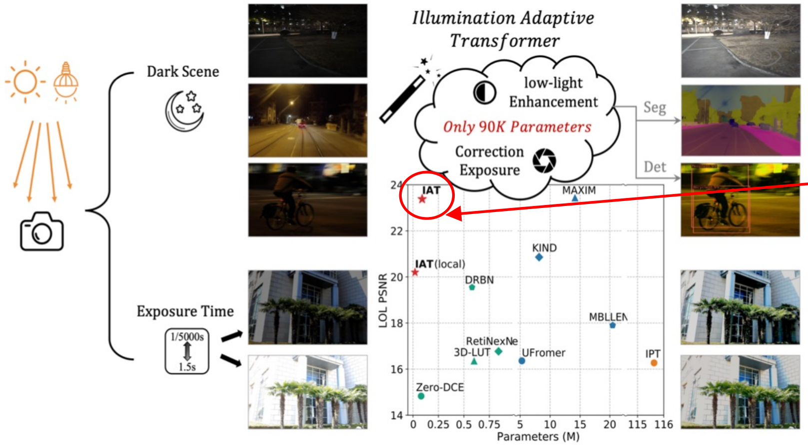
You Only Need 90K Parameters to Adapt Light: a Light Weight Transformer for Image Enhancement and Exposure Correction

Ziteng Cui[1], Kunchang Li[2], Lin Gu[3,1*], Shenghan Su[4], Peng Gao[2], Zhengkai Jiang[5], Yu Qiao[2], Tatsuya Harada[1,3]

1. The University of Tokyo 2. Shanghai AI Laboratory 3. RIKEN AIP
4. Shanghai Jiao Tong University 5. TenCent YouTu Lab

Github Link

Paper Link



Illumination-Adaptive-Transformer

Only 90K Parameters !
0.004s inference speed per image !
State-Of-The-Art !

Several tasks:

(1) Low-light enhancement
(2) Exposure correction
(3) Low-light object detection & Low-light semantic segmentation
(4) Various-light condition object detection

Enhancement Result

Table 1: Experimental results on LOL (V1 & V2) [64] datasets, best and second best results are marked in red and blue respectively, noted here [22] is non-deep learning method and [21] is self-supervised learning method.

Methods	LOL-V1		LOL-V2-real		Efficiency		
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	FLOPs(G)↓	#Params(M)↓	test time(s)↓
LIME* [22]	16.67	0.560	15.24	0.470	-	-	3.241 (M)
Zero-DCE* [21]	14.83	0.531	14.32	0.511	2.53	0.08	0.002 (P)
RetiNexNet [64]	16.77	0.562	18.37	0.723	587.47	0.84	0.841 (T)
MBLLEN [43]	17.90	0.702	18.00	0.715	19.95	20.47	1.981 (T)
DRBN [70]	19.55	0.746	20.13	0.820	37.79	0.58	1.210 (P)
3D-LUT [74]	16.35	0.585	17.59	0.721	7.67	0.6	0.006 (P)
KIND [77]	20.86	0.790	19.74	0.761	356.72	8.16	0e38 (T)
UFormer [63]	16.36	0.771	18.82	0.771	12.00	5.29	0.248 (P)
IPT [9]	16.27	0.504	19.80	0.813	2087.35	115.63	1.365 (P)
RCT [33]	22.67	0.788	-	-	-	-	-
MAXIM [59]	23.43	0.863	22.86	0.818	216.00	14.14	0.602 (P)
IAT (local)	20.20	0.782	20.30	0.789	1.31	0.02	0.002 (P)
IAT	23.38	0.809	23.50	0.824	1.44	0.09	0.004 (P)

Exposure Correction Result

Table 3: Experimental results on exposure correction dataset [2]. Note here HE and LIME [22] are non-deep learning methods. PSNR, SSIM and PI results, reported by competing works, are from [2].

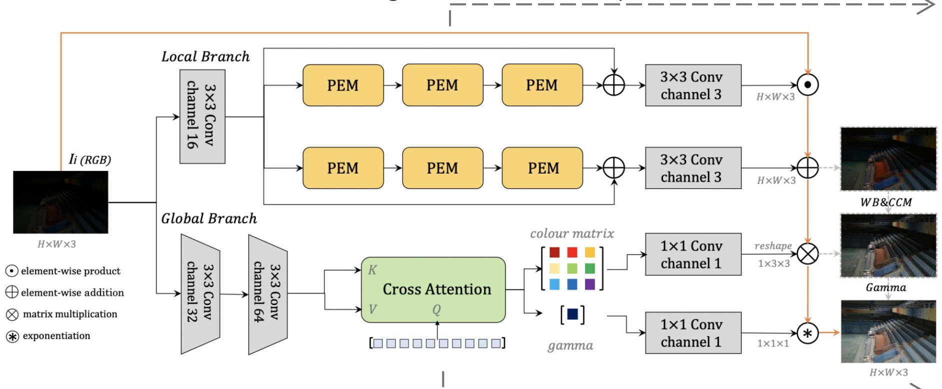
Method	Expert A		Expert B		Expert C		Expert D		Expert E		Avg		PI↓
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	
HE* [20]	16.14	0.685	16.28	0.671	16.52	0.696	16.63	0.668	17.30	0.688	16.58	0.682	2.405
LIME* [22]	11.15	0.590	11.83	0.610	11.52	0.607	12.64	0.628	13.61	0.653	12.15	0.618	2.432
DPED [29] (Sony)	17.42	0.675	18.64	0.701	18.02	0.683	17.55	0.660	17.78	0.663	17.88	0.676	2.806
DPE [13] (S-FiveK)	16.93	0.678	17.70	0.668	17.74	0.696	17.57	0.674	17.60	0.670	17.51	0.677	2.621
RetinexNet [64]	10.76	0.585	11.61	0.596	11.13	0.605	11.99	0.615	12.67	0.636	11.63	0.607	3.105
Deep-UPE [62]	13.16	0.610	13.90	0.642	13.69	0.632	14.80	0.649	15.68	0.667	14.25	0.640	2.405
Zero-DCE [21]	11.64	0.536	12.56	0.539	12.06	0.544	12.96	0.548	13.77	0.580	12.60	0.549	2.865
MSEC [2]	19.16	0.746	20.10	0.734	20.20	0.769	18.98	0.719	18.98	0.727	19.48	0.739	2.251
IAT (local)	16.61	0.750	17.52	0.822	16.95	0.780	17.02	0.773	16.43	0.789	16.91	0.783	2.401
IAT	**19.90**	**0.817**	**21.65**	**0.867**	**21.23**	**0.850**	**19.86**	**0.844**	**19.34**	**0.840**	**20.34**	**0.844**	**2.249**

Model Structure:

target image ← $I_t = G(F(I_i))$, → input image

global ISP branch   local inverse branch

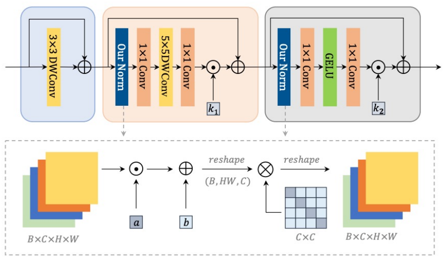
Our local branch is adopt depth-wise convolution for light weight design, consists of two branch to learn global and local map.



Our global branch is inspired by Detection Transformer (DETR)
Using attention queries to dynamic control ISP-related parameters

Pixel-wise Enhancement Module (PEM)

Depth-wise -Conv Transformer



Global Prediction Module (GPM)

Dynamic learnable attention queries