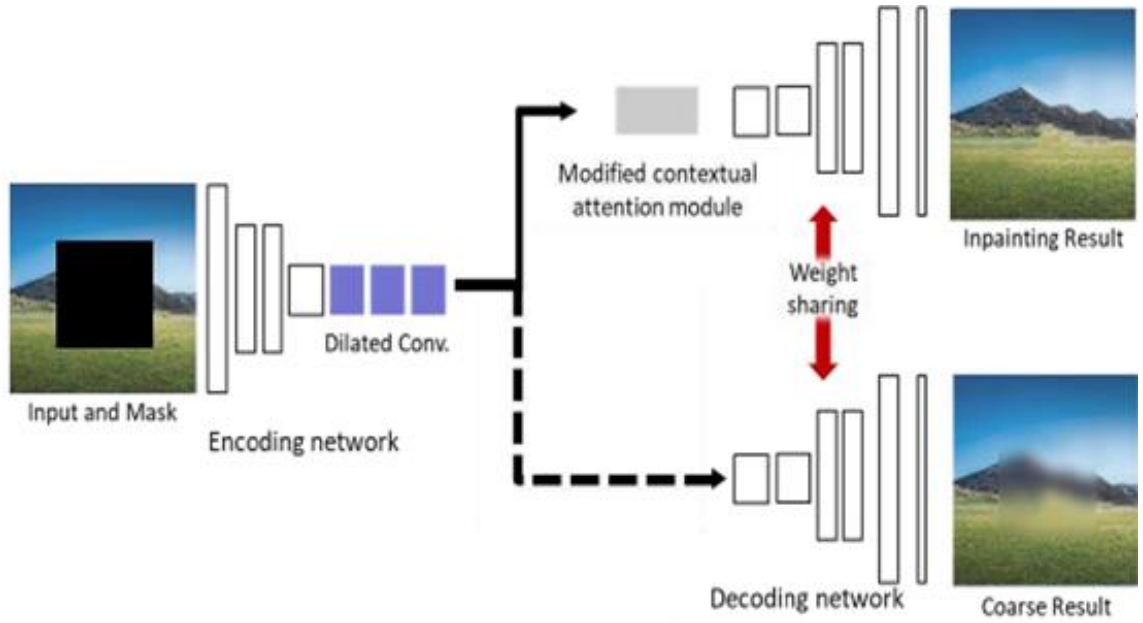


Type	Kernel	Dilation	Stride	Outputs
Convolution	5×5	1	1×1	32
Convolution	3×3	1	2×2	64
Convolution	3×3	1	1×1	64
Convolution	3×3	1	2×2	128
Convolution	3×3	1	1×1	128
Convolution	3×3	1	2×2	256
Dialated convolution	3×3	2	1×1	256
Dialated convolution	3×3	4	1×1	256
Dialated convolution	3×3	8	1×1	256
Dialated convolution	3×3	16	1×1	256

Table 2. Detail architecture of encoding network.

Type	Kernel	Dilation	Stride	Outputs
Convolution $\times 2$	3×3	1	1×1	128
Nearest Neighbor ($\times 2 \uparrow$)	-	-	-	-
Convolution $\times 2$	3×3	1	1×1	64
Nearest Neighbor ($\times 2 \uparrow$)	-	-	-	-
Convolution $\times 2$	3×3	1	1×1	32
Nearest Neighbor ($\times 2 \uparrow$)	-	-	-	-
Convolution $\times 2$	3×3	1	1×1	16
Convolution (Output)	3×3	1	1×1	3

Table 3. Detail architecture of decoding network. The output layer consists of a convolution layer clipped value to the [-1, 1].



Modified CAM

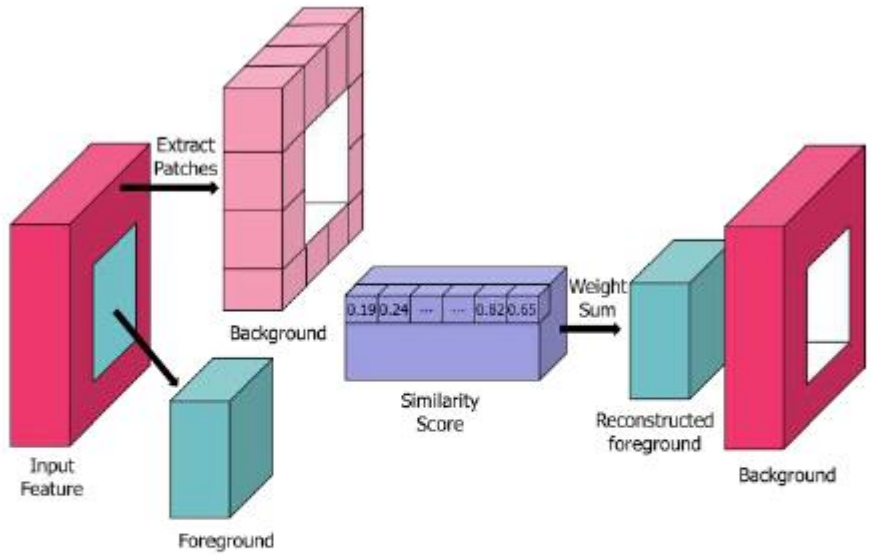


Figure 2. The illustration of the CAM. The conventional CAM reconstructs foreground patches by measuring the cosine similarities with background patches. In contrast, the modified CAM uses the Euclidean distance to compute similarity scores.

Output : [0,∞)

$$s_{(x,y),(x',y')} = \left\langle \frac{f_{x,y}}{\|f_{x,y}\|}, \frac{b_{x',y'}}{\|b_{x',y'}\|} \right\rangle, \tag{2}$$

$$s^*_{(x,y),(x',y')} = \text{softmax}(\lambda s_{(x,y),(x',y')}), \tag{3}$$

where λ is a hyper-parameter for scaled *softmax*. By using $(s^*_{(x,y),(x',y')})$ as weights, the CAM reconstructs features of foreground regions by a weighted sum of background patches to learn the relation between them.

Output : [−1, 1]

$$\tilde{d}_{(x,y),(x',y')} = \tanh\left(-\left(\frac{d_{(x,y),(x',y')} - m(d_{(x,y),(x',y')})}{\sigma(d_{(x,y),(x',y')})}\right)\right), \tag{4}$$

where

$$d_{(x,y),(x',y')} = \|f_{x,y} - b_{x',y'}\|. \tag{5}$$

the truncated distance similarity scores

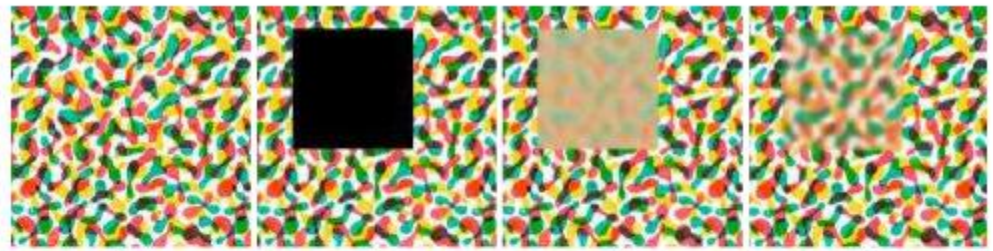


Figure 4. A comparison of the image reconstruction between the cosine similarity and the truncated distance similarity: (a) The original image, (b) masked image, (c) image reconstructed by using the cosine similarity and (d) image reconstructed by using the truncated distance similarity.

	square mask		free-form mask	
	PSNR	SSIM	PSNR	SSIM
Cosine similarity	25.16	0.8950	27.95	0.9218
Euclidean distance	25.57	0.9007	28.59	0.9293

Table 4. Comparison of the performance between the cosine similarity and the Euclidean distance applying on the PEPSI.

RED(region ensemble discriminator)

- combine global and local discriminators

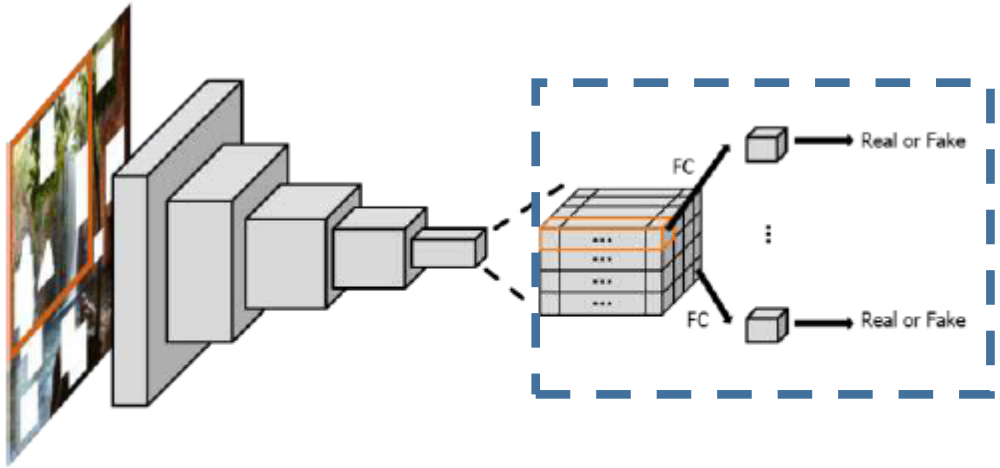


Figure 5. The overview of the RED. The RED aims to classify hole regions which may appear any region with any sizes in an image.

Type	Kernel	Stride	Outputs
Convolution	5×5	2×2	64
Convolution	5×5	2×2	128
Convolution	5×5	2×2	256
Convolution	5×5	2×2	256
Convolution	5×5	2×2	256
Convolution	5×5	2×2	512
FC	1×1	1×1	1

Table 5. Detailed architecture of RED. After each convolution layer, except last one, there is a leaky-ReLU as the activation function. Every layer is normalized by a spectral normalization. The fully-connected layer is applied to every pixel-wise feature block.

Figure 8. Comparison of our method and conventional methods on free-form masked CelebA-HQ datasets. (a) The ground truth (b) The input image of the network (c)Results of the Context Encoder [21] (d) Results of the Globally-Locally [10] (e) Results of the GatedConv [27] (f) Results of the proposed method

Method	Square mask			Free-form mask			Time (ms)
	PSNR		SSIM	PSNR		SSIM	
	Local	Global		Local	Global		
CE [21]	17.7	23.7	0.872	9.7	16.3	0.794	**5.8**
GL [10]	<u>19.4</u>	<u>25.0</u>	0.896	15.1	21.5	0.843	39.4
GCA [28]	19.0	24.9	<u>0.898</u>	12.4	18.9	0.798	22.5
GatedConv [27]	18.7	24.7	0.895	<u>21.2</u>	<u>27.8</u>	<u>0.925</u>	21.4
GatedConv *	17.5	23.5	0.882	19.8	26.4	0.910	14.3
PEPSI(Ours)	**19.5**	**25.6**	**0.901**	**22.0**	**28.6**	**0.929**	<u>9.2</u>
PEPSI *	19.2	25.2	0.894	21.6	28.2	0.923	

Table 6. Results of global and local PSNR, SSIM and operation time with both of square and free-formed masks on CelebA-HQ dataset. * means a model without coarse results.

PEPSI++: Fast and Lightweight Network for Image Inpainting

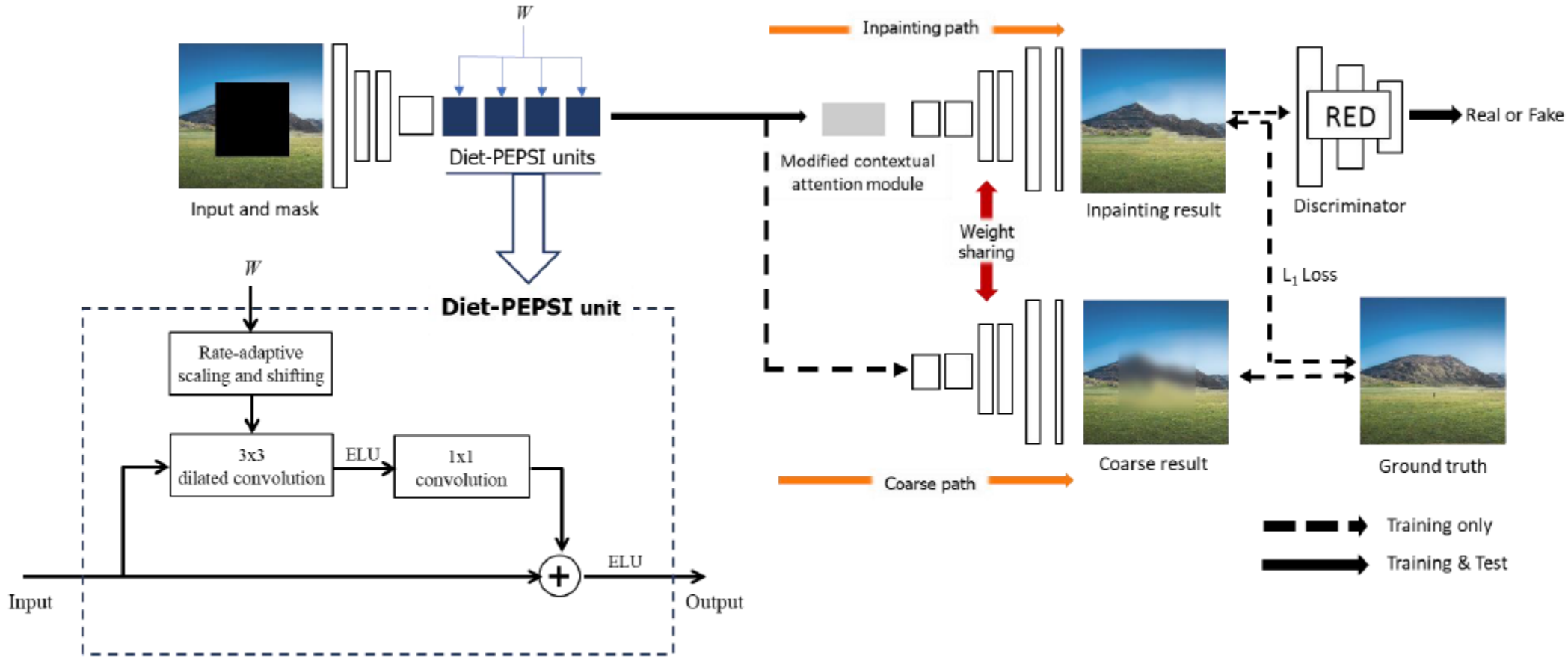


Fig. 6. Architecture of Diet-PEPSI. We replace the multiple dilated convolutional layers with DPUs. In the DPUs, rate-adaptive convolution layers share their weights whereas the 1×1 standard convolutional layers do not share their weights.

the rate-adaptive dilated convolutional layers

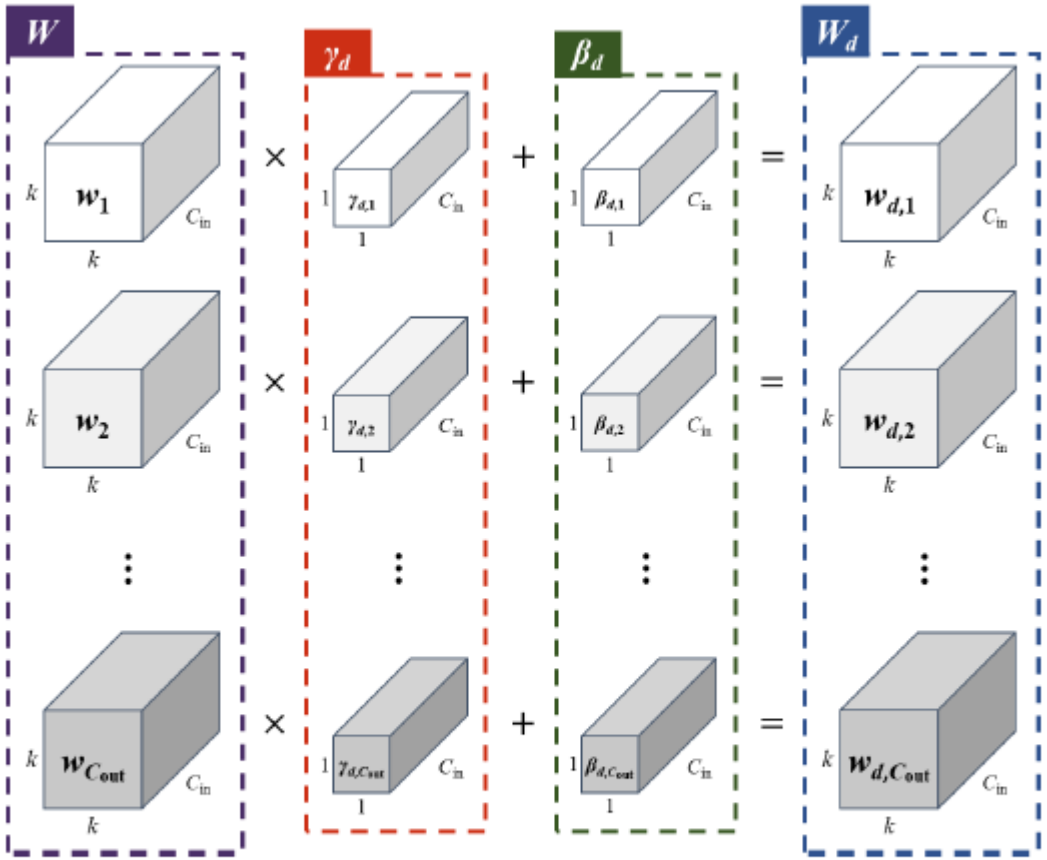


Fig. 5. Rate-adaptive scaling and shifting operations. β_d and γ_d have different values depending on the given rate. Tensor broadcasting is included in scaling and shifting operations.

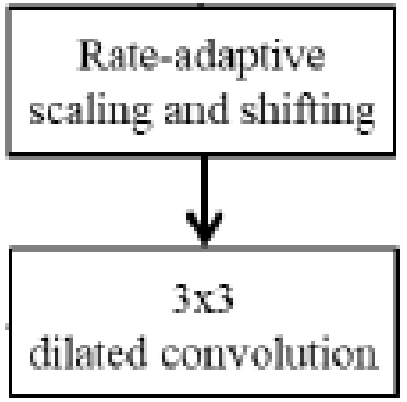
$$W_d = \gamma_d \cdot W + \beta_d,$$

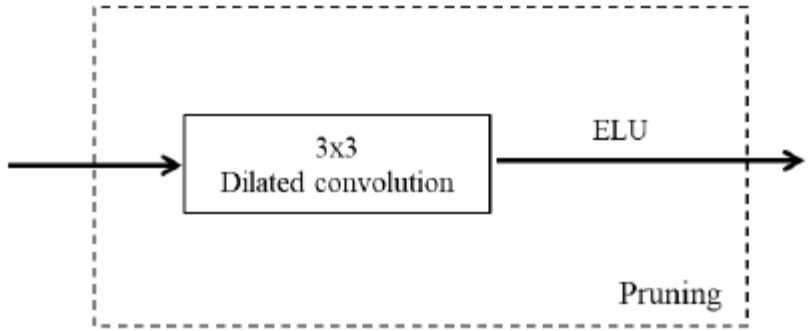
scale $\gamma_d \in \mathbb{R}^{1\times1\times C_{\text{in}}\times C_{\text{out}}}$ and bias $\beta_d \in \mathbb{R}^{1\times1\times C_{\text{in}}\times C_{\text{out}}}$

$$y = x \otimes (\gamma_d W + \beta_d) = x \otimes \gamma_d W + x \otimes \beta_d,$$

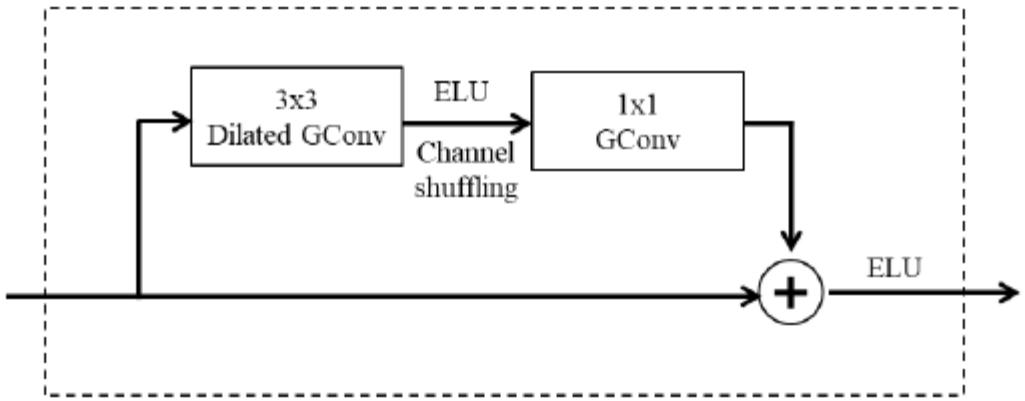
$(9+3n)\times C_{\text{in}}\times C_{\text{out}}$ network parameters

$3\times3\times C_{\text{in}}\times C_{\text{out}}\times n$ network parameters





(a)



(b)

Fig. 12. Illustration of techniques to aggregate the global contextual information while reducing the number of parameters. (a) Dilated convolutional layer with pruning channel. (b) Residual block consisting of group convolutional layers.

TABLE VIII
EXPERIMENTAL RESULTS USING DIFFERENT LIGHTWEIGHT UNITS.

	Square mask		Free-form mask	
	PSNR	SSIM	PSNR	SSIM
Pruning	25.21	**0.8961**	28.28	0.9270
DGC	25.28	0.8959	28.43	0.9270
DPU	**25.38**	0.8960	**28.53**	**0.9278**

TABLE V

RESULTS OF GLOBAL AND LOCAL PSNRS, SSIM, AND OPERATION TIME WITH SQUARE AND FREE-FORMED MASKS ON CELEBA-HQ DATASET.

Method	Square mask			Free-form mask			Time (ms)	Number of Network Parameters
	PSNR		SSIM	PSNR		SSIM		
	Local	Global		Local	Global			
CE [10]	17.7	23.7	0.872	9.7	16.3	0.794	**5.8**	5.1M
GL [7]	19.4	25.0	0.896	15.1	21.5	0.843	39.4	5.8M
GCA [4]	19.0	24.9	0.898	12.4	18.9	0.798	22.5	2.9M
GatedConv [19]	18.7	24.7	0.895	21.2	27.8	0.925	21.4	4.1M
PEPSI	**19.5**	**25.6**	**0.901**	**22.0**	**28.6**	**0.929**	9.2	3.5M
PEPSI w/o coarse path	19.2	25.2	0.894	21.6	28.2	0.923		
Diet-PEPSI	19.4	25.5	0.898	**22.0**	28.5	0.928	10.9	2.5M

TABLE VI

EXPERIMENTAL RESULTS THAT FURTHER REDUCE THE NETWORK PARAMETERS USING THE GROUP CONVOLUTION TECHNIQUE.

	Square mask		Free-form mask		Number of parameters
	PSNR	SSIM	PSNR	SSIM	
PEPSI	25.6	0.901	28.6	0.929	3.5M
Diet-PEPSI	25.5	0.898	28.5	0.928	2.5M
Diet-PEPSI ($g = 2$)	25.4	0.896	28.5	0.928	1.8M
Diet-PEPSI ($g = 4$)	25.2	0.894	28.4	0.926	1.5M