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

Ziteng Cui\textsuperscript{1}  
cui@mi.t.u-tokyo.ac.jp  
Kunchang Li\textsuperscript{2}  
kc.li@siat.ac.cn  
Lin Gu\textsuperscript{3,1*}  
lin.gu@riken.jp  
Shenghan Su\textsuperscript{4}  
su2564468850@sjtu.edu.cn  
Peng Gao\textsuperscript{2}  
 gaopeng@pjlab.org.cn  
Zhengkai Jiang\textsuperscript{5}  
zhengkjiang@tencent.com  
Yu Qiao\textsuperscript{2}  
qiaoyu@pjlab.org.cn  
Tatsuya Harada\textsuperscript{1,3}  
harada@mi.t.u-tokyo.ac.jp

\textsuperscript{1} The University of Tokyo  
\textsuperscript{2} Shanghai AI Laboratory  
\textsuperscript{3} RIKEN AIP  
\textsuperscript{4} Shanghai Jiao Tong University  
\textsuperscript{5} Tencent Youtu Lab

Abstract

Challenging illumination conditions (low light, under-exposure and over-exposure) in the real world not only cast an unpleasant visual appearance but also taint the computer vision tasks. After camera captures the raw-RGB data, it renders standard sRGB images with image signal processor (ISP). By decomposing ISP pipeline into local and global image components, we propose a lightweight fast Illumination Adaptive Transformer (IAT) to restore the normal lit sRGB image from either low-light or under/over-exposure conditions. Specifically, IAT uses attention queries to represent and adjust the ISP-related parameters such as colour correction, gamma correction. With only $\sim 90k$ parameters and $\sim 0.004s$ processing speed, our IAT consistently achieves superior performance over State-of-The-Art (SOTA) on the benchmark low-light enhancement and exposure correction datasets. Competitive experimental performance also demonstrates that our IAT significantly enhances object detection and semantic segmentation tasks under various light conditions. Our code and pre-trained model is available at this url.

1 Introduction

Computer vision has witnessed great success on well-taken images and videos. However, the varying light conditions in the real world poses challenges on both human visual appearance

© 2022. The copyright of this document resides with its authors.  
It may be distributed unchanged freely in print or electronic forms.
and downstream vision tasks (e.g., semantic segmentation and object detection). Images taken under inadequate illumination (Fig.1 left top) suffer from limited photon counts and undesirable in-camera noise. On the other hand, outdoor scenes are often exposed to strong light such as direct sunlight (Fig.1 left down), making image saturated due to the limited range of sensors and non-linearity in the camera image pipeline. To make it worse, both the under and over exposure may exist together, i.e. spatial-variant illumination cast by shadow could make the contrast ratio to be 1000:1 or higher.

Multiple techniques such as low-light enhancement \([19, 20, 31, 33, 39, 40, 42, 51, 53, 55, 69, 72]\), exposure correction \([2, 43, 44, 64, 67]\) have been proposed to adapt to the difficult light condition. Low-light enhancement methods restore the details while suppressing the accompanying noises. Exposure correction methods focus on adjusting the under/over exposure image to reconstruct a clear image against short/long exposure time. While many efforts elaborate on improving human oriented visual perceptual, there also several methods enhance the high-level tasks by boosting their robustness against low light \([15, 25, 36, 50]\) and over-exposure conditions \([41]\). As shown in Fig.1, we aim for a unified lightweight framework that improves both the visual appearance and consequent recognition tasks under challenging real-world light condition.

While sRGB images are most common to everyday life, many existing light adaptive methods operate on raw-RGB images which linearly proportion to the actual scene irradiance. To directly process the sRGB images, we specifically considers the image signal processor (ISP) in camera that renders sRGB from raw-RGB image. We propose a novel two-branches transformer based model to handle this issue, a pixel-wise local branch \(f\) coupled with global ISP branch \(g\). In local branch \(f\), we map the input image to latent feature space and replace transformer’s attention block to depth-wise convolution for light-weight design. In global branch \(g\), we use transformer’s attention queries to control and adjust the global ISP-related parameters (i.e. colour transform matrix, gamma value). In addition, the learned queries could dynamic change under different light condition at same time (i.e. over-exposure and under exposure).

Extensive experiments are conducted on several real-world and synthetic datasets, i.e.,
image enhancement dataset LOL [60] and photo retouching dataset MIT-Adobe FiveK [6], low-light detection dataset EXDark [38] and low-light segmentation dataset ACDC [49]. Results show that our IAT achieve state-of-the-art performance across a range of low-level and high-level tasks. More importantly, our IAT model is of only 0.09M parameters, much smaller than current SOTA transformer-based models [8, 55, 59]. Besides, our average inference speed is 0.004s per image, faster than the SOTA methods taking around 1s per image.

Our contribution could be summarised as follow:

- We have proposed a fast and light-weight framework, Illumination Adaptive Transformer (IAT), to adapt to challenging light conditions in the real world, which could both handle the low-light enhancement and exposure correction tasks.

- We have proposed a novel transformer-style structure to estimate ISP-related parameters to fuse the target sRGB image, wherein the learnable attention queries are utilised to attend the whole image, also we replace the layer norm to a new light normalisation, for better handling low-level vision tasks.

- Extensive experiments on several real-world datasets on 3 low-level tasks and 3 high-level tasks demonstrate the superior performance of IAT over SOTA methods. IAT is light weight and mobile-friendly with only 0.09M model parameters and 0.004s processing time per image. We will release the source code upon publication.

2 Related Works

2.1 Enhancement against Challenging Light Condition

Earlier low-light image enhancement solutions mainly rely on RetiNex theory [33] or histogram equalization [18, 53]. Since LLNet [39] utilised a deep-autoencoder structure, CNN based methods [17, 19, 31, 40, 42, 51, 58, 63, 64, 72] have been widely used in this task and gain SOTA results on the benchmark enhancement datasets [5, 60].

Similar to low-light enhancement, traditional exposure correction algorithms [43, 67] also use image histograms to adjust image intensities. The strategy then tends to correct exposure errors by adjusting the tone curve via a trained deep learning model [45, 66]. Very recently, Afifi et al. [2] propose a coarse-to-fine neural network to correct photo exposure, after that Nsampi et al. [44] introduce attention mechanism into this task.

Beyond low-level vision, low-light/ strong-light scenario also deteriorates the performance of high level vision [15, 25, 36, 41, 50, 73]. Several methods based on data synthesis [73], self-supervised learning [15] and domain adaptation [50] have been proposed to support high level vision tasks under challenging illumination conditions.

2.2 Vision Transformers

Transformer [57] was firstly proposed in NLP area to capture long-range dependencies by global attention. ViT [16] made the first attempt in vision task by splitting the image into tokens before sending into transformer model. Since then, Transformer based models have gained superior performances in many computer vision tasks, including image/video classification [34, 37], object detection [6, 71], semantic segmentation [62], vision-language model [46, 70] and so on.

In low-level vision area, transformer-based models has also made much progress on several sub-directions, such as image super-resolution [53], image restoration [8, 59, 68], image
colorization [3] and bad weather restoration [5]. Very recently, MAXIM [55] use MLP-based model in low-level vision area which also shows MLP’s potential on low-level vision tasks. However, existing transformer & MLP models require much computational cost (e.g. 115.63M for IPT [8], 14.14M for MAXIM [55]), making it hard to implement on mobile and edge devices. Extreme lightweight of our method (0.09M) is particular important in low-level vision and computational photography.

3 Illumination Adaptive Transformer

3.1 Motivation

For a sRGB image $I_i$ taken from light condition $L_i$, the input photons under light condition $L_i$ would project through the lens on capacitor cluster, to pass by the in-camera process [61] and render with image signal processor (ISP) pipeline $G(\cdot)$ [4, 30]. Our goal is to match input sRGB image $I_i$ to the target sRGB image $I_t$ (taken under light condition $L_t$). Existing deep-learning based methods tend to build an end-to-end mapping between $I_i$ and $I_t$ [2, 39, 40] or estimate some high-level representation to assist enhancement task (i.e. illumination map [58], colour transform function [31], 3D look-up table [69]). However, the actual lightness degradation happens in raw-RGB space, and the processes in camera ISP involves more elaborated non-linear operations such as white balance, colour space transform, gamma correction, etc. Therefore, much of research conducts image enhancement [7, 61] directly on raw-RGB data rather than sRGB images.

To this end, Brooks et al. [4] inverse each steps in ISP pipeline (i.e. gamma correction, tone mapping, camera colour transformation) to transform input sRGB image to "unprocessed" raw-RGB data. After that, Afifi and Brown [1] apply an encoder-decoder structure to edit the illumination of sRGB image from input light $I_i$ to target light $I_t$ as following:

$$I_t = G(F(I_i)), \quad (1)$$

where $F$ is an unknown reconstruction function maps $I_i$ to the corresponding raw-RGB data $D = F(I_i)$, and $G$ is camera rendering function that transform $D$ back to target sRGB image $I_t$. Here [1] use the network encoder $f$ to represent $F$, before adding several individual decoders $g_t$ upon encoder $f$. The function maps $f(I_i)$ to target $I_t$ illumination conditions is represented below:

$$I_t = g_t(f(I_i)), \quad (2)$$

For the sake of lightweight network design, inspired by the DETR [6] which controls different object proposals via transformer queries, here we use different queries to control the ISP-related parameters in $g_t(\cdot)$. This re-configures parameters to make the image $I_t$ adaptive to target light condition $L_t$. In training stage, the queries is dynamically updated in each iteration to match the target image $I_t$. Here we simplify the ISP procedures [4, 11, 15] into the equation 3 below. The simplification details could be found in the supplementary.

$$g_t(\cdot) = (\max(\sum_{c_j} W_{c_i,c_j}(\cdot), \epsilon))^{\gamma}, c_i,c_j \in \{r, g, b\}. \quad (3)$$

$W_{c_i,c_j}$ is a $3 \times 3$ joint colour transformation matrix, considering the white balance and colour transform matrix. We adopt 9 queries to control $W_{c_i,c_j}$’s parameters. $\gamma$ denotes the
gamma correction parameter which we use a single query to control. $\varepsilon$ is a very small value to prevent numerical instability. Here we set $\varepsilon = 1e^{-8}$ in our experiments.

For process $F$, we apply a pixel-wise least squares model $f$. Our $f$ consists of two individual branches to predict multiply map $M$ and add map $A$. We then apply a least squares to process input sRGB image: $f(I_i) = I_i \odot M + A$. Here $M$ and $A$ has the same size with $I_i$ to complete pixel-level multiplicative and additive adjustment. Finally, the equation of our IAT model follows:

$$I_t = (\max(\sum_{c_i,c_j} W_{c_i,c_j} (I_i \odot M + A)), 0)^\gamma.$$ (4)

The non-linear operations are decomposed into a local pixel-wise components $f$ and a global ISP components $g$. Thus, we design two individual transformer style branches: local adjustment branch and global ISP branch, to estimate the local pixel-wise components and global ISP components respectively.

### 3.2 Model Structure

Given an input sRGB image $I_i \in \mathbb{R}^{H \times W \times 3}$ under light condition $L_i$, where $H \times W$ denotes the size dimension and 3 denotes the channel dimension ($\{r, g, b\}$). As shown in Fig.2, we propose our Illumination Adaptive Transformer (IAT) to transfer the input RGB image $I_i$ to a target RGB $I_t \in \mathbb{R}^{H \times W \times 3}$ under the proper uniform light $L_t$.

**Local Branch.** In the local branch, we focus on estimating the local components $M, A$ to correct the effect of illumination. Instead of adopting a U-Net [48] style structure, which downsamples the images first before upsampling, we aim to maintain the input resolution through the local branch to preserve the informative details. Therefore, we propose a transformer-style architecture for the local branch. Compared to popular U-Net style structures [2, 40], our structure could deal with arbitrary resolution images without resizing them.

At first, we expand the channel dimension via a $3 \times 3$ convolution and pass them to two independent branches stacked by Pixel-wise Enhancement Module (PEM). For the lightweight design in the local branch, we replace self-attention with depth-wise convolution as suggested in the previous works [21, 34], depth-wise convolution could reduce parameters and...
Our Norm $1 \times 1$ Conv $5 \times 5$ DWConv $1 \times 1$ Conv GELU $1 \times 1$ Conv

Figure 3: Detailed structure of Pixel-wise Enhancement Module (PEM) and Global Prediction Module (GPM).

further save computation cost. As shown in Fig. 3 (a), our PEM first encodes the position information by $3 \times 3$ depth-wise convolution before enhancing local details with PWConv-DWConv-PWConv. Finally, we adopt two $1 \times 1$ convolutions to enhance token representation individually. Specially, we design new kind of normalisation names light normalisation, to replace transformer’s Layer Normalisation \[3\]. As shown in Fig. 3 (a), light normalisation learns to scale $a$ and bias $b$ via two learnable parameters before fusing the channels via the learnable matrix. The matrix is initialised as an identity matrix. For better convergence, we adopt Layer Scale \[54\] which multiplies the features by a small number $k_1/k_2$.

We stack 3 PEMs in each branch and then connect the output features with the input features through element-wise addition. This skip connection \[22\] helps maintain the original image details. Finally, we decrease the channel dimension by a $3 \times 3$ convolutions and adopt ReLU/ Tanh function to generate the local components $M/A$ in Eq. 4.

Global ISP Branch. Global ISP branch accounts for part of the ISP pipeline \[4, 23, 28, 30\] (i.e. gamma correction, colour matrix transform, white balance) when transferring the target RGB image $I_t$. Specifically, the value of each pixel in the target image is determined by a global operation defined in Eq.3.

Inspired by Detection Transformer DETR \[6\], we design global component queries to decode and predict the $W, \gamma$ to generate sRGB image $I_t$. This transformer structure allows capturing global interactions between context and individual pixels. As shown in Fig. 2, we first stack two convolutions as a lightweight encoder, which encodes the features in a high dimension with lower resolution, on the one hand lower resolution would save computational cost which contribute to the light-weight design, on the other hand higher feature representation would be helpful to extract image’s global-level features. Then the generated features are passed to the Global Prediction Module (GPM), Fig. 3 (b) shows the detailed structure of GPM, different from original DETR \[6\] model, our global component queries Q are initialised as zeros without extra multi-head self-attention. Q is global component learnable embedding that attends keys K and values V generated from encoded features. The positional encoding for K and V is from a depth-wise convolution, which is friendly with different input resolutions. After feed forward network (FFN) \[16\] with two linear layers, we add two extra parameters with special initialisation to output colour matrix and gamma. This initialisation makes sure the colour matrix is identity matrix $W$ and the gamma value $g$ is one in the beginning, thus contributing to stable training.
4 Experiments

We evaluate our proposed IAT model on benchmark datasets and experimental settings for both low-level and high-level vision tasks under different illumination conditions. Three low-level vision tasks include: (a) image enhancement (LOL (V1 & V2-real) [60]), (b) image enhancement (MIT-Adobe FiveK [5]), (c) exposure correction [2]. Three high-level visions tasks include: (d) low-light object detection (e) low-light semantic segmentation (f) various-light object detection. The number of PEM number in local branch are both set to 3, while the channel number in PEM is set to 16.

For all low-level vision experiments: \{(a), (b), (c)\}, the IAT model are trained on a single GeForce RTX 3090 GPU with batch size 8. We use Adam optimizer to train our IAT model while the initial learning rate and weight decay are separately set to $2e^{-4}$ and $1e^{-4}$. A cosine learning schedule has also been adopted to avoid over-fitting. For data augmentation, horizontal and vertical flips have been used to acquire better results.

4.1 Low-level Image Enhancement.

For (a) and (b) image enhancement task, we evaluate our IAT framework on benchmark datasets: LOL (V1 & V2-real) [60] and MIT-Adobe FiveK [5].

LOL [60] has two versions: LOL-V1 consists of 500 paired normal-light images and low-light images. 485 pairs are used for training and the other 15 pairs are for testing. LOL-
Table 1: Experimental results on LOL (V1 & V2) datasets, best and second best results are marked in red and blue respectively, noted here is non-deep learning method and is self-supervised learning method.

<table>
<thead>
<tr>
<th>Methods</th>
<th>LOL-V1</th>
<th>LOL-V2-real</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PSNR↑</td>
<td>SSIM↑</td>
<td>PSNR↑</td>
</tr>
<tr>
<td>LIME* [20]</td>
<td>16.67</td>
<td>0.560</td>
<td>15.24</td>
</tr>
<tr>
<td>Zero-DCE* [20]</td>
<td>14.83</td>
<td>0.531</td>
<td>14.32</td>
</tr>
<tr>
<td>RetiNexNet [20]</td>
<td>16.77</td>
<td>0.562</td>
<td>18.37</td>
</tr>
<tr>
<td>MBLLEN [20]</td>
<td>17.90</td>
<td>0.702</td>
<td>18.00</td>
</tr>
<tr>
<td>DRBN [20]</td>
<td>19.55</td>
<td>0.746</td>
<td>20.13</td>
</tr>
<tr>
<td>3D-LUT [20]</td>
<td>16.35</td>
<td>0.585</td>
<td>17.59</td>
</tr>
<tr>
<td>KIND [20]</td>
<td>20.86</td>
<td>0.790</td>
<td>19.74</td>
</tr>
<tr>
<td>UFormer [20]</td>
<td>16.36</td>
<td>0.771</td>
<td>18.82</td>
</tr>
<tr>
<td>IPT [26]</td>
<td>16.27</td>
<td>0.504</td>
<td>19.80</td>
</tr>
<tr>
<td>RCT [31]</td>
<td>22.67</td>
<td>0.788</td>
<td>-</td>
</tr>
<tr>
<td>MAXIM [31]</td>
<td>23.43</td>
<td>0.863</td>
<td>22.86</td>
</tr>
<tr>
<td>IAT (local)</td>
<td>20.20</td>
<td>0.782</td>
<td>20.30</td>
</tr>
<tr>
<td>IAT</td>
<td>23.38</td>
<td>0.809</td>
<td>23.50</td>
</tr>
</tbody>
</table>

V2-real consists of 789 paired normal-light images and low-light pairs. 689 pairs are used for training and the other 100 pairs are for testing. The loss function between input image \( I_i \) and target image \( I_t \) for LOL dataset training is a mixed loss function [56] consisting of smooth L1 loss and VGG loss [29]. In LOL-V1 training, the images are cropped into 256 × 256 to train 200 epochs and then fine-tune on 600 × 400 resolution for 100 epochs. In LOL-V2-real training, the image resolution is maintained at 600 × 400 and trained for 200 epochs. Both LOL-V1 and LOL-V2-real testing the image resolution is maintained at 600 × 400.

We compare our method with SOTA methods [8, 19, 20, 31, 40, 55, 59, 60, 65, 69, 72]. For image quality analysis, we evaluate the peak signal-to-noise ratio (PSNR) and structural similarity index measure (SSIM). For the model efficiency analyze, we report three metrics: FLOPs, model parameters and test time, as shown in the last column of Table 1. We list different model’s test time on their corresponding code platform (M means Matlab, T means TensorFlow, P means PyTorch). As shown in Table 1, IAT (local) only uses the local network to train the model and IAT refers to the whole framework. Our IAT gains SOTA result on both image quality and model efficiency, especially less than 100× FLOPs and parameters usage compare to the current SOTA methods MAXIM [31].

Table 2: Experimental results on MIT-Adobe FiveK dataset.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PSNR↑</td>
<td>18.57</td>
<td>21.57</td>
<td>23.80</td>
<td>21.76</td>
<td>23.04</td>
<td>23.63</td>
<td>25.21</td>
<td>25.32</td>
</tr>
<tr>
<td>SSIM↑</td>
<td>0.701</td>
<td>0.843</td>
<td>0.880</td>
<td>0.871</td>
<td>0.893</td>
<td>0.875</td>
<td>0.922</td>
<td>0.920</td>
</tr>
<tr>
<td>#Params.↓</td>
<td>-</td>
<td>1.3M</td>
<td>3.3M</td>
<td>-</td>
<td>1.0M</td>
<td>0.8M</td>
<td>0.6M</td>
<td>0.09M</td>
</tr>
</tbody>
</table>

MIT-Adobe FiveK dataset contains 5000 images, each was manually enhanced by five different experts (A/B/C/D/E). Following the previous settings [26, 29], we only use experts C’s adjusted images as ground truth images. For MIT-Adobe FiveK dataset training, we use a single L1 loss function to optimize IAT model. Our method is compared with SOTA enhancement methods [12, 26, 27, 42, 48, 58, 58, 69] on FiveK dataset. The image quality results (PSNR, SSIM) and model parameters are reported in Table 2. Our IAT also gain satisfactory result in both quality and efficiency. Qualitative results of LOL and FiveK has been shown in Fig.4. More results could be found in supplementary material.
Table 3: Experimental results on exposure correction dataset [2]. Note here HE and LIME [20] are non-deep learning methods. PSNR, SSIM and PI results, reported by competing works, are from [2].

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>HE* [18]</td>
<td>16.14 0.685</td>
<td>16.28 0.671</td>
<td>16.52 0.696</td>
<td>16.63 0.668</td>
<td>17.30 0.688</td>
<td>16.58 0.682</td>
<td>2.405</td>
</tr>
<tr>
<td>LIME* [20]</td>
<td>11.15 0.590</td>
<td>11.83 0.610</td>
<td>11.52 0.607</td>
<td>12.64 0.628</td>
<td>13.61 0.653</td>
<td>12.15 0.618</td>
<td>2.432</td>
</tr>
<tr>
<td>DPED [3] (Sony)</td>
<td>17.42 0.675</td>
<td>18.64 0.701</td>
<td>18.02 0.683</td>
<td>17.55 0.660</td>
<td>17.78 0.663</td>
<td>17.88 0.676</td>
<td>2.806</td>
</tr>
<tr>
<td>DPE [3] (S-FiveK)</td>
<td>16.93 0.678</td>
<td>17.70 0.668</td>
<td>17.74 0.696</td>
<td>17.57 0.674</td>
<td>17.60 0.670</td>
<td>17.51 0.677</td>
<td>2.621</td>
</tr>
<tr>
<td>RetinexNet [2]</td>
<td>10.76 0.585</td>
<td>11.61 0.596</td>
<td>11.13 0.605</td>
<td>11.99 0.615</td>
<td>12.67 0.636</td>
<td>11.63 0.607</td>
<td>3.105</td>
</tr>
<tr>
<td>Deep-UPE [3]</td>
<td>13.16 0.610</td>
<td>13.90 0.642</td>
<td>13.69 0.632</td>
<td>14.80 0.649</td>
<td>15.68 0.667</td>
<td>14.25 0.640</td>
<td>2.405</td>
</tr>
<tr>
<td>Zero-DCE [3]</td>
<td>11.64 0.536</td>
<td>12.56 0.539</td>
<td>12.06 0.544</td>
<td>12.96 0.548</td>
<td>13.77 0.580</td>
<td>12.60 0.549</td>
<td>2.865</td>
</tr>
<tr>
<td>MSEC [2]</td>
<td>19.16 0.746</td>
<td>20.10 0.734</td>
<td>20.20 0.769</td>
<td>18.98 0.719</td>
<td>18.98 0.727</td>
<td>19.48 0.739</td>
<td>2.251</td>
</tr>
<tr>
<td>IAT (local)</td>
<td>16.61 0.750</td>
<td>17.52 0.822</td>
<td>16.95 0.780</td>
<td>17.02 0.773</td>
<td>16.43 0.789</td>
<td>16.91 0.783</td>
<td>2.401</td>
</tr>
<tr>
<td>IAT</td>
<td>19.90 0.817</td>
<td>21.65 0.867</td>
<td>21.23 0.850</td>
<td>19.86 0.844</td>
<td>19.34 0.840</td>
<td>20.34 0.844</td>
<td>2.249</td>
</tr>
</tbody>
</table>

Table 4: Experimental results on low-light detection dataset EXDark [38], low-light semantic segmentation dataset ACDC [49] and various light detection dataset TYOL [24].

<table>
<thead>
<tr>
<th>Methods</th>
<th>(d) EXDark Detection</th>
<th>(e) ACDC Segmentation</th>
<th>(f) TYOL Detection</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mAP↑ time(s)↓</td>
<td>mIOU↑ time(s)↓</td>
<td>mAP↑ time(s)↓</td>
</tr>
<tr>
<td>base-line</td>
<td>76.4 0.033</td>
<td>63.3 0.249</td>
<td>88.4 0.023</td>
</tr>
<tr>
<td>MBLLEN [40]</td>
<td>76.3 0.086</td>
<td>63.0 0.332</td>
<td>95.3 0.105</td>
</tr>
<tr>
<td>DeepLPF [42]</td>
<td>76.3 0.138</td>
<td>61.9 0.807</td>
<td>94.5 0.223</td>
</tr>
<tr>
<td>Zero-DCE [3]</td>
<td>76.9 0.042</td>
<td>61.9 0.300</td>
<td>95.2 0.030</td>
</tr>
<tr>
<td>IAT</td>
<td>77.2 0.040</td>
<td>62.1 0.280</td>
<td>95.8 0.027</td>
</tr>
</tbody>
</table>

4.2 Exposure Correction.

For the (c) exposure correction task, we evaluate IAT on the benchmark dataset proposed by [49]. The dataset contains 24,330 sRGB images, divided into 17,675 training images, 750 validation images, and 5905 test images. Images in [49] are adjusted by MIT-Adobe FiveK [5] dataset with 5 different exposure values (EV), ranging from under-exposure to over-exposure condition. Same as [5], test set has 5 different experts’ adjust results (A/B/C/D/E). Following the setting of [5], the training images are cropped to 512 × 512 patches and the test image is resized to have a maximum dimension of 512 pixels. We compare the test images with all five experts’ results. Here we use L1 loss function for exposure correction training.

The evaluation result is shown in Table 3. Our comparison methods include both traditional image processing methods (Histogram Equalization [18], LIME [20]) and deep learning methods (DPED [3], DPE [3], RetinexNet [3], Deep-UPE [3], Zero-DCE [3], MSEC [2]). Evaluation metrics are same as [5], including PSNR, SSIM and perceptual index (PI). Table 3 shows that our IAT model has gained best result on all evaluation indices. Compared to the second best result MSEC [2], IAT has much fewer parameters (0.09M v.s. 7M) and less evaluation time (0.004s per image v.s. 0.5s per image). Qualitative result has been shown in Fig.4 and more visual results are given in supplementary material.

4.3 High-level Vision

For high-level vision tasks: \{(d), (e), (f)\}, we use IAT to restore the image before feeding to the subsequent recognition algorithms based on mmdetection and mmsegmentation [49]. For a fair comparison, we run all of the experiments in the same setting: same input size, same data augmentation methods (expand, random crop, multi-size, random flip...), same training epochs and same initial weights. We train the recognition algorithm on the
datasets enhanced by IAT. We compare our methods with original datasets as well as datasets enhanced by other enhancement methods \cite{19, 40, 42}.

For object detection task in (d) EXDark dataset \cite{38} and (f) TYOL dataset \cite{24}. EXDark includes 7,363 real-world low-light images, ranging from twilight to extreme dark environment with 12 object categories. We take 80\% images of each category for training and the other 20\% for testing. TYOL includes 1680 images with 21 classes. We take 1365 images for training and other for evaluation. For both datasets, we perform object detection with YOLO-V3 \cite{47}, all the input images have been cropped and resized to $608 \times 608$ pixel size, we use SGD optimizer to train YOLO-V3 with batch size 8 for 25 epochs to EXDark and 45 epochs to TYO-L, the initial learning rate is $1 \times 10^{-3}$ and weight decay is $1 \times 10^{-4}$. The detection metric mAP and per-image evaluation time is shown in Table. 4. Our IAT model gains best results in both accuracy and speed compared to the baseline model and other enhancement methods \cite{19, 40, 42}.

For semantic segmentation in (e) ACDC dataset \cite{49}, we take 1006 night images in the ACDC dataset and then adopt DeepLab-V3+ \cite{10} to train on the ACDC-night train set and test on ACDC-night val set. The DeepLab-V3+ \cite{10} model is initialized by an Cityscape dataset \cite{14} pre-train model, we tuned the pre-train model by SGD optimizer with batch size 8 for 20000 iters, initial learning rate is set to 0.05, momentum and weight decay are separately set to 0.9 and $5 \times 10^{-4}$. We show the segmentation metric mIOU and per-image evaluation time in Table. 4, we found that all the enhancements methods invalid in this setting, this may because the lightness condition in ACDC \cite{49} is various and exceeds the generalisation ability of the enhancement model. For this problem, we propose to joint training our IAT model with following segmentation network (as well as detection network), which would solves this problem and improve the semantic segmentation/ object detection results in low-light conditions, detailed analyse please refer to Sec.B of supplementary material.

5 Conclusion

We propose a novel lightweight transformer framework IAT, by adapting ISP-related parameters to adapt to challenging light conditions. Despite its superior performance on several real-world datasets for both low-level and high-level tasks, IAT is extremely light with a fast speed. The lightweight and mobile-friendly IAT has the potential to become a standing tool for the computer vision community.

However, one main drawback of the IAT module is that, the image signal processor (ISP) has been simplified due to the light-weight demand, we think that more detailed ISP-related parts could be concerned and interpolate to the IAT module. In further, we’d also like to implement IAT on 3D human relighting task, to solve more complex lighting problems under 3D condition.

6 Acknowledgement

Corner symbol '*' in the author name means the corresponding author. This work supported by JST Moonshot R&D Grant Number JPMJMS2011 and JST, ACT-X Grant Number JP-MJAX190D, Japan. This work also supported by National Natural Science Foundation of China (Grant No. 62206272) and Shanghai Committee of Science and Technology (Grant No. 21DZ1100100).

\footnote{For more experimental details and ablation analyse, please refer to the supplementary material.}
References


