Perceptual Image Enhancement for Smartphone Real-Time Applications

Marcos V. Conde[1], Florin Vasluianu[1], Javier Vazquez-Corral[2], Radu Timofte[1]

[1]Computer Vision Lab, CAIDAS, IFI, University of Würzburg, Germany

[2]Computer Vision Center and Computer Science Dept., Universitat Autònoma de Barcelona, Spain

{marcos.conde-osorio,radu.timofte}@uni-wuerzburg.de

https://github.com/mv-lab/AISP

Abstract

Recent advances in camera designs and imaging pipelines allow us to capture high-quality images using smartphones. However, due to the small size and lens limitations of the smartphone cameras, we commonly find artifacts or degradation in the processed images. The most common unpleasant effects are noise artifacts, diffraction artifacts, blur, and HDR overexposure. Deep learning methods for image restoration can successfully remove these artifacts. However, most approaches are not suitable for real-time applications on mobile devices due to their heavy computation and memory requirements.

In this paper, we propose LPIENet, a lightweight network for perceptual image enhancement, with the focus on deploying it on smartphones. Our experiments show that, with much fewer parameters and operations, our model can deal with the mentioned artifacts and achieve competitive performance compared with state-of-the-art methods on standard benchmarks. Moreover, to prove the efficiency and reliability of our approach, we deployed the model directly on commercial smartphones and evaluated its performance. Our model can process 2K resolution images under 1 second in mid-level commercial smartphones.

1. Introduction

In recent years the number of images that are captured has increased exponentially. The main reason for this surge comes from the ubiquitous presence of smartphones in our daily life. Phone manufacturers are continuously competing with the goal of delivering better images to their consumers in order to increase their sales. Therefore, a lot of research has been focused on improving the perceptual quality of these sRGB images.

Image restoration aims at improving the images captured by the cameras by removing different degradations introduced during image acquisition. These degradations can be introduced due to the physical limitations of cameras, for
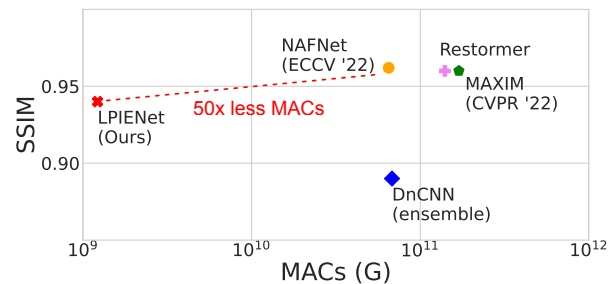

Figure 1. Comparison of computational cost and performance of SOTA methods for SIDD image denoising [2, 11, 50, 53, 61].

example the small aperture and limited dynamic range of smartphone cameras [63], or by inappropriate lighting conditions (*i.e.* images captured in low-light). To solve these problems, image restoration is usually understood as an ill-posed problem, in which, given the degraded image the algorithm needs to output a clean image.

To be embedded in-camera by a manufacturer, an image restoration algorithm should comply with strong requirements in terms of quality, robustness, computational complexity, and execution time. In general, digital cameras have a set of resources in which to allocate all the operations in the ISP pipeline [17]. Therefore, any new operation to be introduced in this pipeline should be of good enough quality to "pay" for the resources it will consume. Moreover, for an algorithm to be embedded in a camera, it is required to always improve over the input image, *i.e.* to be robust for any possible circumstance and input signal.

Image restoration is a traditional problem, and its study began as soon as we started to capture images, and many famous methods, such as Non-local Means for image denoising [8], are almost 20 years old. These traditional methods were usually defined by hand-crafted priors that narrowed the ill-posed nature of the problems by reducing the set of plausible solutions. However, since 2012 there has been a switch to deep learning based image restoration algorithms,

as these methods have proven to be very powerful to generalize priors from a large number of images.

Unfortunately, despite the great advances and performance, research on image restoration and enhancement using deep learning usually forgets the previous defined need for obtaining algorithms that have low computational complexity and execution time; and therefore many of them cannot be integrated into modern smartphones due to their complexity *i.e.* FLOPs or memory requirements.

In this paper, we therefore aim at defining a new image enhancement algorithm that achieves competitive results in comparison with state-of-the-art methods on different related tasks, yet, at the same time, presents a low-complexity and a competitive execution time in current off-the-shelf smartphones as proven by the use of the AIScore [31]. A first example of this behaviour is shown in Figure 1, where we compare our method to the current state-of-the-art in image denoising. As it can be seen, our methods is as close as 0.02 in SSIM to the state-of-the-art, while having at least ×50 less MACs. More details will appear later in Section 5.

In summary, **our contributions** are as follows:

- We propose a lightweight U-Net based architecture characterized by the inverted residual attention (IRA) block. Similar to contemporary works in this field, yet more efficient and smaller.

- We optimize our model in terms of used parameters and computational cost (*i.e.* FLOPs, MACs), thus being able to achieve real-time performance on current smartphone GPUs at FullHD input image resolution. This improvement is illustrated in figure 1.

- We propose a new type of analysis, from a production point of view, observing the behaviour of our model when deployed on commercial smartphones.

2. Related Work

Image restoration is split in a large number of subproblems, and in this paper we focus on four of the most popular in current research: image denoising, image deblurring, HDR image reconstruction from a single image, and Under-Display-Camera (UDC) image restoration.

Image denoising Image denoising has been a topic of research for more than 30 years. The most famous traditional image denoising methods are the non-local ones, such as Non-Local-Means [8] and BM3D [18]. More recently, multiple methods have studied different image representations to facilitate the denoising problem for this well-behaved algorithms [24, 51].

As in other image restoration problems, research on image denoising has moved towards deep learning models. The first remarkable work on denoising with deep learning is probably Zhang *et al.* [66] DnCNN, where they proposed to learn a CNN to estimate the noise distribution of the input image. Since then, plenty of other deep learning methods have appeared [4, 10, 28, 35, 54, 60, 61, 63, 64]. This has also been possible thanks to the appearance of challenges such as [1, 2] that facilitate a benchmark for compering methods together with training and testing images. For a deeper analysis we refer the reader to the survey in [49].

Image deblurring Image deblurring is a traditional problem in image restoration. Its main objective is to remove the blur that appears in the input image, this can be caused by different factors (*i.e.* camera shake, object motion, or lack of focus) and output a sharp image. As it is the case for image denoising, initially the different algorithms were based on hand-made priors or constraints, mostly treating image restoration as an inverse filtering problem [14, 23, 57], but these methods were surpassed with the appearance of deep learning models [15, 34, 40, 48, 50]. For a more in-depth analysis of deep learning methods applied to this problem, we point the reader to the survey in [65]

HDR reconstruction Digital cameras can only capture around two orders of magnitude in luminance, and have therefore a very limited Dynamic Range. This is far beyond the luminance differences that appear in the real world, and the reason why, when capturing images, we may end up with highlights in very bright regions or information loss in very dark ones.

The first works aiming at recovering the dynamic range of images were based on a set of multiple images from the same scene. The seminar work of Debevek and Malik [19] assumed that from a set of multiples images of the same scene, it is possible to recover a single Camera Response Function (CRF) and undo the camera process. This hypothesis was true for film cameras, but as recently proved by Gil Rodriguez *et al.* [44] this is not true for current digital cameras, as color channels are not independent and the camera modifies the non-linearity for different exposure values.

Currently, thanks to deep learning and different benchmark challenges [43], there has been a surge of methods aiming at recovering the full High Dynamic Range (HDR) from a single input image, a problem also named inverse tone mapping [5, 6]. This rising started with the work of Eilersten *et al.* [20], where they proposed a U-Net architecture for solving this problem. Other deep learning based methods for HDR single-image reconstruction are the ones in [36, 47]. Regarding deep learning methods for multiple images input, we should also mention the work in [9].

UDC image restoration Recently, a new imaging system, the Under-Display Camera (UDC) has appeared. The UDC system consists of a camera module placed underneath and closely attached to the semi-transparent Organic Light-Emitting Diode (OLED) display [71]. This solution provides an advantage when it comes to the user experience analysis, with the full-screen design providing a higher level of comfort. The disadvantage of this solution is that the OLED display acts as an obstacle for the light interacting with the camera sensor, inducing additional reflections, refractions and other effects connected to the Image Signal Processing (ISP) [17] model characterizing the camera.

Zhou *et al.* [72] and their 2020 ECCV challenge [71] were the first works that directly addressed this novel restoration problem using deep learning. Baidu Research team [71] proposed the Residual dense based on Shade-Correction for T-OLED UDC Image Restoration. In [72], the authors devised an MCIS to capture paired images, and solve the UDC image restoration problem as a blind deconvolution problem. More recently, Feng *et al.* [21] proposed one of the world's first production UDC device for data collection, experiments, and evaluations. They also proposed a new model called DISCNet [21], and provided a benchmark for multiple blind and non-blind methods on UDC datasets.

General problem formulation It is important to note that the UDC problem can be seen as a generalization of the other three listed above. Following the formulation proposed in [21], the UDC problem can be formulated as:

$$y = \gamma(C(x * k + n)), \tag{1}$$

where x represents the clean image with high dynamic range, k is the point spread function (PSF) (i.e. the blurring kernel), $*$ represents the 2D convolution operator, and n denotes the camera noise. Also, $C(\cdot)$ emulates the reduction of dynamic range, following $C(x) = \min(x, x_{max})$, where x_{max} is a range threshold, and $\gamma(\cdot)$ represents a Tone-Mapping Function.

From this formulation we can clearly see that when we suppose there is not noise ($n = 0$ at all pixels), x_{max} is large enough and we do not perform any Tone-Mapping, we end up with the traditional deblurring formulation $y = x * k$. Similarly, if we suppose that the kernel k is a Dirac Delta, x_{max} is equal to the maximum possible value of the input signal and we do not perform any Tone-Mapping we end up with the traditional image denoising formulation $y = x + n$. As we will be using in this paper the SIDD dataset [2], we will suppose that $n = \mathcal{N}(0, \beta)$, where $\beta^2(y) = \beta_1 y + \beta_2$ and β_1 represents the shot noise and β_2 represents the independent additive Gaussian noise. Finally, supposing that there is not noise ($n = 0$ at all pixels), and that the kernel k is a Dirac Delta, we end up with the HDR image reconstruction problem.

However, as explained in the introduction, none of the aforementioned methods have analyzed these problems from the efficiency point of view. The proposed models can generate high quality results but cannot be integrated into modern smartphones due to their complexity *i.e.* FLOPs.

3. Proposed Method

We propose a new model called *LPIENet*, following a U-Net [45] like architecture, standard in image restoration [21, 54, 61, 65, 72]. The main building blocks are inverted residual attention blocks, we refer to them as IRA blocks. These are selected due to their efficiency [30]. This is a blind image restoration method, and therefore, we do not rely on the PSF or other information about the camera sensor. This architecture is illustrated in Figure 2. The initial model consists on 5 blocks (3 encoders and 2 decoders) with $[16, 32, 64, 32, 16]$ channels respectively, and 0.13M parameters. To prove the scalability of our approach we design a deeper network by increasing the number of channels to $[32, 64, 128, 64, 32]$. We refer to this modification as *LPIENet-L*. Our base model represents a solution with $5\times$ less parameters than other methods [11, 21, 54, 61, 72]. Note that *LPIENet* was tailored specifically to be efficient when deployed on various smartphone devices. As this model is designed to perform denoising, deblurring and UDC image restoration in real-time conditions on mobile devices, we optimize it considering SSIM performance and a lower number of FLOPs. As we will prove in Section 5, this model achieves competitive performance while processing Full-HD and 2K resolution images in real-time on a wide range of commercial smartphones. Moreover, as we pointed out in Section 2, our methods combine ideas from **deblurring** [41] and **HDR** [9, 37] networks, and attention methods [56, 61]. We believe these tasks are extremely correlated with the UDC restoration problem.

3.1. Method Description

As we show in Figure 2, *LPIENet* consists of: Three encoder blocks (E_1, E_2, E_3) and two decoder blocks (D_1, D_2). Each encoder block consists of the following sequence, what we previously named an IRA block: two inverted linear residual blocks [30], an attention block [56] to enrich the feature representation, and finally, a downsampling layer (*i.e.* max-pooling). The decoder blocks D_1, D_2 follow the same structure. We replace the downsampling operations by a bilinear upsampling layer, and concatenate the skip-connections [45]. An important detail to improve efficiency is the fact that we upsample the features after activating them, not before. Since the inverted residual blocks [30] have a limited receptive field and representation, we further activate them using a combination of spatial and color attention [56].
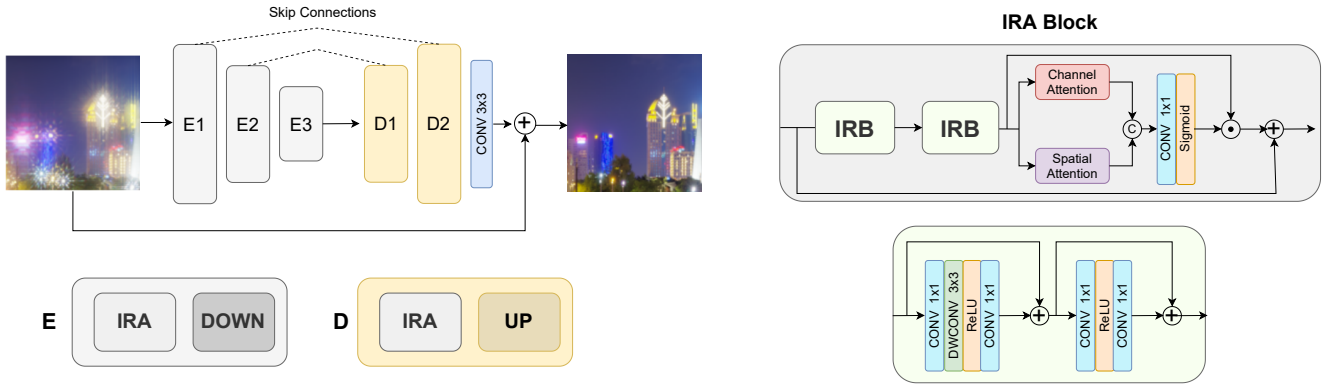
Figure 2. Architecture of the proposed *LPIENet* network and the IRA Block. Our model is designed considering current TFLite supported operations and mobile devices limitations. The attention mechanisms [11,56] have been optimized to require less memory and computation.

Contemporary work by Chen *et al.* has shown this is an efficient image restoration solution [11]. Other important considerations are: (i) we do not use Batch Normalization (BN) layers, which consume the same amount of GPU memory as convolutional layers, and they also increase computational complexity [70], (ii) `LayerNorm` and `GELU` have shown a consistent performance improvement in image restoration [11], however, due to the deployment limitations we do not use such techniques. (iii) We report MACs or FLOPs considering the following standard relationship between both terms: $MACs \approx 0.5 \times FLOPs$

4. Experimental Setup

4.1. Datasets

SIDD Medium [1, 2] is a real image denoising dataset providing 320 image pairs, two image pairs from each one of the 160 scenes. The dataset introduces great variability in terms of sensors, with five different smartphone camera sensors used for data acquisition, and images captured under different exposure and lighting conditions. A noise model estimation method is then used to produce the counterparts of the noise-affected images. The authors provide $\approx 80\%$ of the data for training and validation, with the remaining 20% not being publicly available. Methods are compared and tested using their on-line SIDD benchmark [1] and test set of 1280 images. We further processed the high-resolution training images (*i.e.* 3000×2000) to a set of non-overlapping image crops, extracting 12 crops of resolution 1000×1000 per each training image.

UDC SYNTH [21, 22]. The original RGB images from SYNTH [21] have resolution 800×800, we extracted non-overlapping patches of size 400×400. We extracted patches

(instead of downsampling the image) to preserve the high-frequency details. Moreover, images were transformed from the original domain to the tone-mapped domain using the following function $f(x) = x/(x + 0.25)$. Therefore the pixel intensities are in the range $[0, 1)$.

GoPro [40] This dataset by Nah *et al.* is widely used in motion deblurring, for training and evaluation. It consists of 3214 pairs of blurry and sharp HD resolution images (1280×720). This is a synthetic dataset since the blurred images are produced by averaging several high-speed clean images. Following the standard practice [40], we use 2103 pairs for training and 1111 pairs for testing. During training we use random paired crops of size 540×960.

4.2. Implementation details

To avoid artifacts, the output of our models is clipped into range $[0, 1 - p]$, where p is 10^{-5}. Our models were implemented in Tensorflow 2 and trained using a TPU v3.

We used Adam optimizer [33] with default hyper-parameters, and an initial learning rate of 2e-3. The learning rate is reduced by 50% during plateaus up to a minimum learning rate of 1e-6. We use basic augmentations: horizontal and vertical flip, and rotations. We set 4 as mini-batch size and trained to convergence for a few days (*i.e.* 500 epochs). We found very profitable to train on large resolution, we start training with random crops of size 400×400 en eventually increase the image size up to HD resolution.

Loss function The model is trained using a weighted sum of a $\mathcal{L}1$ loss, a SSIM loss and a Gradient loss [38]. The final loss function is:

$$\mathcal{L} = \alpha \mathcal{L}_{SSIM} + \mathcal{L}_1 + \beta \mathcal{L}_{Grad} \tag{2}$$

where α and β are set empirically to scale the loss.

Table 1. Quantitative Results for Under-display Camera (UDC) Image Restoration using SYNTH [21]. We show fidelity and perceptual metrics. (*) indicates methods proposed at the MIPI UDC Challenge [22]. We consult some numbers from [21,22]. We highlight in bold the best blind and non-blind methods.

Method	PSNR (dB) ↑	SSIM ↑	LPIPS ↓	# Params (M)	Uses PSF
Wiener Filter (WF) [42]	27.30	0.83	0.330	-	✓
SRMDNF [21]	34.80	0.96	0.036	1.5	✓
SFTMD [21]	42.35	0.98	0.012	3.9	✓
DISCNet (PSF) [21]	**42.77**	**0.98**	**0.012**	3.8	✓
RDUNet [45, 70]	34.37	0.95	0.040	8.1	✗
DE-UNet [72]	38.11	0.97	0.021	9.0	✗
DISCNet (w/o PSF) [21]	38.55	0.97	0.030	2.0	✗
RushRushRush *	39.52	0.98	0.021	n/a	✗
eye3 *	36.69	0.97	0.032	n/a	✗
FMS Lab *	35.77	0.97	0.045	n/a	✗
EDLC2004 *	35.50	0.96	0.045	n/a	✗
SAU_LCFC *	32.75	0.96	0.056	n/a	✗
LPIENet-L (ours)	**40.12**	**0.98**	0.020	0.6	✗
LPIENet (ours)	34.10	0.95	0.031	0.1	✗

4.3. Experimental Results

4.3.1 Quantitative Results

UDC image restoration We evaluate our models on the SYNTH dataset [21], also used in the "UDC MIPI 2022 Challenge" [22], and compare them against current state-of-the-art methods (excluding contemporary methods and methods that have not been officially published). Note that these results are not fully reproducible since multiple approaches do not provide open-sourced code. Moreover, the SYNTH dataset [21] test set (and PSF) is public, which makes difficult a fair comparison with methods that might tend to overfit it. As we show in Table 1, we achieve competitive results for this problem, while using models with fewer parameters than the other methods. As an example, we can look at the comparison against DiscNet [21] base blind version, as we used the same initial setup to build up our model. We can see how our method outperforms this one by almost 2 dBs. We extend this analysis and prove the benefits of our method in Section 5.

Real image denoising In Table 2 we present our results for the SIDD [2]. As we can see our method is quite competitive, getting as close as 0.02 in SSIM to the state-of-the-art. We should recap here the analysis shown in Figure 1, were we can see that our methods requires 100x less MACs than Restormer [61] and MAXIM [50] (we will see the difference in MACs in the next section). This allows our method to be deployed in current smartphones (as we will show next in Section 5), while Restormer is far from that capability. Please remind that this was the goal of our paper: to obtain a lightweight model able to compete with state-of-the-art, but with reduced runtime and complexity, so it is possible to embed the model in current smartphones. We provide our architecture ablation study in Table 3. The attention mechanisms [11] help to improve the performance of the efficient inverted residual blocks.

Table 2. Quantitative sRGB Denoising results on the SIDD [2]. Our model outperforms classical famous methods and can denoise images in real-time in smartphones. Numbers consulted at [2, 11].

Method	PSNR (dB) ↑	SSIM ↑
Noisy	23.70	0.480
BM3D [18]	25.65	0.685
NLM [8]	26.75	0.699
KSVD [3]	26.88	0.842
FoE [46]	25.58	0.792
WNNM [27]	25.78	0.809
TNRD [13]	24.73	0.643
EPLL [73]	27.11	0.870
DnCNN [66]	30.71	0.695
CBDNet [29]	33.28	0.868
CycleISP [62]	39.52	0.957
DAGL [39]	38.94	0.953
Mirnet [63]	39.73	0.959
Restormer [61]	40.02	0.960
LPIENet (ours)	37.47	0.940
LPIENet (ensemble)	37.73	0.943

Method	SSIM ↑	GMACs ↓
Baseline UNet [21, 45]	0.855	5.8
Res Block → Dual Inv Res Block [30]	0.840	1.0
Dual Inv Block → IRA Block (Ours)	0.940	1.3
(Ours) LPIENet + kernel size = 5	0.941	1.4

Table 3. Ablation study for Real Image Denoising. SSIM reported on SIDD [2]. GMACs calculated using a 256×256 RGB input.

4.3.2 Qualitative Results

Real image denoising Example results for the real image denoising case are shown in Figure 3. In this Figure we can clearly see that our results are competing again the state-of-the-art at a fraction of computational cost. We recommend the reader to focus on the words "Forward" and "Park" in the top image, and in the word "screen" in the bottom image. Once again please remember that our model has **at least 50x less MACs** than any of the methods outrunning it. In Section 5, we deploy our method and CycleISP [62] in commercial smartphones, and compare runtimes.

HDR image reconstruction Example results for the case of HDR image reconstruction are shown in Figure 5. We can see in this Figure how our method is able to better reconstruct the textures, colors and geometric properties of the hallucinated objects, in comparison versus the other ones, that are state-of-the-art. More in detail, we want the reader to focus on the leaves of the plant in the top image, and the light specularities in the bottom one. Please also note that these images are from the SYNTH dataset [21], because, as we explained in the related work, UDC image restoration generalises the problem of HDR reconstruction.

Table 4. Description of the selected commercial smartphone devices.

Phone Model	Launch	Chipset	CPU	GPU	RAM (GB)	AI Score ↑
(# 1) Samsung A50	03/2019	Exynos 9610	8 cores	Mali-G71 GP3	4	45.4
(# 2) OnePlus Nord 2 5G	07/2021	MediaTek Dimensity 1200	8 cores	Mali-G77 MC9	8	194.3
(# 3) OnePlus 8 Pro	04/2020	Qualcomm Snapdragon 865 5G	8 cores	Adreno 650	12	137.0
(# 4) Realme 8 Pro	03/2021	Qualcomm Snapdragon 720G	8 cores	Adreno 618	8	60.6

Image deblurring We process images from the GoPro dataset [40] test split. We improve the base blurry inputs by 2dBs using our method (*i.e.* average PSNR of 27.8 dB in the test set). We also outperform baseline methods Xu *et al.* [58], Hyun *et al.* [32], Whyte *et al.* [55] and Gong *et al.* [25]. We provide visual results in Figure 6 and in the supplementary material.

We also acknowledge other state-of-the-art methods such as NAFNet [11], UFormer [53], Restormer [61], MAXIM [50] and MPRNet [64], tested on the same deblurring and denoising setup, however, we do not compare with them as many are contemporary and more complex models that achieve the most competitive performance at the cost of being extremely inefficient. We analyze this in Table 5.

5. Efficiency Analysis

We evaluate the performance of different methods using four smartphone devices. In Table 4, we provide all the details regarding the devices used in our benchmark. As a reference metric to measure the performance of each mobile device in the scenario of the deep learning models deployment, we provide the AI Score [31]. This AI score is computed over a set of experiments (*i.e.* category recognition, semantic segmentation,or super-resolution), where inference precision and hardware/software behaviour are observed and quantified into a score metric depending on the importance of each of the factors. All the tests are described by the authors in [31], and are available for the Android smartphones through the application *AI Benchmark* [31], which also provides support for deep learning models deployment on different hardware options (CPU, GPU, NPU), and with various precision types (*i.e.* FP16, FP32, INT8). We refer to the work [31] for more details about this app.

Deployment Limitations. Before introducing our benchmark setup and compared methods, we must note the limitations of this efficiency analysis: (i) models need to be converted to .tflite format, this conversion is not trivial and might lead to loss of performance, especially if the model was originally implemented using PyTorch (*i.e.* PT → ONNX → TF → TFLite) . (ii) Multiple state-of-the-art operations such as GELU activation or LayerNorm [11] are not currently supported. For these reasons, multiple methods cannot be currently deployed using the standard setup [31] or require a Tensorflow implementation.

We adapt our models to follow this standard [31] and re-implement (if possible) other methods.

Benchmark setup. Models are tested on CPU and GPU (TFLite Delegate) FP16 after default Tensorflow-lite conversion and optimization, using as input a tensor image of different resolutions. The FLOPs are calculated for each resolution. We show in Figure 4 the evaluation process using *AI Benchmark app* [31].

Methods comparison. DiscNet [21] cannot be directly converted to the required TFlite format due to unsupported operations. For this reason, we use DiscNet "baseline (a)" defined at [21] to evaluate the model, this provides a fair lower-bound of the complete method's performance.

SRMDNF [72] and SFTMD [72] cannot be directly converted either, therefore, we use a canonical UNet model with the same number of FLOPs to approximate their performance. As we show in Table 6, our proposed model *LPIENet* can process Full-HD images in real-time in commercial mobile devices GPUs, being ×5 faster than DiscNet [21]. Our model can also process low-resolution images on CPU in real-time (under 1 second per image), and its performance only decays 0.04 with respect to DiscNet [21] in terms of SSIM [52].

Table 5. Comparison between **LPIENet** and other state-of-the-art methods, in terms of perceptual performance (SSIM) and necessary mult-adds (MACs). "Task" indicates the task the compared solutions were proposed for: denoising (1), deblurring (2), HDR imaging (3), and UDC camera restoratiron (4). "Res" specifies the input resolution used to report the number of GMACs. The last column specifies their advantage in terms of perceptual performance (SSIM) over our model.

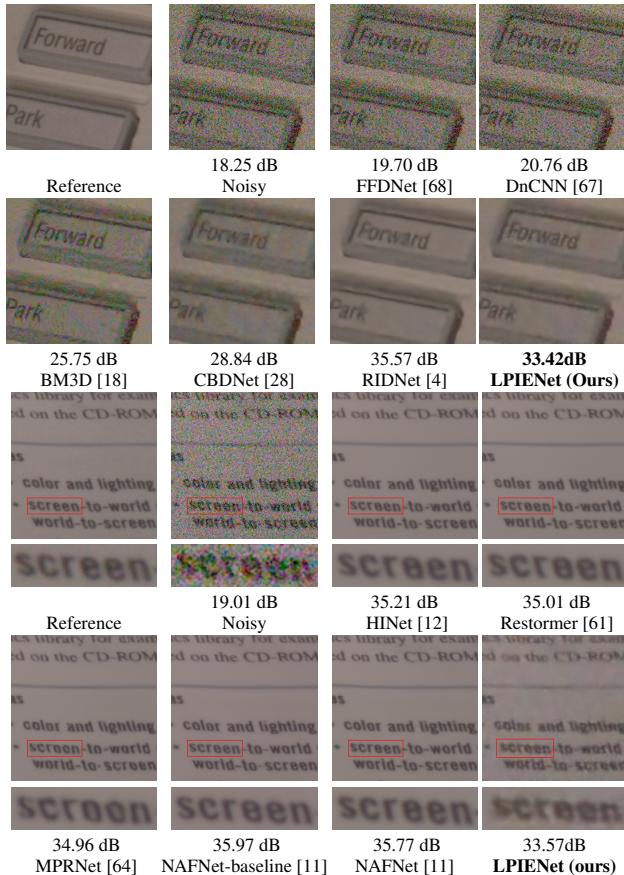| Method	Task	↓MACs (G)	Res.	SSIM Diff.
MPRNet [64]	1/2	588	256	0.15
SRMDNF [69]	4	475.5	800	0.02
DISCNet [21]	4	221	800	0.01
HINet [12]	1/2	170.7	256	0.15
MAXIM [50]	1/2	169.5	256	0.17
Restormer [61]	1/2	140	256	0.17
UFormer [53]	1/2	89.5	256	0.17
DE-UNet [72]	4	84.5	800	-0.01
HDR-NTIRE [43]	3	≤66	FHD	-
NAFNet [11]	1/2	65	256	0.24
LPIENet	1/2/3/4	**12**	800	-
LPIENet	1/2/3/4	**1.3**	256	-

Figure 3. Real image denoising results of different methods on a challenging sRGB image from the SIDD dataset [2]. Our method can recover details and produce pleasant results while being $10\times$ smaller than other methods.

In Figure 5 we provide qualitative samples in challenging scenarios, our model is able to improve notably the perceptual quality of the UDC degraded images in real-time. Note that in the perception-distortion tradeoff [7], we consider more important perceptual metrics [16, 26], as we aim for pleasant results for humans.

Limitations. Besides the mentioned implementation limitations that are limiting performance (*i.e.* `LayerNorm` is not supported, and was proven to be essential in image restoration [11]), we find the following limitations in our method: (i) the use of pixel-wise convolutions an inverted residual blocks limits the receptive field, this is a clear drawback when solving HDR and deblurring problems (*i.e.* the model cannot hallucinate and generate realistic content as more complex methods), (ii) the self-imposed limitation of operations is an evident performance limitation.
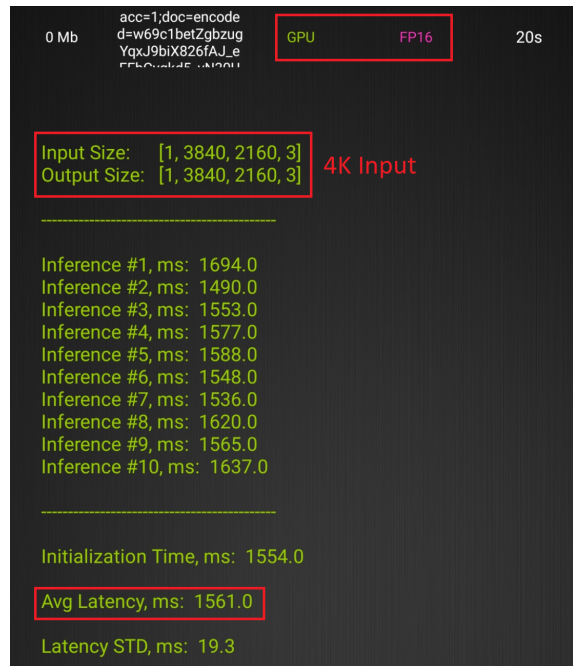

Figure 4. AI Benchmark [31] model performance evaluation. Our method can process 4K inputs in 1.5 seconds using GPU FP16.

6. Conclusions

In this paper, we have presented a new lightweight model for perceptual image enhancement, called LPIENet. Our experiments and tests prove its ability to compete against the state-of-the-art in different image restoration and enhancement tasks, namely image denoising, image deblurring, HDR reconstruction and UDC image restoration. Our new model consists of only 0.1M parameters and is able to process high resolution images in current off-the-shelf phones in less than a second.

The main novelty of our new model is on the combination of a U-Net like architecture with inverted residual attention (IRA) blocks that allows to drastically reduce the number of operations required. We are able to challenge well-established models with a fraction of their reported number of parameters or number of performed operations. Finally, we introduce a novel benchmark for image enhancement based on efficiency and deployment capabilities on real smartphones. We refer the reader to our project page for supplementary material.

Table 6. Efficiency benchmark for UDC Image Restoration on different commercial smartphones. We show the performance of our method in terms of runtime at different image resolutions, device architectures and running scenarios (CPU, GPU). Runtimes are the average of at least 5 iterations. Our method is the one of the few methods described in this work that can process high-resolution images, even 2K, without tiling or patching the input image, while we achieving high perceptual quality results. Note that the methods marked with ✗ failed the test due to memory requirements or runtime constraints. FHD is considered 1920×1080.

| Method name | FLOPs ↓ (G) | Resolution (px) | SSIM ↑ | Runtime (s) | | | | | | | |
				Phone #1		Phone #2		Phone #3		Phone #4	
				CPU	GPU	CPU	GPU	CPU	GPU	CPU	GPU
SFTMD [72]	2460	800×800	0.986	✗	✗	✗	✗	✗	✗	✗	✗
RDUNet [59]	2461	FHD		✗	21.2	437.8	2.7	✗	5.1	764.5	✗
RDUNet [59]	759	800×800	0.970	261.1	6.2	65.4	0.86	137.1	1.50	113.9	4.49
RDUNet [59]	78	256×256		1.4	0.6	0.39	0.09	0.58	0.18	0.64	0.52
CycleISP [62]	12410	FHD		✗	98.7	✗	20.51	✗	✗	✗	✗
CycleISP [62]	3830	800×800	0.957	✗	29.4	844.9	5.9	✗	10.17	1264	✗
CycleISP [62]	392	256×256		14.6	3.0	5.8	0.64	8.2	1.0	7.54	3.2
DiscNet [21]	1434	FHD		1055	13.7	261.5	2.0	61.1	5.3	455.2	✗
DiscNet [21]	442	800×800	0.974	52.4	4.0	13.5	0.63	15.6	1.4	23.3	3.98
DiscNet [21]	256	256×256		1.1	0.41	0.3	0.07	0.6	0.14	0.58	0.38
LPIENet	310.4	4K		✗	6.9	368.1	1.5	73.0	1.6	✗	✗
LPIENet	82.8	2K		24.8	1.8	6.9	0.43	13.1	0.4	12.40	1.2
LPIENet	77.6	FHD	0.940	22.0	1.7	6.3	0.41	12.0	0.4	11.57	1.15
LPIENet	23.95	800×800		3.8	0.52	1.5	0.13	2.2	0.13	2.4	0.372
LPIENet	2.45	256×256		0.38	0.065	0.14	0.031	0.23	0.015	0.26	0.044
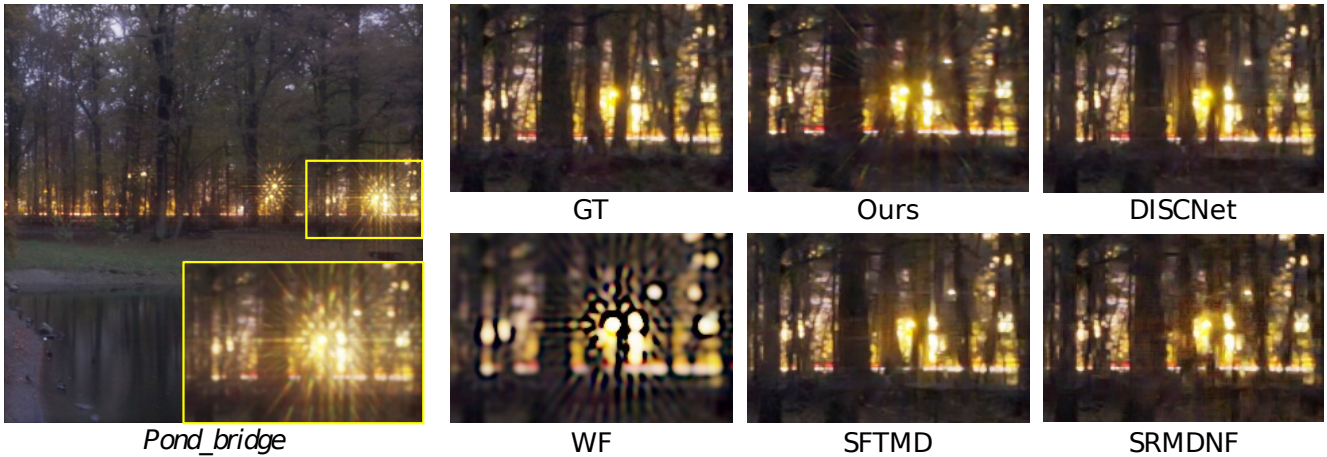
Figure 5. Visual comparison on SYNTH [21] synthetic validation images. Our method *LPIENet-L* recovers fine details, and produces high perceptual quality images without unpleasant artifacts. Image selection from DiscNet [21].
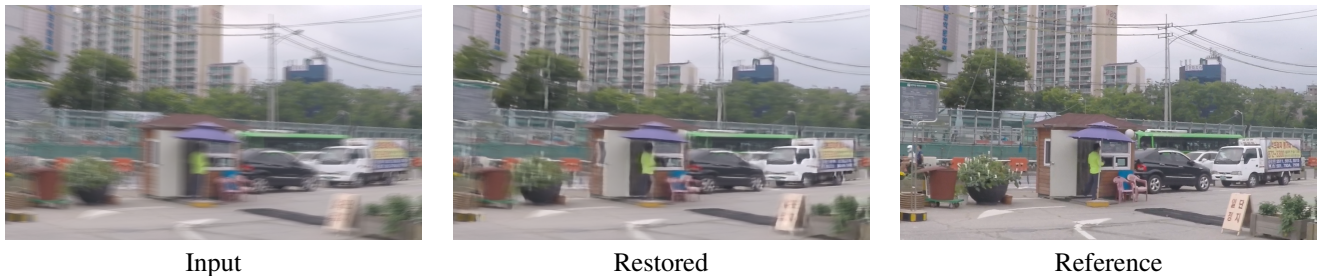

Input   Restored   Reference

Figure 6. Qualitative results on the GoPro dataset [40] test split. Our model can also reduce notably motion blur.

References

[1] Abdelrahman Abdelhamed, Mahmoud Afifi, Radu Timofte, and Michael S Brown. Ntire 2020 challenge on real image denoising: Dataset, methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 496–497, 2020.

[2] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1692–1700, 2018.

[3] M. Aharon, M. Elad, and A. Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.

[4] Saeed Anwar and Nick Barnes. Real image denoising with feature attention. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3155–3164, 2019.

[5] Francesco Banterle, Kurt Debattista, Alessandro Artusi, Sumanta Pattanaik, Karol Myszkowski, Patrick Ledda, and Alan Chalmers. High dynamic range imaging and low dynamic range expansion for generating hdr content. In *Computer graphics forum*, volume 28, pages 2343–2367. Wiley Online Library, 2009.

[6] Francesco Banterle, Patrick Ledda, Kurt Debattista, and Alan Chalmers. Inverse tone mapping. In *Proceedings of the 4th international conference on Computer graphics and interactive techniques in Australasia and Southeast Asia*, pages 349–356, 2006.

[7] Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6228–6237, 2018.

[8] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 2, pages 60–65. Ieee, 2005.

[9] Sibi Catley-Chandar, Thomas Tanay, Lucas Vandroux, Aleš Leonardis, Gregory Slabaugh, and Eduardo Pérez-Pellitero. Flexhdr: Modelling alignment and exposure uncertainties for flexible hdr imaging. *arXiv preprint arXiv:2201.02625*, 2022.

[10] Meng Chang, Qi Li, Huajun Feng, and Zhihai Xu. Spatial-adaptive network for single image denoising. In *European Conference on Computer Vision*, pages 171–187. Springer, 2020.

[11] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. *arXiv preprint arXiv:2204.04676*, 2022.

[12] Liangyu Chen, Xin Lu, Jie Zhang, Xiaojie Chu, and Chengpeng Chen. Hinet: Half instance normalization network for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 182–192, June 2021.

[13] Y. Chen and T. Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1256–1272, 2017.

[14] Sunghyun Cho and Seungyong Lee. Fast motion deblurring. In *ACM SIGGRAPH Asia 2009*, pages 1–8. 2009.

[15] Sung-Jin Cho, Seo-Won Ji, Jun-Pyo Hong, Seung-Won Jung, and Sung-Jea Ko. Rethinking coarse-to-fine approach in single image deblurring. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4641–4650, 2021.

[16] Marcos V Conde, Maxime Burchi, and Radu Timofte. Conformer and blind noisy students for improved image quality assessment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 940–950, 2022.

[17] Marcos V. Conde, Steven McDonagh, Matteo Maggioni, Ales Leonardis, and Eduardo Pérez-Pellitero. Model-based image signal processors via learnable dictionaries. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(1):481–489, Jun. 2022.

[18] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007.

[19] Paul E Debevec and Jitendra Malik. Recovering high dynamic range radiance maps from photographs. In *ACM SIGGRAPH 2008*, pages 1–10. 2008.

[20] Gabriel Eilertsen, Joel Kronander, Gyorgy Denes, Rafał K. Mantiuk, and Jonas Unger. Hdr image reconstruction from a single exposure using deep cnns. *ACM Trans. Graph.*, 36(6), nov 2017.

[21] Ruicheng Feng, Chongyi Li, Huaijin Chen, Shuai Li, Chen Change Loy, and Jinwei Gu. Removing diffraction image artifacts in under-display camera via dynamic skip connection network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 662–671, June 2021.

[22] Ruicheng Feng, Chongyi Li, Shangchen Zhou, Wenxiu Sun, Qingpeng Zhu, Jun Jiang, Qingyu Yang, Chen Change Loy, and Jinwei Gu. Mipi 2022 challenge on under-display camera image restoration: Methods and results. *arXiv preprint arXiv:2209.07052*, 2022.

[23] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T Roweis, and William T Freeman. Removing camera shake from a single photograph. In *Acm Siggraph 2006*, pages 787–794. 2006.

[24] Gabriela Ghimpeteanu, Thomas Batard, Marcelo Bertalmío, and Stacey Levine. A decomposition framework for image denoising algorithms. *IEEE Trans. Image Process.*, 25(1):388–399, 2016.

[25] Dong Gong, Jie Yang, Lingqiao Liu, Yanning Zhang, Ian D. Reid, Chunhua Shen, Anton van den Hengel, and Qinfeng Shi. From motion blur to motion flow: A deep learning solution for removing heterogeneous motion blur. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3806–3815, 2017.

[26] Jinjin Gu, Haoming Cai, Chao Dong, Jimmy S Ren, Radu Timofte, Yuan Gong, Shanshan Lao, Shuwei Shi, Jiahao

Wang, Sidi Yang, et al. Ntire 2022 challenge on perceptual image quality assessment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 951–967, 2022.

[27] S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[28] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1712–1722, 2019.

[29] Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, and Lei Zhang. Toward convolutional blind denoising of real photographs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1712–1722, 2019.

[30] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[31] Andrey D. Ignatov, Radu Timofte, William Chou, Ke Wang, Max Wu, Tim Hartley, and Luc Van Gool. Ai benchmark: Running deep neural networks on android smartphones. In *ECCV Workshops*, 2018.

[32] Tae Hyun Kim, Byeongjoo Ahn, and Kyoung Mu Lee. Dynamic scene deblurring. In *2013 IEEE International Conference on Computer Vision*, pages 3160–3167, 2013.

[33] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

[34] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8878–8887, 2019.

[35] Yang Liu, Zhenyue Qin, Saeed Anwar, Pan Ji, Dongwoo Kim, Sabrina Caldwell, and Tom Gedeon. Invertible denoising network: A light solution for real noise removal. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13365–13374, 2021.

[36] Yu-Lun Liu, Wei-Sheng Lai, Yu-Sheng Chen, Yi-Lung Kao, Ming-Hsuan Yang, Yung-Yu Chuang, and Jia-Bin Huang. Single-image hdr reconstruction by learning to reverse the camera pipeline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1651–1660, 2020.

[37] Yu-Lun Liu, Wei-Sheng Lai, Yu-Sheng Chen, Yi-Lung Kao, Ming-Hsuan Yang, Yung-Yu Chuang, and Jia-Bin Huang. Single-image hdr reconstruction by learning to reverse the camera pipeline. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[38] Cheng Ma, Yongming Rao, Yean Cheng, Ce Chen, Jiwen Lu, and Jie Zhou. Structure-preserving super resolution with gradient guidance. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7769–7778, 2020.

[39] Chong Mou, Jian Zhang, and Zhuoyuan Wu. Dynamic attentive graph learning for image restoration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4328–4337, 2021.

[40] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3883–3891, 2017.

[41] Seungjun Nah, Sanghyun Son, Suyoung Lee, Radu Timofte, and Kyoung Mu Lee. Ntire 2021 challenge on image deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 149–165, 2021.

[42] François Orieux, Jean-François Giovannelli, and Thomas Rodet. Bayesian estimation of regularization and point spread function parameters for wiener–hunt deconvolution. *JOSA A*, 27(7):1593–1607, 2010.

[43] Eduardo Pérez-Pellitero, Sibi Catley-Chandar, Richard Shaw, Aleš Leonardis, Radu Timofte, Zexin Zhang, Cen Liu, Yunbo Peng, Yue Lin, Gaocheng Yu, et al. Ntire 2022 challenge on high dynamic range imaging: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1009–1023, 2022.

[44] Raquel Gil Rodríguez, Javier Vazquez-Corral, and Marcelo Bertalmío. Issues with common assumptions about the camera pipeline and their impact in HDR imaging from multiple exposures. *SIAM J. Imaging Sci.*, 12(4):1627–1642, 2019.

[45] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[46] S. Roth and M. J. Black. Fields of experts: a framework for learning image priors. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 860–867 vol. 2, 2005.

[47] Marcel Santana Santos, Tsang Ing Ren, and Nima Khademi Kalantari. Single image hdr reconstruction using a cnn with masked features and perceptual loss. *ACM Trans. Graph.*, 39(4), jul 2020.

[48] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8174–8182, 2018.

[49] Chunwei Tian, Lunke Fei, Wenxian Zheng, Yong Xu, Wangmeng Zuo, and Chia-Wen Lin. Deep learning on image denoising: An overview. *Neural Networks*, 131:251–275, 2020.

[50] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxim: Multi-axis mlp for image processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5769–5780, 2022.

[51] Javier Vazquez-Corral and Marcelo Bertalmío. Angular-based preprocessing for image denoising. *IEEE Signal Process. Lett.*, 25(2):219–223, 2018.

[52] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[53] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17683–17693, June 2022.

[54] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17683–17693, 2022.

[55] Oliver Whyte, Josef Sivic, Andrew Zisserman, and Jean Ponce. Non-uniform deblurring for shaken images. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 491–498, 2010.

[56] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[57] Li Xu and Jiaya Jia. Two-phase kernel estimation for robust motion deblurring. In *European conference on computer vision*, pages 157–170. Springer, 2010.

[58] Li Xu, Shicheng Zheng, and Jiaya Jia. Unnatural l0 sparse representation for natural image deblurring.

[59] Qirui Yang, Yihao Liu, Jigang Tang, and Tao Ku. Residual and dense unet for under-display camera restoration. In Adrien Bartoli and Andrea Fusiello, editors, *Computer Vision – ECCV 2020 Workshops*, pages 398–408, Cham, 2020. Springer International Publishing.

[60] Zongsheng Yue, Hongwei Yong, Qian Zhao, Deyu Meng, and Lei Zhang. Variational denoising network: Toward blind noise modeling and removal. *Advances in neural information processing systems*, 32, 2019.

[61] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5728–5739, 2022.

[62] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Cycleisp: Real image restoration via improved data synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[63] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Learning enriched features for real image restoration and enhancement. In *European Conference on Computer Vision*, pages 492–511. Springer, 2020.

[64] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14821–14831, 2021.

[65] Kaihao Zhang, Wenqi Ren, Wenhan Luo, Wei-Sheng Lai, Björn Stenger, Ming-Hsuan Yang, and Hongdong Li. Deep image deblurring: A survey. *International Journal of Computer Vision*, pages 1–28, 2022.

[66] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *Trans. Img. Proc.*, 26(7):3142–3155, jul 2017.

[67] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

[68] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018.

[69] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 6, 2018.

[70] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2472–2481, 2018.

[71] Yuqian Zhou, Michael Kwan, Kyle Tolentino, Neil Emerton, Sehoon Lim, Tim Large, Lijiang Fu, Zhihong Pan, Baopu Li, Qirui Yang, et al. Udc 2020 challenge on image restoration of under-display camera: Methods and results. In *European Conference on Computer Vision*, pages 337–351. Springer, 2020.

[72] Yuqian Zhou, David Ren, Neil Emerton, Sehoon Lim, and Timothy Large. Image restoration for under-display camera. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9175–9184, 2021.

[73] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *2011 International Conference on Computer Vision*, pages 479–486, 2011.