Fast and Efficient Restoration of Extremely Dark Light Fields

Mohit Lamba
Indian Institute of Technology Madras
Department of Electrical Engineering
ee18d009@smail.iitm.ac.in

Kaushik Mitra
Indian Institute of Technology Madras
Department of Electrical Engineering
kmitra@ee.iitm.ac.in

Abstract

The ability of Light Field (LF) cameras to capture the 3D geometry of a scene in a single photographic exposure has become central to several applications ranging from passive depth estimation to post-capture refocusing and view synthesis. But these LF applications break down in extreme low-light conditions due to excessive noise and poor image photometry. Existing low-light restoration techniques are inappropriate because they either do not leverage LF's multi-view perspective or have enormous time and memory complexity. We propose a three-stage network that is simultaneously fast and accurate for real world applications. Our accuracy comes from the fact that our three stage architecture utilizes global, local and view-specific information present in low-light LFs and fuse them using an RNN inspired feedforward network. We are fast because we restore multiple views simultaneously and so require less number of forward passes. Besides these advantages, our network is flexible enough to restore a $m \times m$ LF during inference even if trained for a smaller $n \times n$ ($n < m$) LF without any finetuning. Extensive experiments on real low-light LF demonstrate that compared to the current state-of-the-art, our model can achieve up to $1\ dB$ higher restoration PSNR, with $9\times$ speedup, 23% smaller model size and about $5\times$ lower floating-point operations.

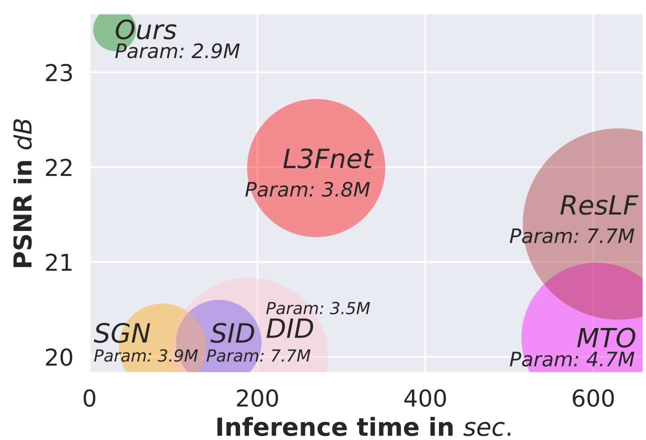

Figure 1. Inference speed and restoration quality comparison for different methods computed over the L3F-100 dataset [26]. Marker size is proportional to number of floating point operations. The runtime is for restoring a 9×9 dark LF with each SAI having 432×624 spatial resolution on a CPU.

1. Introduction

Unlike conventional cameras, a lenslet-based Light Field (LF) camera [1, 37, 30, 12] captures multiple views, called Sub-Aperture-Images (SAIs), of a scene in a single exposure. This implicit method of capturing a scene's 3D structure has enabled a wide range of applications such as post-capture refocusing & aperture control [37, 36], depth estimation [20, 52, 48], structure from motion [38] , augmented reality [18], and autonomous driving [5]. However, in low light, such as night-time, the SAIs are heavily corrupted by photon noise and contain inadequate color information. This prohibits performing any feature correspondences or satisfying photometric constraints across SAIs, rendering the captured LF useless for downstream applications. Thus it is crucial to design a method for low-light LF restoration.

The existing low-light enhancement techniques are mostly designed for single-frame images [3, 4, 13, 14] that do not utilize the rich LF information. Consequently, their restorations tend to be blurry or noisy. Very recently, a few LF based methods such as L3Fnet [26] and MTO [58] have been proposed to alleviate this problem to a reasonable extent. However, as shown in Fig.1, they have an enormous time-memory complexity that is prohibitive for a real world deployment. For example, even on a high-end CPU the existing LF based methods take $5-10$ minutes to restore a single 9×9 LF. Our goal [1] is to design a much faster and memory efficient solution with possibly better restoration quality.

To better capture complementary information present in different LF views our model, as shown in Fig. 2, consists

[1]This work was supported in part by *IITM Pravartak Technologies Foundation*.

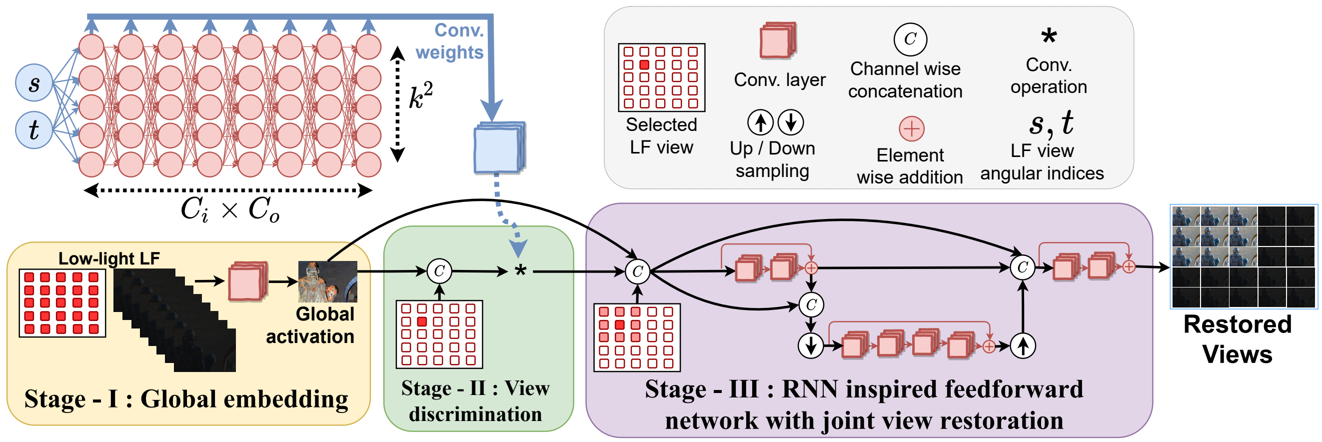

Figure 2. Our three-stage network for restoring light fields captured in extreme low-light conditions.

of three stages: Stage-I looks at all the views to compute a global embedding, Stage-II output is view-specific and Stage-III gives exclusive attention to the local neighborhood of LF views. Finally, our RNN inspired feedforward network uses all three complementary information to restore LF views.

In Stage-III, we use a RNN inspired feedforward network and not a standard U-net [42]. The immensely successful U-net architecture was originally designed for single-frame images like those obtained from conventional DSLR and smartphone cameras. But we observed that it lacks the expressiveness to capture long-range dependencies between various LF views, especially if the number of views is large. In contrast, RNNs are specially designed for long-range sequence modeling and past methods such as LFNet [53] used RNNs for super-resolving LF views. However, compared to feedforward architectures such as U-nets, RNNs have lower inference speed and more susceptible to vanishing gradients. To thus obtain better restoration quality and faster inference, we analyze the multi-scale processing of a U-net architecture in a RNN framework and, consequently, develop a novel feedforward network by unfolding RNNs in time.

Stage-I and Stage-III of our network share the weights across all SAIs and thus lose the sense of discriminating between SAIs while processing a SAI and its neighborhood. However, differentiating between SAIs is very important because each SAI is captured from a different portion of the main camera lens, leading to different characteristics and distortions across SAIs. We thus introduce Stage-II in between Stage-I and Stage-III, which uses separate network weights for different SAIs. Like ResLF [59], we could have used different weights for each SAI throughout the network but at the expense of having a prohibitively huge model size, inappropriate for real-world deployment.

Another interesting feature of Stage-II is that instead of directly learning the weights for each SAI during training, it instead learns the parameters for a series of fully connected layers, which can estimate the convolutional weights for different SAI based on their angular coordinates (s, t). This allows the network to restore a $m \times m$ LF during inference, even if it is trained for a smaller $n \times n$, $(n < m)$ LF.

Almost all neural network based LF methods require n^2 forward passes to process a $n \times n$ LF, which is the main reason for extremely slow inference. Recognizing this fact, we restore multiple SAIs in a single forward pass to substantially reduce the total number of forward passes required to restore low-light LF.

Our contributions: 1) We use a three-stage architecture to utilize global, local and view specific information present in low-light LF and fuse them using a novel RNN inspired feedforward network for superior restoration. 2) Our model can restore a $m \times m$ LF, even if trained for a smaller $n \times n$ LF $(n < m)$. This is because Stage-II of our network estimates convolutional weights for different SAIs during inference and does not freeze the weights at the end of training. 3) Instead of restoring LF views in steps, we restore multiple views in a single forward pass for faster inference. 4) Compared to state-of-the-art, our model achieves 1 dB higher restoration PSNR, with $9\times$ speedup, 23% smaller model size and at least $5\times$ lower floating-point operations, as shown in Fig. 1.

2. Related work

LF based methods: Numerous methods have been proposed for LF images addressing various concerns such as increasing the spatial resolution [21, 54, 53, 59, 55], denoising [35, 2, 8, 44, 9], depth estimation [20, 52, 48] and saliency detection [51, 56]. But all these methods mainly consider good lighting conditions. However, under extreme low-light conditions, the signal is heavily corrupted by photon noise with almost no color information. Single-frame denoising methods such as SGN [13] and BM3D [7] can be used to denoise LF views individually but this does not guarantee epipoles preservation. Consequently, few methods have been proposed for LF denoising. Mitra and Veeraraghavan [35] using the disparity cues modeled 4D LF patches using Gaussian Mixture Models (GMM) and provided a combined algorithm for LF super-resolution and denoising. Dansereau *et al*. [8] used frequency domain passband filtering for LF denoising and LFBM5D [2] extended the single-frame denoising BM3D algorithm for 4D LF images. Nevertheless, these methods cannot address the color restoration aspect of low light restoration. Secondly, the noise level found in extreme low-light is much greater than what these methods were designed. To address these concerns, recently, L3Fnet [26] and MTO [58] were proposed. L3Fnet used a two-stage deep-learning architecture for restoring low light LFs. In stage-I, L3Fnet extracted the overall 4D LF geometry and in stage-II this information was used to restore each LF view. L3Fnet also released a publicly available low-light LF dataset for training and benchmarking. MTO was, however, only tested for synthetic low-light images.

Single-frame low-light methods: Single-frame methods have witnessed a lot of progress towards low-light enhancement. The earliest approaches relied on modifying the image's histogram [23, 40, 47, 6, 19, 29] to increase its dynamic range. Later approaches, however, used the retinex theory [27, 28] to decompose the low-light image into illumination and reflectance components and use them for low-light enhancement [50, 10, 14, 11, 31, 4, 57, 49]. All these methods have considered weakly illuminated scenes and not night-time extreme low-light conditions. SID [3], a landmark paper on low-light enhancement, used a U-net architecture for extreme low-light single-frame restoration. Since then, several other works have come up aiming to address very low-light conditions [13, 33, 25] but we still find SID having the best tradeoff between speed and accuracy for single-frame images.

3. Fast restoration of low-light LF

Fig. 2 shows our three-stage network for restoring $n \times n$ light field images captured in extreme low-light conditions. The four main challenges addressed by this network towards low-light LF restoration are denoising, epipoles preservation, color restoration and fast inference.

Stage-I looks at all SAIs to have a global understanding, Stage-II operates on specific views and Stage-III exclusively focuses on the local neighborhood of SAIs to be restored. Our RNN inspired feedforward network then fuses these complementary information to restore multiple SAIs in a single forward pass.

3.1. Network architecture

Stage-I: Generally, multiple image denoising [32, 15] gives better results than single image denoising because of utilizing the complementary information present in different shots. For a LF camera, however, this complementary information is readily available in a single camera exposure, in the form of SAIs. Thus, in Stage-I, we depth-wise concatenate all the low-light LF views and pass them through a convolutional layer to produce a *global activation* having the same spatial resolution as LF views. This *global activation* is necessary to preserve the LF geometry across all LF views.

Stage-II: While Stage-I looks at all the SAIs simultaneously, Stage-II operates on SAIs individually. Stage-II randomly selects a non-peripheral LF view and depth-wise concatenates the global activation to it. The two are then fused using a single convolutional layer. The convolutional layer's weights are estimated using a series of fully connected layers and angular indices, (s, t), of the chosen SAI.

Stage-III: Stage-III selects the 3×3 neighborhood of the LF view chosen in Stage-II. Stage-I's global activation, Stage-II's view-specific output and these nine LF views are then depth-wise concatenated. Our feedforward network then processes this combined tensor to restore the entire 3×3 neighborhood jointly. Concatenating the 3×3 neighborhood at the beginning of Stage-III is very crucial. Without this step, it becomes extremely difficult for the network to capture the small baseline between LF views, and as a result, the epipolar geometry of LFs is destroyed in the restored LFs.

For each non-peripheral view selected in Stage-II, the entire 3×3 neighborhood gets restored at the end of Stage-III. Thus, to save computation, views with a non-overlapping neighborhood should be selected in Stage-II. But, in cases where n is not a multiple of 3, there will be instances where a view will be restored twice. In such cases, we randomly chose one of them. More complicated measures such as using a weighted average to combine them may be used, but we did not find them to improve the restoration quality.

Pre-processing: Under extreme low-light conditions, the colors captured by any optical system are very poor with low-intensity values. Restoring colors, thus, generally requires amplifying the input image, as adopted by SID [3] and L3Fnet [26]. Much of this amplification in our case is implicitly done by the network by using larger weights and biases for convolutional layers. If, however, the low-light image amplification is also externally supervised to help the network adjust to different lighting conditions, the restoration quality enhances. Thus, taking inspiration from L3Fnet, we compute a six bin histogram from the green channel of the incoming low-light LF and jointly learn the weights for each bin. Using these six weights, we compute a weighted average of the histogram values and multiply it

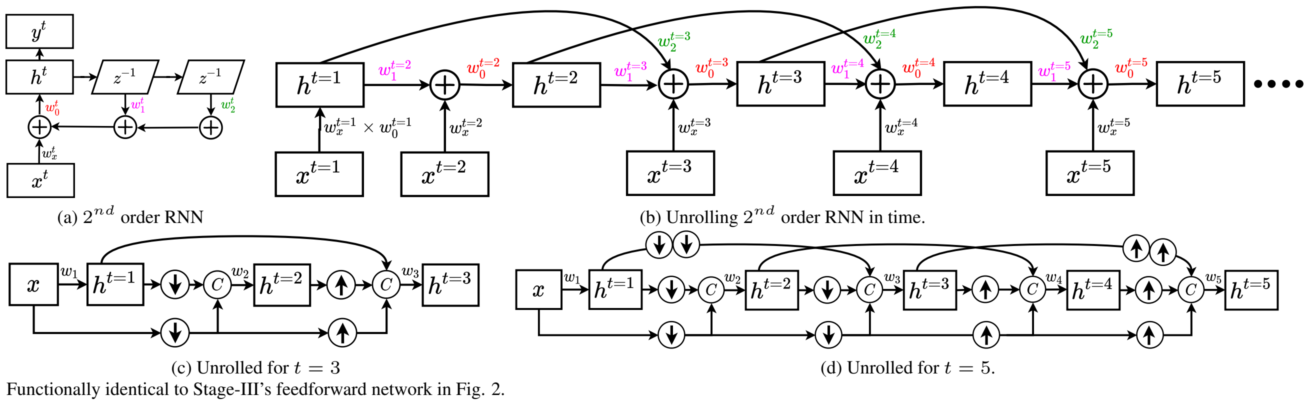

Figure 3. Various steps involved in designing the feedforward network used in Stage-III of the proposed solution.

with the whole LF before feeding to our network.

Loss function: Let I and $\hat{I}$ denote the GT and restored LF, $I_{s,t}$ and $\hat{I}_{s,t}$ denote SAIs of GT and restored LF, $\mathcal{F}(\cdot)$ denote the *amplitude* of a 2D DFT operation, ψ denote the $relu2\_2$ and $relu3\_3$ convolutional layers [22] of VGG-16 architecture. Then the loss function for training is,

$$\frac{0.2}{n^2}\sum_{s,t}\Big(||\mathcal{F}(I_{s,t})-\mathcal{F}(\hat{I}_{s,t})||_1+||\psi(I_{s,t})-\psi(\hat{I}_{s,t})||_1\Big)+||I-\hat{I}||_1$$

Occasionally the L3F dataset [26] exhibits very small translational misalignment. Thus, we use DFT amplitude loss, which is invariant to small translational shifts in signal. We could have also used contextual loss [34], but this slows down the training by a factor of $3\times$ with almost no gain in performance.

3.2. Estimating weights in Stage-II

Stage-I and Stage-III of our network share weights across all SAIs. ResLF [59] instead used a different set of network weights for each SAI for superior restoration but at the expense of significantly increasing the model size. Thus, in our network only Stage-II uses view-specific convolutional weights. One drawback with this approach is that this limits the model in restoring only those SAIs during inference whose weights were learned during training. We, however, want our network to be flexible enough to restore a 9×9 LF even if trained for a smaller LF, say 7×7 LF. Thus, we resort to learning a mechanism during training that can estimate the convolutional weights using SAIs angular location and whose parameters do not depend on the number of views to be restored.

A convolutional layer requires,

$$\left(k^2\times C_i\times C_o\right)+C_o \tag{1}$$

parameters, where C_i and C_o are the number of input and output channels and $k\times k$ is the size of the kernel. The global activation of Stage-I and LF SAIs, both have three channels. Thus depth-wise concatenation gives $C_i = 6$. C_o and k are set to 3.

To estimate convolutional layer's parameters, we use $C_i \times C_o$ number of fully connected layers, each having k^2 nodes. If convolution biases are also to be estimated, as indicated by the second addend in Eq.(1), an additional fully connected layer can be used in the end having C_o nodes (not shown in Fig. 2). ReLU nonlinearity is present after each layer. Input to these fully connected layers are the angular indices (s, t), and the values obtained at each node, after the forward pass, are reshaped to become the convolutional layer's parameters. For measuring (s, t), the central SAI is considered the origin. For example, in a 9×9 LF, the extreme top-left SAI will have $s = t = -4$. In summary, if $f(\cdot)$ denotes the action of fully connected layers, $cat(\cdot,\cdot)$ denotes depth-wise concatenation, $*$ denotes 2D convolution and $I_{s,t}$ denotes (s, t) SAI of low-light LF, then the output of Stage-II is,

$$\begin{aligned} ConvWeights &= reshape\left[f(s,t)\right] \\ StageIIo/p &= ConvWeights * cat(StageIo/p, I_{s,t}) \end{aligned} \tag{2}$$

3.3. RNN inspired feedforward network

U-net's multi-scale architecture has been immensely successful in the Computer Vision community but was not designed to model long-range dependencies between different LF views. In contrast, RNNs were specially designed for sequence modeling. We thus analyze a N^{th} order RNN [46], unfold it in time and propose a set of rules to transform them into a feedforward network.

We model the hidden state h^t of a N^{th} order RNN that keeps track of N preceding states as

$$h^t = w_0^t\left(w_x^t x^t + \sum_{j=1}^{N} w_j^t\cdot h^{t-j}\right). \tag{3}$$

Here, h^t and x^t are the hidden state and the inputs to the N^{th} order RNN at timestamp t. The terms denoted by w are the RNN weights. A larger N implies more memory units and so to keep the model complexity low, we fix $N = 2$ as

2^{nd} order RNN	Feedforward network
$w_x^{t=1} \times w_0^{t=1}$	w_1 (Residual block)
$w_1^{t=2}$	$2\times\downarrow$
$w_x^{t=2}$	$2\times\downarrow$
$w_0^{t=2}$	w_2 (Residual block)
$w_1^{t=3}$	$2\times\uparrow$
$w_2^{t=3}$	Identity
$w_x^{t=3}$	Identity
$w_0^{t=3}$	w_3 (Residual block)

Table 1. Using the proposed set of rules given in Sec. 3.3 to unfold a 2^{nd} order RNN into a feedforward network shown in Fig. 3 c).

shown in Fig. 3 a). Since feedforward networks are simpler to train and have faster inference speed, we unfold the 2^{nd} order RNN in time as described by Soltani and Jiang [46] and is shown in Fig. 3 b). Taking inspiration from multi-scale processing of U-net's, we now propose the following rules to transform them into feedforward networks:

1. We replace all element-wise addition operations with channel-wise concatenation operation.
2. At each timestamp, the input x^t is the channel-wise concatenated global activation of Stage-I, view-specific output of Stage-II and 3×3 neighborhood from Stage-III and is denoted by x.
3. RNN weights which lead to the formation of a feature map are Residual blocks [17] with 3×3 kernel. In our case these weights are w_0^t, denoted in red in Fig. 3 b). As after unrolling a RNN into a feedforward network, the conception of timestamp t becomes meaningless, w_0^t will be denoted simply as w_i, where i is a integer.
4. All other weights are either up/down sampling operations, implemented using Pixel-Shuffle [45, 13].
5. If a weight is looking at N^{th} preceding state, then it can perform up/down sampling operation by a factor of 2^α, where $\alpha \in \{0, N\}$.

Based on the above set of suggested rules, several feedforward networks are possible depending on how many timestamps the RNN is unfolded. Fig. 3 c) and d) show some feedforward networks obtained using the above stated rules, where w_i's are Residual blocks with 3×3 kernels. Infact, Fig. 3 c) is functionally identical to Stage-III's RNN inspired feedforward network shown in Fig. 2. The exact mapping of weights from unfolded 2^{nd} order RNN, shown in Fig. 3 c), to our Stage-III feedforward network is given in Tab. 1. More discussion can be found in supplementary. Key features of these feedforward networks are:

- Each Residual block has direct access to the input x consisting of 3×3 neighborhood of the SAI, leading to better epipoles preservation.
- As we had started with $N = 2$ order RNN, each Residual Block can directly access N preceding feature maps. Thus, a larger N offers much greater expressiveness but with proportionally larger time and memory complexity that might be unnecessary.
- The number of scale-spaces we can have depends on how many timestamps t we unfold the RNN.

4. Experiments

4.1. Experimental settings

We used PyTorch [39] running on Intel Xeon E5-1620V4 CPU with 64GB RAM and GTX 1080Ti GPU. The network shown in Fig. 2 was initialized using MSRA [16] initialization and trained for $50,000$ iterations using ADAM optimizer [24] with a learning rate of 10^{-4}. *Weight normalization* [43] was used for each convolutional layer. All three stages were trained end-to-end for about 10 hours on a single GPU. During training 128×128 patches were randomly cropped from each SAI with batch size of 2 and employing horizontal and vertical flipping. For testing, we used full spatial resolution.

We used the publicly available Low-Light-Light-Field (L3F) dataset [26] collected using commercially available Lytro Illum for benchmarking the proposed solution. L3F dataset was collected in the evening when the light intensity falling on the camera lens was on an average of about 10 lux. Ground truth (GT) images were captured using longer exposure time ranging from $1 - 30$ seconds. The optimal GT exposure time was then reduced by $20\times$, $50\times$ and $100\times$ to capture the low-light LFs and were arranged into L3F-20, L3F-50 and L3F-100 subsets. The L3F-100 is the most challenging amongst the three subsets. Each set consists of 27 scenes, of which 9 are reserved for testing. Each LF consists of 15×15 views. Peripheral views suffer ghosting and vignetting artifacts [59, 53], and the prevailing practice is to ignore views equally from all boundaries [59, 53, 54, 55, 35] and evaluate for central $n \times n$ views, $n \in [5, 7, 9]$. But oddly, L3Fnet ignored more views from the top and left boundary and evaluated for central 8×8 views. To align our evaluation with most existing works on LFs, we evaluate all algorithms for central 9×9 views.

We also show results on the Stanford General Light Field dataset [41] by simulating low-light conditions. To simulate low-light, we first divide the intensity by $s \in [9, 11]$ and then darken it using gamma correction with $\gamma \in [1.5, 2]$. Finally, signal-dependent Poisson noise is added to simulate the photon noise. The dataset consists of 57 scenes, of which 17 were reserved for testing.

We compare the proposed method against 9 existing methods, namely PBS [57], RetinexNet [4], DID [33], SGN [13], SID [3], LFBM5D [2], MTO [58], ResLF [59]

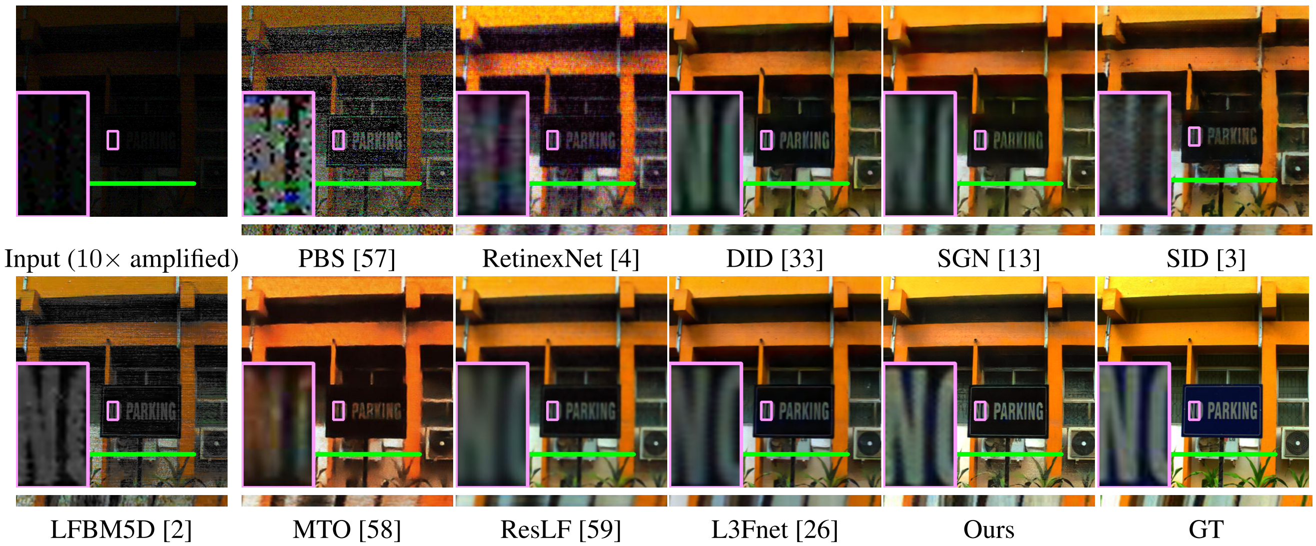

Figure 4. Visual and epipolar comparison of the restorations achieved by our method and existing approaches on the L3F-100 dataset. The figure shows only the central SAI.

	L3F-20	L3F-50	L3F-100
PBS [57]	20.60/0.66	16.04/0.51	13.34/0.36
RetinexNet [4]	21.22/0.70	18.38/0.57	17.21/0.40
DID [33]	23.47/0.76	22.03/0.66	20.08/0.59
SGN [13]	23.50/0.74	22.56/0.65	20.10/0.57
SID [3]	23.91/0.74	22.27/0.64	20.15/0.56
LFBM5D [2]	23.86/0.77	20.32/0.62	18.01/0.44
MTO [58]	24.02/0.75	22.60/0.66	20.20/0.57
ResLF [59]	24.56/0.79	22.98/0.70	21.40/0.66
L3Fnet [26]	24.63/0.80	23.07/0.72	21.99/0.68
Ours	**25.24/0.81**	**24.05/0.73**	**23.45/0.71**

Table 2. PSNR(dB)/SSIM comparison of our method with existing methods on the L3F-20, L3F-50 and L3F-100 datasets [26]. **Bold** represents best value and underline indicates second-best.

	Parameters (million ↓)	MACS (giga ↓)	GPU (msec. ↓)	CPU (sec. ↓)
SGN	3.9	4,645	2,399	87
SID	7.7	4,290	2,979	154
MTO	4.7	18,412	12,145	604
ResLF	7.78	29,820	14,399	630
L3Fnet	3.8	15,213	5,439	270
Ours	**2.9**	**2,423**	**450**	**31**

Table 3. Computational complexity of exiting methods.

and L3Fnet [26]. We compare against single-frame low-light restoration methods, namely PBS, RetinexNet, DID, SGN and SID because unlike recently proposed MTO and L3Fnet, very few methods directly aim for low-light LF restoration. All methods were re-trained/finetuned on low-light LF dataset using publicly available codes. The code for MTO was obtained from authors upon request.

To adapt ResLF for low-light restoration we removed the last Pixel-Shuffle block to match input and output spatial resolution and retrained it on L3F dataset. We also observed that ResLF only operates on intensity channel and simply extrapolates the color channels. This may be alright for recovering high frequency details for Super-Resolution tasks but not for color enhancement and denoising operation required for low-light restoration. We thus re-trained the network with all RGB channels. We also tried using our loss function instead of just L1 loss used by ResLF. All above modifications to ResLF gave atleast $3dB$ higher PSNR, and so we use this version for comparisons. Likewise, MTO was majorly designed for grayscale synthetic low-light LF and so naturally its performance on real low-light RGB LF was poor. So we re-trained it with RGB low-light LFs and use this version for comparisons. Finally, a limitation of L3Fnet is that it cannot restore peripheral views. Thus to get 9×9 restored LF from L3Fnet it was given additional information and was provided with 11×11 central SAIs. The code will be available at *mohit-lamba94.github.io/DarkLightFieldRestoration*

4.2. Comparison with existing methods

Quantitative comparisons. Tab. 2 presents a quantitative comparison of our method with existing approaches on the real low-light L3F dataset. Our method significantly outperforms all existing approaches in terms of PSNR/SSIM metrics. The L3F-100 dataset is very challenging compared to the L3F-20 and L3F-50 datasets because of extremely low pixel intensity and significant photon noise. Thus all methods have the lowest PSNR/SSIM

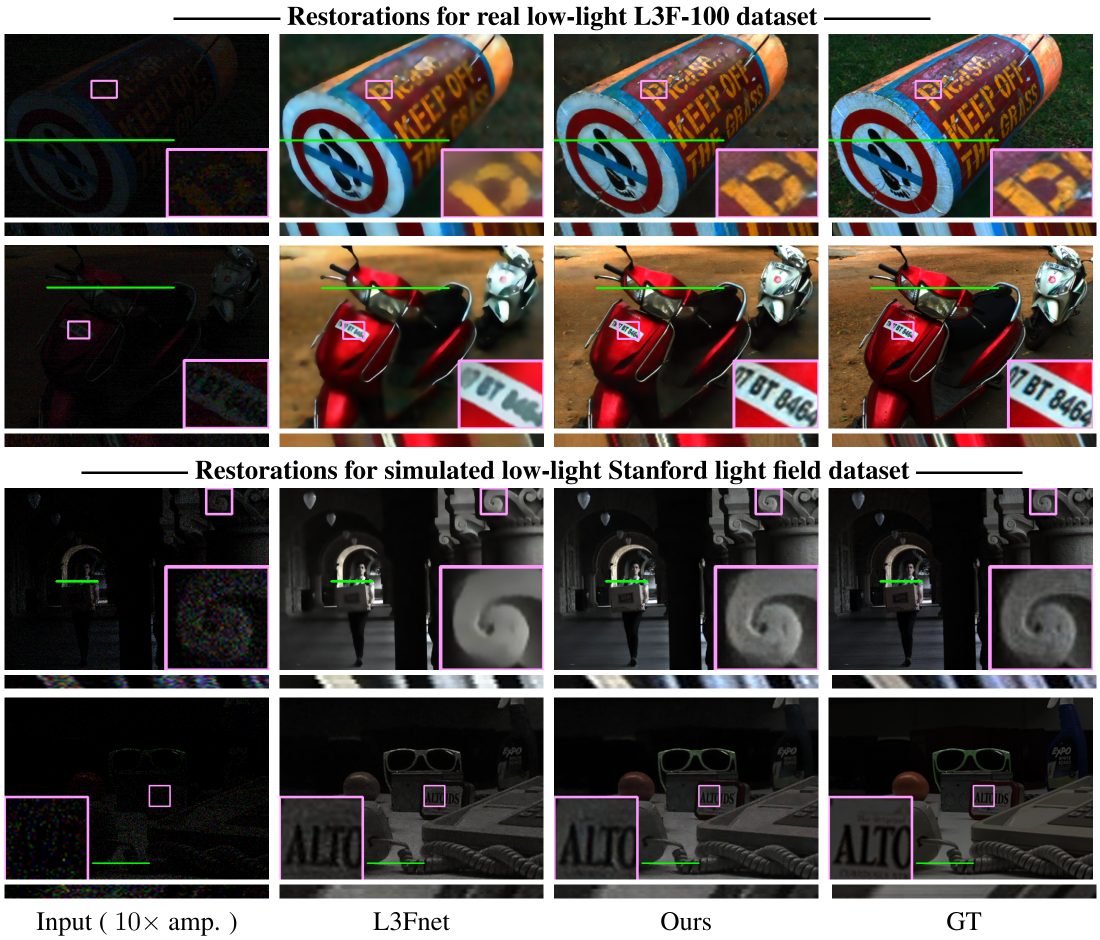

Figure 5. More visual and epipolar comparisons between the state-of-the-art L3Fnet and our method. Our method is able to restore finer details much better than L3Fnet with less blurriness. The input images of L3F-100 dataset are $10\times$ amplified for visualization.

for the L3F-100 dataset and the highest for the L3F-20 dataset. A side evidence of the effectiveness of our method is that the difference in the restoration quality between the L3F-20 and L3F-100 dataset is only about $2dB$ for us, but for other methods such as PBS it is more than $7dB$.

Qualitative comparisons. Methods like PBS, RetinexNet and LFBM5D have quite noisy restorations. These methods are appropriate for mild denoising but cannot tackle excessive photon noise in very dark conditions. On the contrary, restorations of methods like SID and SGN are quite blurry as they do not jointly utilize information from all SAIs. Though we re-trained MTO on real low-light LFs, it struggles to recover colors from low-light LFs. Most results shown in MTO paper [58] are for grayscale synthetic low-light LFs. Although, ResLF uses a much deeper network, its restorations are lower than ours. This is mostly because, unlike U-net architecture, ResLF does not perform multi-scale processing and so has a very small receptive field (about 25×25). This may be sufficient for recovering details for super-resolution task but not for color restoration and noise suppression in real low-light LFs. Compared to the current state-of-the-art L3Fnet, our restorations better preserve finer details. This fact is also corroborated by additional visual comparisons shown Fig. 5. Quantitatively also, L3Fnet achieves a PSNR/SSIM of $28.76dB/0.85$ on the Stanford general light field dataset, while our method achieves $29.30dB/0.86$.

Computational Complexity. Tab. 3 shows the computational complexity of the top-performing models for restoring a 9×9 LF with 432×624 spatial resolution. We observe that our method offers a significant improvement for every metric. Specifically, compared to state-of-the-art L3Fnet, we have a 23% smaller model size (i.e number of parameters), use about $5\times$ lower floating point operations (i.e GMACS) and about $9\times$ faster on both GPU and CPU. The main reason for our extremely fast inference is that we requires only 9 forward pass of Stage-II and Stage-III to restore a single 9×9 LF. In contrast, other methods require 81 forward pass. One final limitation of L3Fnet is that, given a $n \times n$ LF, it can restore only central $n-2 \times n-2$ views.

	SID	MTO	ResLF	L3Fnet	Ours
PSNR (db)	19.01	19.31	21.10	21.90	**23.44**
SSIM	0.48	0.51	0.56	0.60	**0.66**

Table 4. Average PSNR/SSIM of EPIs constructed from LFs restored by different methods on L3F-100 dataset.

Train: 5×5 Test: 9×9 | Train: 7×7 Test: 9×9 | Train: 9×9 Test: 9×9

Figure 6. Our network's generalization capability in restoring LFs of $m \times m$ angular resolution even if trained for a $n \times n$ ($n < m$) LF. Irrespective of training resolution, i.e. $n \in [5, 7, 9]$, the restoration quality and the epipoles for restoring a 9×9 LF are comparable.

Epipolar comparisons. The best way to analyze the LF geometry after restoration is through GIF animations, which can be found in the supplementary. We also compare the PSNR/SSIM of Epipolar Planar Images (EPIs) quantitatively in Tab. 4 on the L3F-100 dataset and find that the LF geometry is well preserved in our restorations. To compute EPIs, we randomly selected 10 rows from the central SAI and formed the EPIs by collecting rows at same spatial location from every other SAI.

4.3. Generalizing to LFs of different sizes

Fig. 6 demonstrate the flexible nature of our network in restoring a $m \times m$ LF even if trained for a smaller $n \times n$ ($n < m$) LF. We trained our model for three angular resolutions, namely 5×5, 7×7 and 9×9 on the L3F-100 dataset and tested them for restoring all 9×9 LF views. During testing, models trained for $n \times n$ ($n \in [5, 7, 9]$) views only considered the central $n \times n$ views of the incoming low-light 9×9 LF for Stage-I. Stage-II and Stage-III remain as described in Sec. 3. Quantitatively, the drop in PSNR/SSIM for training on 5×5 LF instead of 9×9 is $0.6dB/0.03$ and for training on 7×7 LF is only $0.4dB/0.02$. Thus, although the best practice is to train on largest possible angular resolution, our network does a decent job even if trained for smaller angular resolution, which could be useful when limited training data is available. Such generalization is not possible with existing networks.

4.4. Ablation studies

Tab. 5 shows several ablation studies conducted to understand the effectiveness of different components of our model. We conducted the ablation studies on the L3F-100 dataset and re-trained the model in each case.

In the first ablation study we replaced our RNN inspired feedforward network with a regular U-net. In doing so we

	PSNR / SSIM
Use U-net in Stage-III	23.02/0.70
Share conv weights in Stage-II	23.10/0.70
No neighbouring SAIs in Stage-III	22.58/0.68
Unfold RNN for $t = 5$ instead of $t = 3$	**23.69/0.71**
Training without DFT loss	22.17/0.68
Proposed	<u>23.45</u>/**0.71**

Table 5. Ablation studies for our network on the L3F-100 dataset. Though using the feedforward network unrolled for $t = 5$ has higher PSNR, it has much greater time and computational complexity and we prefer using the feedforward network with $t = 3$.

did not change the number of model parameters nor the number of convolutional layers. The PSNR dropped from 23.45 dB to 23.02 dB. This is expected because in contrast to U-net, each residual block present in our RNN inspired feedforward has direct access to input LF views and upto 2 preceding feature maps.

In the second ablation study, we use only a single convolutional layer in Stage-II and share its weights across all SAIs. Consequently, the network now becomes oblivious to the angular location of the SAIs and we find the PSNR drops to 23.10 dB.

In the third ablation study, we do not feed the 3×3 neighborhood to our RNN inspired feedforward network. The PSNR significantly drops to 22.58 dB and we also observed that the epipoles were not appropriately restored. Besides Stage-II and Stage-III, Stage-I is required to preserve epipolar geometry across views, as demonstrated in L3Fnet.

In the fourth ablation study, we replaced our feedforward network present in Stage-III with the one unrolled for $t = 5$ instead of $t = 3$. This increased the PSNR to 23.69 dB but with much greater computational complexity. We thus continue to use the $t = 3$ feedforward network.

Finally, re-training the network without the DFT loss causes significant drop in PSNR.

5. Conclusion

We proposed a novel three-stage network that can be trained end-to-end to utilize three complementary information present in real low-light LFs, namely global, local and view-specific. These complementary information were then fused together by our RNN inspired feedforward network to restore very low-light LFs. Additionally, the network restores multiple SAIs in a single forward pass for significantly faster inference. Overall, the combined effect of these contributions were that compared to state-of-the-art L3Fnet, we achieved up to 1 dB higher restoration PSNR, with $9\times$ speedup, 23% smaller model size and about $5\times$ lower floating-point operations. Finally, we also showed that our network is flexible enough to restore a 9×9 LF during inference even if trained for 5×5 or 7×7 LF.

References

[1] Edward H Adelson and John Y. A. Wang. Single Lens Stereo with a Plenoptic Camera. *IEEE Trans. Pattern Analysis & Machine Intelligence*, 14(2):99–106, 1992.

[2] Martin Alain and Aljosa Smolic. Light Field Denoising by Sparse 5D Transform Domain Collaborative Filtering. In *Int. Workshop on Multimedia Signal Processing*, pages 1–6, 2017.

[3] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to See in the Dark. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

[4] Wei Chen, Wang Wenjing, Yang Wenhan, and Liu Jiaying. Deep retinex decomposition for low-light enhancement. In *British Machine Vision Conference*, 2018.

[5] Perwa Christian and Wietzke Lennart. Raytrix: Light Field technology. `https://raytrix.de`. [Accessed July-2020].

[6] Dinu Coltuc, Philippe Bolon, and J-M Chassery. Exact histogram specification. *IEEE Transactions on Image Processing*, 15(5):1143–1152, 2006.

[7] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering. *IEEE Trans. Image Process.*, 16(8):2080–2095, 2007.

[8] Donald G Dansereau, Daniel L Bongiorno, Oscar Pizarro, and Stefan B Williams. Light Field Image Denoising using a Linear 4D Frequency-Hyperfan all-in-focus Filter. In *Computational Imaging XI*, volume 8657, page 86570P. Int. Society for Optics and Photonics, 2013.

[9] Donald G Dansereau, Oscar Pizarro, and Stefan B Williams. Decoding, Calibration and Rectification for Lenselet-Based Plenoptic Cameras. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2013.

[10] Xueyang Fu, Delu Zeng, Yue Huang, Yinghao Liao, Xinghao Ding, and John Paisley. A fusion-based enhancing method for weakly illuminated images. *Signal Processing*, 129:82–96, 2016.

[11] Xueyang Fu, Delu Zeng, Yue Huang, Xiao-Ping Zhang, and Xinghao Ding. A weighted variational model for simultaneous reflectance and illumination estimation. In *CVPR*, 2016.

[12] Steven J Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F Cohen. The Lumigraph. In *Siggraph*, volume 96, pages 43–54, 1996.

[13] Shuhang Gu, Yawei Li, Luc Van Gool, and Radu Timofte. Self-guided network for fast image denoising. In *IEEE Int. Conf. Comput. Vis.*, 2019.

[14] Xiaojie Guo, Yu Li, and Haibin Ling. LIME: Low-Light Image Enhancement via Illumination Map Estimation. *IEEE Trans. Image Process.*, 26(2):982–993, 2017.

[15] Samuel W Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan T Barron, Florian Kainz, Jiawen Chen, and Marc Levoy. Burst photography for high dynamic range and low-light imaging on mobile cameras. *ACM Trans. Graph. (TOG)*, 35(6):192, 2016.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[18] Fu-Chung Huang, David P Luebke, and Gordon Wetzstein. The light field stereoscope. In *SIGGRAPH Emerging Technologies*, pages 24–1, 2015.

[19] Haidi Ibrahim and Nicholas Sia Pik Kong. Brightness preserving dynamic histogram equalization for image contrast enhancement. *IEEE Transactions on Consumer Electronics*, 53(4):1752–1758, 2007.

[20] Hae-Gon Jeon, Jaesik Park, Gyeongmin Choe, Jinsun Park, Yunsu Bok, Yu-Wing Tai, and In So Kweon. Accurate depth Map Estimation from a Lenslet Light Field Camera. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2015.

[21] Jing Jin, Junhui Hou, Jie Chen, and Sam Kwong. Light field spatial super-resolution via deep combinatorial geometry embedding and structural consistency regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2260–2269, 2020.

[22] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[23] Yeong-Taeg Kim. Contrast enhancement using brightness preserving bi-histogram equalization. *IEEE transactions on Consumer Electronics*, 43(1):1–8, 1997.

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *ICLR*, 2015.

[25] Mohit Lamba, Atul Balaji, and Kaushik Mitra. Towards fast and light-weight restoration of dark images. In *British Machine Vision Conference*, 2020.

[26] Mohit Lamba, Kranthi Kumar Rachavarapu, and Kaushik Mitra. Harnessing multi-view perspective of light fields for low-light imaging. *IEEE Transactions on Image Processing*, 30:1501–1513, 2021.

[27] Edwin Land. The retinex. *American Scientist*, 52(2):247–264, 1964.

[28] Edwin H Land. The Retinex Theory of Volor Vision. *Scientific American*, 237(6):108–129, 1977.

[29] Chulwoo Lee, Chul Lee, and Chang-Su Kim. Contrast enhancement based on layered difference representation of 2d histograms. *IEEE transactions on image processing*, 22(12):5372–5384, 2013.

[30] Marc Levoy and Pat Hanrahan. Light Field Rendering. In *Conf. Comput. graphics and interactive techniques*. ACM, 1996.

[31] Mading Li, Jiaying Liu, Wenhan Yang, Xiaoyan Sun, and Zongming Guo. Structure-revealing low-light image enhancement via robust retinex model. *IEEE Transactions on Image Processing*, 27(6):2828–2841, 2018.

[32] Ziwei Liu, Lu Yuan, Xiaoou Tang, Matt Uyttendaele, and Jian Sun. Fast burst images denoising. *ACM Transactions on Graphics (TOG)*, 33(6):1–9, 2014.

[33] Paras Maharjan, Li Li, Zhu Li, Ning Xu, Chongyang Ma, and Yue Li. Improving extreme low-light image denoising via residual learning. In *Int. Conf. Multimedia and Expo*, 2019.

[34] Roey Mechrez, Itamar Talmi, and Lihi Zelnik-Manor. The Contextual Loss for Image Transformation with Non-Aligned Data. In *Eur. Conf. Comput. Vis.*, 2018.

[35] Kaushik Mitra and Ashok Veeraraghavan. Light Field Denoising, Light Field Superresolution and Stereo Camera Based Refocussing Using a GMM Light Field Patch Prior. In *IEEE Comput. Vis. Pattern Recog. Workshops*, 2012.

[36] Ren Ng. Fourier slice photography. In *ACM Trans. Graph.(TOG)*, volume 24, pages 735–744, 2005.

[37] Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, Pat Hanrahan, et al. Light Field Photography with a Hand-Held Plenoptic Camera. *Comput. Science Technical Report CSTR*, 2(11):1–11, 2005.

[38] Sotiris Nousias, Manolis Lourakis, and Christos Bergeles. Large-Scale, Metric Structure from Motion for Unordered Light Fields. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NIPS*. 2019.

[40] Stephen M Pizer, E Philip Amburn, John D Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart ter Haar Romeny, John B Zimmerman, and Karel Zuiderveld. Adaptive histogram equalization and its variations. *Computer vision, graphics, and image processing*, 39(3):355–368, 1987.

[41] Abhilash Sunder Raj, Michael Lowney, Raj Shah, and Gordon Wetzstein. Stanford Lytro Light Field Archive. `http://lightfields.stanford.edu/general.html`. [Accessed Feb-2021].

[42] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional Networks for Biomedical Image Segmentation. In *Int. Conf. Medical image computing and Comput.-assisted intervention*. Springer, 2015.

[43] Tim Salimans and Durk P Kingma. Weight normalization: A simple reparameterization to accelerate training of deep neural networks. In *NIPS*, 2016.

[44] A. Sepas-Moghaddam, P. L. Correia, and F. Pereira. Light Field Denoising: Exploiting the Redundancy of an Epipolar Sequence Representation. In *True Vis. Capture, Transmission and Display of 3D Video (3DTV-CON)*, pages 1–4, 2016.

[45] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[46] Rohollah Soltani and Hui Jiang. Higher order recurrent neural networks. *arXiv preprint arXiv:1605.00064*, 2016.

[47] J Alex Stark. Adaptive image contrast enhancement using generalizations of histogram equalization. *IEEE Transactions on image processing*, 9(5):889–896, 2000.

[48] Michael W Tao, Sunil Hadap, Jitendra Malik, and Ravi Ramamoorthi. Depth from combining defocus and correspondence using light-field cameras. In *IEEE Int. Conf. Comput. Vis.*, 2013.

[49] Ruixing Wang, Qing Zhang, Chi-Wing Fu, Xiaoyong Shen, Wei-Shi Zheng, and Jiaya Jia. Underexposed Photo Enhancement Using Deep Illumination Estimation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[50] Shuhang Wang, Jin Zheng, Hai-Miao Hu, and Bo Li. Naturalness preserved enhancement algorithm for non-uniform illumination images. *IEEE Trans. Image Process.*, 22(9):3538–3548, 2013.

[51] Tiantian Wang, Yongri Piao, Xiao Li, Lihe Zhang, and Huchuan Lu. Deep learning for light field saliency detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[52] Ting-Chun Wang, Alexei A Efros, and Ravi Ramamoorthi. Occlusion-Aware Depth Estimation using Light-Field Cameras. In *IEEE Int. Conf. Comput. Vis.*, 2015.

[53] Yunlong Wang, Fei Liu, Kunbo Zhang, Guangqi Hou, Zhenan Sun, and Tieniu Tan. LFNet: A Novel Bidirectional Recurrent Convolutional Neural Network for Light-Field Image Super-Resolution. *IEEE Trans. Image Process.*, 27(9):4274–4286, 2018.

[54] Sven Wanner and Bastian Goldluecke. Spatial and Angular Variational Super-Resolution of 4D Light Fields. In *Eur. Conf. Comput. Vis.*, pages 608–621. Springer, 2012.

[55] Youngjin Yoon, Hae-Gon Jeon, Donggeun Yoo, Joon-Young Lee, and In So Kweon. Learning a Deep Convolutional Network for Light-Field Image Super-Resolution. In *Int. Conf. Comput. Vis. Workshop*, 2015.

[56] Jun Zhang, Meng Wang, Liang Lin, Xun Yang, Jun Gao, and Yong Rui. Saliency detection on light field: A multi-cue approach. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 13(3):1–22, 2017.

[57] Qing Zhang, Ganzhao Yuan, Chunxia Xiao, Lei Zhu, and Wei-Shi Zheng. High-quality exposure correction of underexposed photos. In *ACM Int. Conf. Multimedia*, 2018.

[58] Shansi Zhang and Edmund Y Lam. Learning to restore light fields under low-light imaging. *Neurocomputing*, 456:76–87, 2021.

[59] Shuo Zhang, Youfang Lin, and Hao Sheng. Residual Networks for Light Field Image Super-Resolution. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.