Mitigating Optical Flare Artifacts: Migrating Towards a Transformer-Based Approach

David Tomarov
Department of Computer Science,
University of Toronto
david.tomarov@mail.utoronto.ca

Wenrui Wu
Department of Computer Science,
University of Toronto
wenrui.wu@mail.utoronto.ca

Anuj Chavan
Dept. of Electrical and Computer
Engineering, University of Toronto
anuj.chavan@mail.utoronto.ca

Abstract—Lens flare is a common artifact in photography, caused by unintended reflections and scattering within a camera's optical system. Despite advances in optical design, consumer cameras frequently capture images with distracting flare artifacts that obscure details and reduce image quality. These artifacts vary widely in appearance due to the complex interplay of lens design, manufacturing imperfections, and environmental factors like dust and scratches, rendering them a random phenomena. Existing computational methods for flare removal often rely on simplistic assumptions or hardware-specific solutions, limiting their generalization and effectiveness in real-world scenarios. To address these challenges, we try to improve and further, build upon existing learning-based approach that leverages semi-synthetic data for training. The existing framework generates diverse, realistic flare-corrupted image pairs using a combination of wave-optics modeling for scattering effects and a data-driven approach for internal reflections. This dataset allows us to train more recent architectures (previously underutilized in flare reduction) such as U-Net++, U-Net3+ and ViT to remove flare while preserving light sources, paving out a way to integrate vision based transformer approaches in place of the original U-Net implementation. Our models, trained exclusively on semi-synthetic data, generalize well to real-world images, demonstrating robust performance across diverse scenes and flare types. The codes and trained models are publicly available at *https://github.com/Convolution/computational_imaging.git*

Index Terms—Computational Imaging, Lens Flare Removal, Convolutional Networks (U-Nets), Transformers, Encode Decoder

1 Introduction

STRONG light source scenes in photographs frequently display lens flares, which are noticeable visual artifacts brought on by accidental reflections and scattering within the camera system. Flare artifacts can obscure image content, diminish detail, and be distracting. Even tiny light sources can result in noticeable artifacts when captured by consumer cameras, despite tremendous efforts in optical design to reduce lens flare.

The lens's optics, the light source's position, manufacturing flaws, and dust and scratches from regular use all affect flare patterns. Lens flare manifests in a variety of ways due to the wide range of underlying causes. Halos, streaks, brilliant lines, saturated blobs, color bleeding, haze, and many more are examples of common artifacts. Because of this variability, removing flares is a notoriously difficult and random phenomenon.

The majority of lens flare removal techniques now in use naively rely on template matching or intensity thresholding to locate and identify the artifact, failing to take into consideration the physics of flare creation. As a result, they are not effective in more complicated real-world situations and can only identify and maybe eliminate a small variety of flares, such as saturated blobs.

The absence of training data is the primary obstacle. A mechanism to "switch" the artifacts on and off without also altering the scene's illumination would be necessary, and collecting a large number of perfectly aligned image pairs with and without lens flare would be laborious at best and impossible at worst. This can be done with a lot of work by gathering pairs of tripod-shot photos in which the photographer manually positions an occluder between the camera and the illuminant in one picture. However, this method is too time-consuming to generate the dozens or millions of image pairs that are often needed for neural network training. Additionally, this limits usability as it works only when the flare-causing illuminant lies outside of the camera's field of view.

To get around this problem, we add artificial flares to the data that are produced using a wave optics-based model that simulates the scattering situation (such as dust, scratches, and other flaws). Since an exact optical model for a commercial camera is frequently unavailable, a rigorous data-driven technique is utilized to address the unintentional reflections between lens elements. We are able to produce a sizable and varied dataset of semi-synthetic flare-corrupted photos using this formulation, together with

ground-truth flare-free images.

Another challenge is removing flare while keeping the visible light source intact. Even with semi-synthetic data, this is challenging since we are unable to isolate the light source from the flare-only layer without altering the flare it causes. As a result, if the network is trained naively, it will attempt to eliminate both the flare and the light source, producing outputs that are implausible. In order to do this, we employ a post-processing step to maintain the light source in the output and a loss function that disregards the light source region.

We minimize a loss function on the residual (also known as the inferred flare) and the expected flare-free image during training. The networks can eliminate various forms of flare in a range of scenarios and only need a single RGB image captured by a regular camera during testing. The models perform well when applied to real-world photos, despite being trained solely on semi-synthetic data.

2 Related Work

2.1 Hardware solutions

To reduce flare, high-end camera lenses frequently use complex optical designs and materials. Every glass component that is added to a compound lens to enhance image quality also increases the likelihood that light will be reflected off of its surface, producing flare. Applying anti-reflective (AR) coating to lens components is a popular method that lowers internal reflection by destructive interference. Nevertheless, this coating's thickness cannot be ideal because it can only be tailored for specific wavelengths and angles of incidence. Furthermore, it is costly to apply an AR coating to every optical surface, and it may conflict with or prevent other coatings (such as anti-scratch and anti-fingerprint).

2.2 Computational methods

A two-step procedure is used by certain systems [1], [2], [3]: first, the scene behind the flare zone is recovered using inpainting [4], and second, the lens flare is detected based on its distinct shape, location, or intensity (i.e., by recognizing a saturated region). These techniques are susceptible to misclassifying all bright regions as flares and only operate on specific sorts of flares, such as bright spots. Furthermore, these methods ignore the fact that the majority of lens flares are better represented as a semitransparent overlay on top of the underlying scene by classifying each pixel as either ″flare″ or ″not flare.″

2.3 Hybrid methods

Researchers have employed computational imaging, in which post-processing algorithms and camera hardware are built together. Using structured occlusion masks, Talvala et al [5] and Raskar et al [6]. tried to block flare-causing light selectively. They then used either direct-indirect separation or a light field-based technique to recover the scene free of flares. Despite their elegance, their usefulness is restricted since they need specialized hardware.

2.4 Learning-based image decomposition

Wu et al. [7] had success removing flares by utilizing a modified U-net architecture, but their work struggled in scenes that had strong flare over the entire image, and they did not consider more recent architectures available nowadays.

3 Physics of lens flare

All of the rays from a point light source should converge and refract to a single spot on the sensor when the camera is in focus. In practice, real lenses scatter and reflect light along unintended paths, resulting in flare artifacts. The scattered and reflected parts only constitute a small fraction of each incident light ray. As a result, flare is ubiquitous but invisible in the majority of photos. However, the tiny percentage of scattered and reflected rays from a bright light source (like the sun) that is many orders of magnitude brighter than the rest of the picture will cause apparent artifacts at other pixels on the image. The geometry of the scattering from dust and scratches, and that of the multiple reflections, result in characteristic visual patterns. On a high level, flares can be classified into two major categories: scattering-induced and reflection-induced.

Scattering flare Although a perfect lens is 100% refractive, actual lenses contain several flaws that scatter light. Either regular wear (dust and scratches) or manufacturing flaws (dents) could be the cause of the scattering (or diffraction). This results in a secondary set of rays that are scattered or diffracted rather than traveling down their intended routes, in addition to the original rays that are refracted. Scratches create streaks that seem to ″emit″ radially from the light source, whereas dust adds a rainbow-like look. Additionally, scattering can make the area surrounding the light source appear hazy by reducing contrast.

Reflective flare Every air–glass interface in a practical lens system presents a chance for a tiny amount of reflection, usually around 4%. These reflected flares usually appear on the image along the straight line that connects the principal point and the light source. As seen in **??**, they are sensitive to the angle of incidence of the light source, but not to rotation about the optical axis. The aperture's geometry, size, and placement determine the flare's shape; if the aperture partially blocks the reflection more than once, arc-shaped artifacts may arise. AR coating can be utilized to lessen reflection, as was previously indicated. But wavelength also affects how effective this coating is, thus lens flare can have a range of hues, usually blue, purple, or pink. Since reflected flare is dependent on lens design, it is predicted that cameras of the same design—for example, all of the primary camera modules of the iPhone 12—will image the same scene with comparable reflective flares.

Challenges in flare removal It's frequently challenging to tell the various kinds of flare apart or distinguish them visually. The position, size, intensity, and spectrum of the light source, as well as the design and flaws of the lens, can all have a substantial impact on how the flare appears. Building a fully physics-based method to analytically detect and eliminate every kind of flare is therefore impossible, particularly when there are several artifacts in a single image.

4 Reconstruction Algorithm

Our goal is to train a model that predict a flare free image, given an image corrupted by flare.

4.1 Losses

The loss function is based on the idea of only the flare caused by the light source should be removed [7]. With that in mind we measure two separate loss and sum them to form our loss function. The first one is the loss of the image with can be measured by a L1 norm and a perceptual loss. Notice we do not want the model to inpaint the whole light source so we take the image region as parts exclude the saturated light source and express it as

$$\hat{I}_0 = I_0 \odot M + f(I_F, \Theta) \odot (1 - M) \tag{1}$$

where M is a binary mask that mask out the satuation if the input I_F is greater than 0.99. The the loss function for the scenes can be written as

$$\mathcal{L}_I = \left\| \hat{I}_0 - I_0 \right\|_1 + \sum_\ell \lambda_\ell \left\| \Phi_\ell(\hat{I}_0) - \Phi_\ell(I_0) \right\|_1 \tag{2}$$

Then we will use a similar loss function to model the flare. First the region of the flare can be calculated by the following formula

$$\hat{F} = I_F - f(I_F, \Theta) \odot (1 - M) \tag{3}$$

Then the flare loss is given by

$$\mathcal{L}_I = \left\| \hat{F} - F \right\|_1 + \sum_\ell \lambda_\ell \left\| \Phi_\ell(\hat{F}) - \Phi_\ell(F) \right\|_1 \tag{4}$$

And the final loss function is simply

$$\mathcal{L} = \mathcal{L}_I + \mathcal{L}_F \tag{5}$$

4.2 Light Source Blending

After removing the flare, we would like to add back the light source to the image without the flare. Since the light source itself is likely saturated, it could be easily identified based on the intensity in mask M. To create a gradual transition for the light source, a smoothing is applied to the mask M, and then introduced back to the image without the flare.

5 Methodology

The typical use of convolutional networks is on classification tasks, where the output to an image is a single classification label. However, in many imaging tasks (particularly in segmentation tasks), the desired output should include localization on the pixel, while capturing global context. Ronneberger et al. [8] proposed the U-Net which showed significant success in various biomedical segmentation tasks, outperforming previous networks.

To address the challenge of lens flare removal, Wu et al. [7] employed a U-Net architecture, drawing inspiration from its success in biomedical segmentation. Recognizing that lens flare is predominantly a low-frequency artifact, they trained the U-Net on a dataset of semi-synthetic images to predict a low-resolution flare-only image. In our work we had limited computational resources, so we downsampled the scene and flare data by a factor of 3 to speed up the training process.

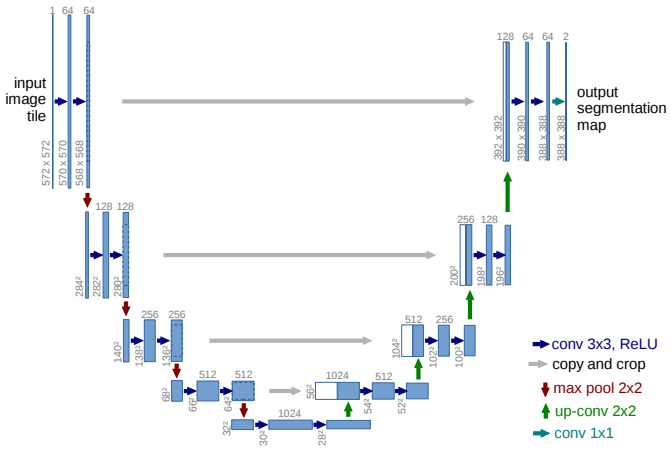


Fig. 1: U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

5.1 Model Architectures

5.1.1 *Baseline U-Net*

The architecture is illustrated in Figure 1. It consists of a contracting path (left side) and an expansive path (right side). The contracting path follows the typical architecture of a convolutional network. It consists of the repeated application of two 3x3 convolutions (unpadded convolutions), each followed by a rectified linear unit (ReLU) and a 2x2 max pooling operation with stride 2 for downsampling. At each downsampling step we double the number of feature channels. Every step in the expansive path consists of an upsampling of the feature map followed by a 2x2 convolution ("up-convolution") that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3x3 convolutions, each followed by a ReLU. The cropping is necessary due to the loss of border pixels in every convolution. At the final layer a 1x1 convolution is used to map each 64-component feature vector to the desired number of classes. In total the network has 23 convolutional layers.

5.1.2 *U-Net++*

The U-Net++ architecture [9] in Figure 2 is a deeply supervised encoder-decoder network that aims to improve the accuracy of medical image segmentation by redesigning the skip pathways that connect the encoder and decoder subnetworks. Instead of directly transferring feature maps from the encoder to the decoder as in the standard U-Net, U-Net++ introduces nested, dense skip pathways that gradually enrich the semantic information of high-resolution feature maps from the encoder before they are fused with the corresponding decoder feature maps. Each skip pathway consists of a dense convolution block with a variable number of convolution layers, depending on the pyramid level. These blocks employ a series of convolution operations and concatenation layers to progressively bridge the semantic gap between the feature maps from different levels of the encoder and decoder. This approach is based on the

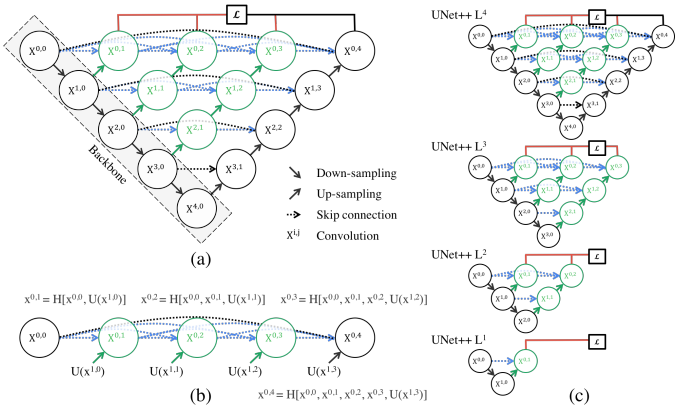


Fig. 2: (a) UNet++ consists of an encoder and decoder that are connected through a series of nested dense convolutional blocks. The main idea behind UNet++ is to bridge the semantic gap between the feature maps of the encoder and decoder prior to fusion. In the figure, black indicates the original U-Net, green and blue show dense convolution blocks on the skip pathways, and red indicates deep supervision. Red, green, and blue components distinguish UNet++ from U-Net. (b) Detailed analysis of the first skip pathway of UNet++. (c) UNet++ can be pruned at inference time, if trained with deep supervision.

hypothesis that the optimization process will be facilitated when the feature maps from the encoder and decoder are semantically similar. By incorporating dense skip pathways, U-Net++ addresses the limitation of standard U-Net, which directly fuses semantically dissimilar feature maps, potentially leading to suboptimal performance. This architectural enhancement enables U-Net++ to more effectively capture fine-grained details of the foreground objects, resulting in more accurate segmentation masks, especially for medical images where precision is critical.

5.1.3 U-Net3+

The U-Net 3+ architecture [10], as described in Figure 3, builds upon the U-Net++ by further enhancing the utilization of multi-scale features and introducing additional components to improve accuracy. While U-Net++ utilizes nested and dense skip connections to reduce the semantic gap between the encoder and decoder, U-Net 3+ proposes full-scale skip connections that incorporate information from all scales of the encoder and decoder, enabling the capture of both fine-grained details and coarse-grained semantics. Each decoder layer receives input from smaller-scale encoder layers via non-overlapping max-pooling, from the same-scale encoder layer directly, and from larger-scale decoder layers via bilinear interpolation, creating a comprehensive multi-scale feature representation. This approach contrasts with U-Net++ which focuses on bridging the semantic gap primarily between adjacent encoder and decoder levels, while U-Net 3+ aims for a more holistic integration of multi-scale information. Additionally, U-Net 3+ incorporates full-scale deep supervision by connecting each decoder stage to a hybrid loss function, allowing for hierarchical learning and optimization at various scales. The hybrid loss function consists of focal loss, MS-SSIM loss, and IoU loss, accounting for pixel-, patch-, and map-level segmentation accuracy, respectively. The MS-SSIM loss, in particular, assigns higher weights to fuzzy boundaries, promoting more accurate segmentation of organ boundaries. Furthermore, U-Net 3+ introduces a classification-guided module (CGM) mitigating over-segmentation inimages by multiplying the classification output with the segmentation output. These advancements collectively enhance the accuracy and efficiency of U-Net 3+ which clearly pose an advantage to the problem at hand.

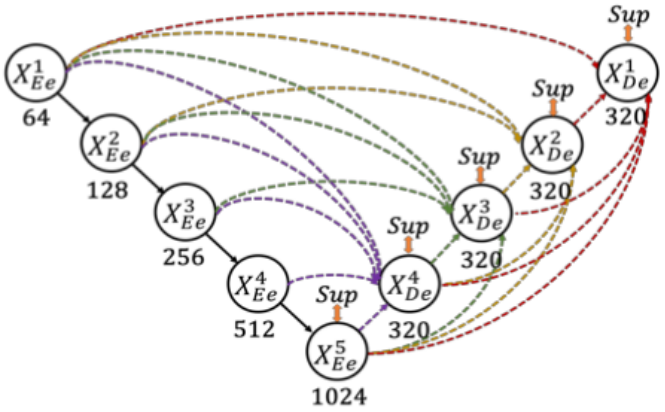


Fig. 3: Each decoder level has skip connections to smaller- and same-scale feature maps from encoder, and larger-scale feature maps from decoder. The black components represent the original U-Net. The green and blue components represent the dense convolution blocks on the skip pathways. The red components represent deep supervision.

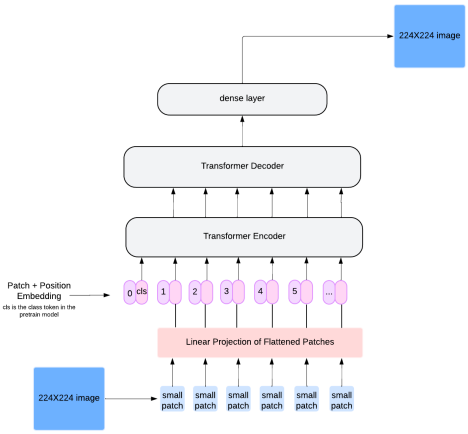


Fig. 4: ViT model

5.1.4 ViT

The overall ViT archetecture is presented in Figure 4. The basic structure is mentioned in [11]. ViT is data hungry model and unfortunately we do not have millions of data for training, to be able to test the model with the data set we have (around 27500 images), we used a pretrained ViT model. However, the original task for the pretrainedmodel is image classification and here we are dealing with image to image task. So we only use the encoder part of it. Then we connect the encoder with our transformer decoder, notice we omit the cls token as it is not needed in our task. Then we pass the output of decoder to a fully connected

network with the number of units equal to patch dimension. Through the dense layer, we got the flattened representation of output image. And finally we reconstruct 224x224 image from the flattened representation.

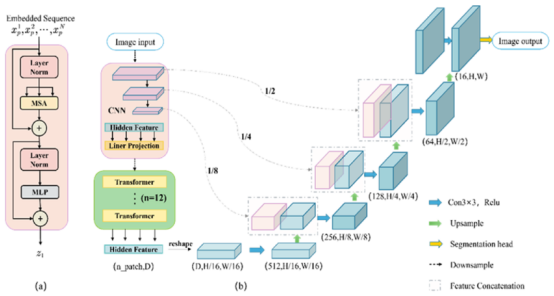


Fig. 5: ViT model

5.1.5 *TransUnet*

Transformers are powerful in modeling global context. However, a classical transformer model exclusively focus on modeling the global feature and might result in low resolution and lack of detailed localization information. Simple upsampling does not work to recover these information. On the other hands, Unet like networks have a great solution for extracting low-level cues. Flares can be considered as global features since they can appear in anywhere in the image. Thus we want to take the advantage of transfomers when extracting global features and also be able to preserve high frequency details in the image. TransUnet [12] fits this scenario perfectly. We adapt the TransUnet structure, use pretrained ImageNet weights for the transformer block and changed the number of transformers from 12 to 6 so that it can be done with our current computing resources. The structure of this model is shown in Figure 5.

5.2 Algorithmic Implementation

The algorithm for flare removal involves a comprehensive pipeline that integrates data preparation, model training, and evaluation. Input data is prepared by downsampling a large dataset of flare-free images (27,449 images) to a resolution of 224 pixels, alongside a separate downsampling of lens flare images (5,001 images) to dimensions of 353×263 pixels. Test datasets, comprising real (20 images) and synthetic (37 images) scenes, are also downsampled to ensure compatibility with the training resolution. The training employed U-Net, U-Net++, U-Net3+, ViT and TransUnet on semi-synthetic flare-corrupted images generated by compositing flare-free images with flare overlays. The model is configured to predict a flare-free scene, which inherently risks removing the light source. To address this, a post-processing step blends the predicted flare-free scene with the original flare input, ensuring that the light source is retained in the final output. Model training spans 150 epochs, using designated directories for input data, flare overlays, and log storage. The evaluation phase utilizes separate scripts to assess the model on real and synthetic test datasets, comparing predictions against ground truth images. Finally, qualitative and quantitative results are generated by testing the trained model on held-out datasets, producing both visual outputs and evaluation metrics (PSNR and SSIM). This three-step framework—input generation, CNN/Transformer-based prediction, and post-processing—effectively removes lens flare artifacts while preserving the natural light source, ensuring high-quality, artifact-free image as shown in Figure 6.

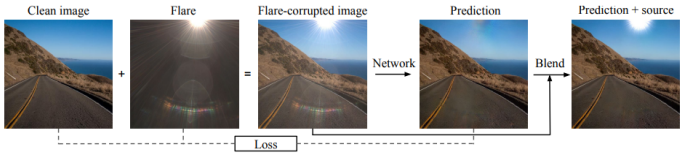


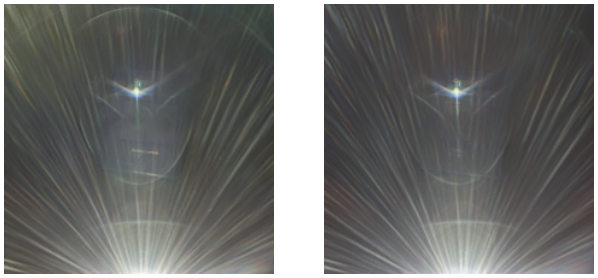
Fig. 6: All of our experiments follow the same three steps framework: 1) Generate training input by randomly compositing a flare-free natural image and a flare image. 2) A CNN is trained to recover the flare-free scene (in which the light source may also have been removed, which is undesirable). 3) After prediction, Blend the input light source back into the output image.

6 Experimental Results

Method	Synthetic		Real	
	PSNR	SSIM	PSNR	SSIM
U-Net (Baseline)	33.88	0.9309	**30.25**	**0.9142**
U-Net++	34.83	0.9286	30.21	0.9119
U-Net3+	33.38	0.9246	29.96	0.9140
ViT	32.47	0.8931	29.64	0.8432
TransUnet	**35.67**	**0.9358**	29.82	0.8972

TABLE 1: Quantitative comparison with related methods on synthetic and real data.

To see how all the models are performed, we use both synthetic and real images. Table 1 and Table 2 presents the quantitative and qualitative performance comparison respectively, using Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index Measure (SSIM) as evaluation metrics. We evaluate the performance over the test set with respect to the ground truth images.



(a) U-Net segmented flare

(b) TransUnet segmented flare

Fig. 7: Qualitative comparison of UNet and TransUNet segmented flare.

On the synthetic dataset, TransUnet achieved the highest performance, with a PSNR of 35.67 and SSIM of 0.9358, outperforming the baseline U-Net by notable margins of 1.79 dB in PSNR and 0.0059 in SSIM. This improvement can be attributed to TransUnet's hybrid architecture, which

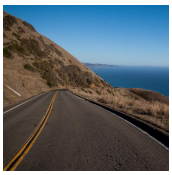
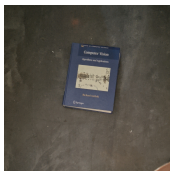
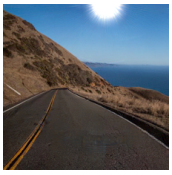
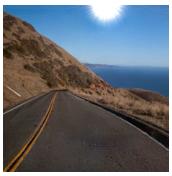
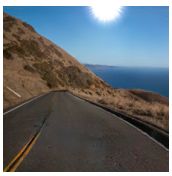
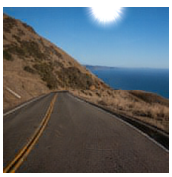
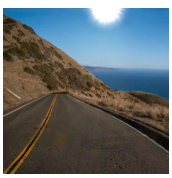
Ground Truth			
Input Image			
U-Net	 PSNR: 32.23 SSIM: 0.9283	 PSNR: 29.02 SSIM: 0.8332	 PSNR: 33.84 SSIM: 0.9731
U-Net++	 PSNR: 31.63 SSIM: 0.9300	 PSNR: 29.42 SSIM: 0.8073	 PSNR: 33.29 SSIM: 0.9663
U-Net3+	 PSNR: 31.08 SSIM: 0.9158	 PSNR: 28.66 SSIM: 0.8537	 PSNR: 32.22 SSIM: 0.9661
ViT	 PSNR: 32.28 SSIM: 0.8672	 PSNR: 28.67 SSIM: 0.7554	 PSNR: 33.92 SSIM: 0.9432
Trans-Unet	 PSNR: 33.19 SSIM: 0.9339	 PSNR: 28.65 SSIM: 0.8174	 PSNR: 32.55 SSIM: 0.9710

TABLE 2: Visual comparison of models evaluated on synthetic and real scene images.

effectively integrates convolutional layers for local feature extraction and transformers for capturing global dependencies. U-Net++, with its densely connected skip connections, followed with a PSNR of 34.83 and SSIM of 0.9286, indicating its enhanced capacity to model complex spatial relationships. The performance of U-Net3+ (PSNR: 33.38, SSIM: 0.9246) suggests diminishing returns from additional skip connections compared to U-Net++. ViT, with a PSNR of 32.47 and SSIM of 0.8931, underperformed, likely due to its reliance on transformers alone, which may lack sufficient inductive bias for spatially constrained tasks like image reconstruction.

On the real dataset, U-Net (Baseline) showed competitive performance, achieving a PSNR of 30.25 and SSIM of 0.9142, marginally surpassing U-Net++ and U-Net3+. TransUnet maintained strong performance with a PSNR of 29.82, though its SSIM dropped to 0.8972, suggesting possible overfitting to synthetic data characteristics. The ViT model exhibited the lowest SSIM (0.8432), highlighting its struggle to generalize to real-world variations due to the lack of convolutional inductive priors.

These results suggest that architectures incorporating both local and global feature extraction mechanisms (e.g., TransUnet) excel on synthetic data where patterns are well-structured, but their generalization to real data may be constrained by overfitting. In contrast, convolution-dominant architectures like U-Net variants demonstrate consistent performance across both datasets due to their robust spatial feature learning. The trade-off between global context modeling and spatial inductive biases highlights the importance of balancing these factors for diverse data domains.

We believe the reason is the following: (1) Usage of a pretrained transformer model based on the task of image classification. Even though transfomers are good transfer-learners, under the limited dataset, they are not able to generalize well. (2) Comparing the predicted flare image in Figure 7, we can see that transformer models try to only capture the possible flares but CNN models have captured more general scene information. This might result from lack of local inductive bias in transformers. Attention mechanism makes it focus on global justification, so it tends to omit the region where flare intensity is low and only learns from stronger intensity region generally closer to the light source. (3) Transformer based model generally have greater number of parameters than CNNs, which in addition to the relat ively small flare dataset, might have overfit the global pattern of flare dataset as emphasized in transformer block with the inability to generalize well to real scene data.

7 Conclusion

In this paper, we explore the performance of CNN- and transformer-based models for lens flare removal. CNN-based models demonstrated excellent performance in capturing local spatial features, enabling robust generalization even with a relatively small dataset. However, their reliance on local context limited their capacity to fully capture global scene information, occasionally impacting and modifying scenes.

On the other hand, transformer-based models leveraged their global attention mechanisms to achieve more accurate

flare prediction while better preserving background color intensity. This was particularly evident in synthetic datasets, where structured flare patterns were prevalent. Despite this advantage, transformers struggled with real-world images, likely due to overfitting on the limited synthetic training data and a lack of strong local inductive bias. Their higher parameter count and dependence on data-intensive training data poses a limitation on their performance.

Given their demonstrated potential in leveraging global context and preserving scene integrity, despite the use of downsampled inputs (which may have impacted model performance), transformers present a promising avenue which we aim to further explore in the domain of flare removal. Additionally, investigating hybrid architectures that integrate the local feature extraction strengths of CNNs with the global attention capabilities of transformers could lead to more versatile and robust solutions.

REFERENCES

[1] C. S. Asha, S. K. Bhat, D. Nayak, and C. Bhat, "Auto removal of bright spot from images captured against flashing light source," *2019 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, pp. 1–6, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:211689922

[2] F. Chabert, "Automated lens flare removal," 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:16064440

[3] P. Vitoria and C. Ballester, "Automatic flare spot artifact detection and removal in photographs," *Journal of Mathematical Imaging and Vision*, vol. 61, pp. 515 – 533, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:254653751

[4] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on Image Processing*, vol. 13, pp. 1200–1212, 2004. [Online]. Available: https://api.semanticscholar.org/CorpusID:757375

[5] E.-V. Talvala, A. Adams, M. Horowitz, and M. Levoy, "Veiling glare in high dynamic range imaging," *ACM Trans. Graph.*, vol. 26, no. 3, p. 37–es, Jul. 2007. [Online]. Available: https://doi.org/10.1145/1276377.1276424

[6] R. Raskar, A. K. Agrawal, C. A. Wilson, and A. Veeraraghavan, "Glare aware photography: 4d ray sampling for reducing glare effects of camera lenses," *ACM SIGGRAPH 2008 papers*, 2008. [Online]. Available: https://api.semanticscholar.org/CorpusID:763304

[7] Y. Wu, Q. He, T. Xue, R. Garg, J. Chen, A. Veeraraghavan, and J. T. Barron, "How to train neural networks for flare removal," *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 2219–2227, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:238531781

[8] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *ArXiv*, vol. abs/1505.04597, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:3719281

[9] Z. Zhou, M. M. Rahman Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support*, D. Stoyanov, Z. Taylor, G. Carneiro, T. Syeda-Mahmood, A. Martel, L. Maier-Hein, J. M. R. Tavares, A. Bradley, J. P. Papa, V. Belagiannis, J. C. Nascimento, Z. Lu, S. Conjeti, M. Moradi, H. Greenspan, and A. Madabhushi, Eds. Cham: Springer International Publishing, 2018, pp. 3–11.

[10] H. Huang, L. Lin, R. Tong, H. Hu, Q. Zhang, Y. Iwamoto, X. Han, Y. Chen, and J. Wu, "Unet 3+: A full-scale connected unet for medical image segmentation," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1055–1059, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:215828394

[11] A. Dosovitskiy, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[12] J. Chen, Y. Lu, Q. Yu, X. Luo, E. Adeli, Y. Wang, L. Lu, A. L. Yuille, and Y. Zhou, "Transunet: Transformers make strong encoders for medical image segmentation," *ArXiv*, vol. abs/2102.04306, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:231847326