Computer Vision for Deblurring and Enhancing Image or Video Data

Anurag Bapat abapat@usc.edu
Jake Treska treska@usc.edu
Swaminathan Chellappa schellap@usc.edu
Yongchen Lin linyong@usc.edu
Monika Yadav monikaya@usc.edu

Abstract

The rise of advanced computer vision models is revolutionizing the world of medicine, security, surveillance, autonomous vehicles, as well as numerous other fields that focus on analyzing real-world image data. Despite recent advancements, computer vision models are still dependent on clear images needed to achieve high accuracy during inference and, if given blurred images, can result in significant drops in performance. While many models exist to deblur these images, most contain overly complicated architectures making real-time inference impossible, which is required for many current computer vision applications. To address this issue, we have implemented three types of deblurring models in increasing size and complexity, that are able to accurately deblur image and video data while allowing the speed needed for real-time inference. Our models strike a balance between deburring accuracy and speed allowing them to be used to enhance images for a variety of real-time computer vision tasks. This paper will further discuss our results, and how they compare to current pre-built models in both accuracy and inference speed.

1 Introduction

This report addresses the challenge of blind deblurring for images and video. Blur may be uniform or non-uniform, depending on factors such as camera shake, fast-moving subjects, or focus inaccuracies. This degradation leads to a loss of detail and information that is crucial for applications like forensic or crime-scene analysis, obstacle detection in self-driving vehicles, and other computer-vision tasks.

The goal is to restore a blurred image or video frame to its sharpest state by reversing the degradation incurred during capture. Blurriness is typically modeled as the convolution of a sharp image with a blur kernel, plus additive noise. If the kernel is known, one can apply non-blind deconvolution; however, in most real-world situations the kernel is unknown, which greatly increases problem complexity. Moreover, blur may vary spatially, certain regions can be more blurred than others, or the entire frame may suffer uniform blur. These uncertainties and high computational demands make deblurring a difficult problem, despite decades of research.

Early solutions relied on signal-processing principles and handcrafted priors, for example, methods that estimate a blur kernel and then apply mathematical regularization to restore the image. While effective on controlled inputs, these approaches become computationally intensive on real-world images exhibiting spatially varying blur and noise. Deep learning has since transformed the field: modern neural networks can learn a direct mapping from a blurry input to its sharp counterpart, eliminating the need to estimate the blur kernel explicitly. Trained on abundant data, these models achieve substantially sharper and more realistic restorations than earlier methods, even in dynamic scenes.

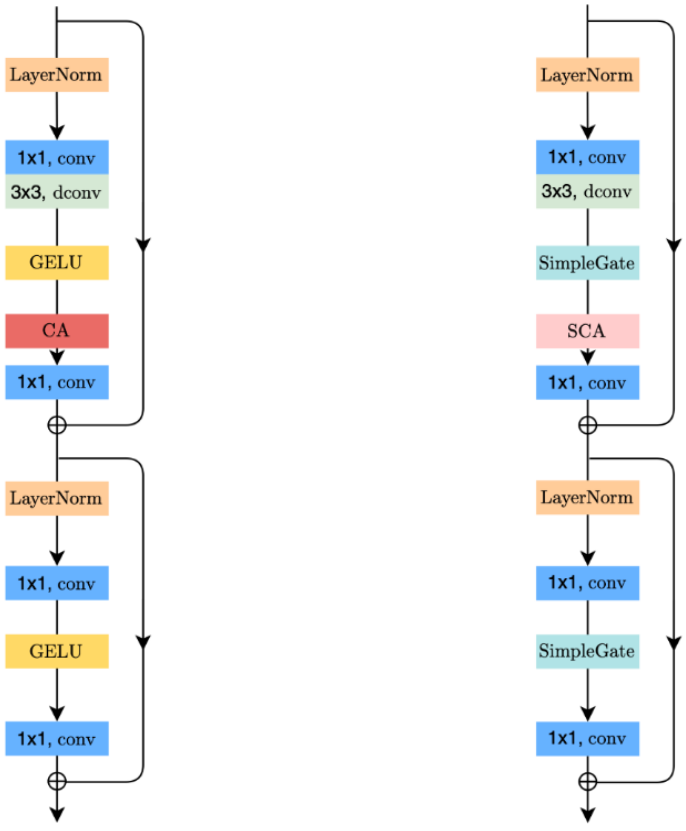
State-of-the-art approaches range from generative adversarial frameworks (Kupyn et al. [2019]) to transformer-based architectures and convolutional neural networks (Chen et al. [2022]). Although highly effective, they are bulky and inefficient: with over 20 million parameters, they demand significant time for both training and inference. Due to slow inference speeds and large memory footprints, these models cannot run on mid-range GPUs, let alone edge devices such as mobile phones, and they fail to provide real-time processing.

In this paper we propose a lightweight deblurring architecture whose smallest variant has just 1.7 million parameters. By carefully balancing representational capacity and complexity, our design dramatically reduces inference time while maintaining high-quality restoration. This parameter-efficient method delivers near real-time performance on mid-range GPUs and enables on-device deblurring for edge devices without compromising image quality.

2 Existing work

2.1 NAFNet: Nonlinear Activation Free Network

In their paper, Chen et al. [2022] describe a network architecture for image denoising and deblurring, aiming to find an architecture with low inter-block and intra-block complexities that can achieve state-of-the-art (SOTA) performance. We were primarily interested in the part describing deblurring. They propose a baseline that does better than the SOTA methods and simplify it by replacing the activation functions (NAFNet), which, again, performs better than the SOTA methods. Hence, it is called Nonlinear Activation Free Network.



(a) Baseline intra-block architecture

(b) NAFNet intra-block architecture

Figure 1: Comparison between baseline and NAFNet intra-block architectures. Images taken from Chen et al. [2022]

They divide the system into two areas of focus: inter-block complexity and intra-block complexity. They use UNet as the architecture that reduces inter-block complexity.

Gated Linear Units (GLUs) are effective in computer vision tasks. They use GELU (Gaussian error linear unit), a special case of GLU in their intra-block architecture, which improves performance. Additionally, while building NAFNet from their baseline, they replace the nonlinear activation functions in GLU with an element-wise product of feature maps (Refer to Figure 1). The logic behind this is that this product of two linear transformations generates nonlinearity that can replace the activation function, thus simplifying the architecture. This indicates that nonlinear activation functions may not be necessary for SOTA computer vision methods.

We explored NAFNet for our purposes. However, we did not get desirable results concerning our objective of real-time video deblurring. NAFNet's inference is slow and could only give us a frame rate of <1 FPS for one of their prebuilt models, NAFNet-GoPro-width32, although it gave us a good PSNR of ~33 dB and SSIM of >0.9. We tried to reduce the size of the model by reducing the number of channels and the encoder block size. As expected, the results degraded, leading to a PSNR of ~15 dB and SSIM of ~0.45 while the inference rate only increased to 2-2.5 frames per second.

2.2 DeblurGAN-V2

Kupyn et al. [2019] present a successor to the original DeblurGAN-v2 as a new Generative Adversarial Network (GAN). DeblurGAN-v2 works with a wide range of backbones allowing us to adjust the architecture for performance or efficiency.

A GAN consists of two components: a generator and a discriminator. In the context of image deblurring, the generator produces artificial deblurred images and asks the discriminator whether it thinks the fed image is artificial or real. These two components form a two-player minimax game. DeblurGAN-v2 uses the Relativistic GAN, which estimates the probability that some given data is more realistic than the fake data. DeblurGAN-v2 implements a feature pyramid network (FPN) in the generator, which improves the generator's ability to handle spatially varying blur. Traditional multi-scale pyramids relied on forward passes for each scaled version of the input image, which is inefficient in terms of time and computation. An FPN, on the other hand, computes multi-scale features from a single input image using a backbone (explained below), which is more computationally efficient and significantly lowers inference time.

The FPN-embedded architecture is backbone agnostic. The feature extractor network is plugged into the generator and acts as the backbone of DeblurGAN-v2. It can be thought of as the entry point for images to be processed. This gives us the flexibility to tweak the architecture based on our needs. By default, DeblurGAN-v2 uses ImageNet-pre-trained backbones like Inception-ResNet-v2 for strong deblurring performance and MobileNet V2 for efficiency. DeblurGAN-v2 with MobileNet-DSC indicates the possibility of real-time video deblurring.

MobileNet-DSC refers to a configuration of the DeblurGAN model which utilizes the MobileNet v2 backbone and replaces standard convolutions with depth-wise separable convolutions (DSC) throughout the whole network. The result is a deblurring model suitable for mobile or on-device deblurring. Notably, MobileNet v2 achieved comparable performance to other state-of-the-art methods on the GoPro dataset in terms of SSIM scores while being up to 100x faster. During this evaluation, it was shown to have an inference time of 0.04 seconds per image, making it suitable for real-time video deblurring at 25 fps.

2.3 Restormer: Efficient Transformer for High-Resolution Image Restoration

Transformer based models have continued to revolutionize image deblurring tasks due to their natural advantage over CNN based architectures that struggle with mapping long range dependencies. While transformers are able to easily address this problem with multi-headed attention, the tradeoff is its computational complexity that grows quadratically in regards to the spatial resolution. Zamir et al. [2022] addresses this problem by implementing a restoration transformer, restormer, that achieved state-of-the-art accuracy while mitigating the computational overhead present in modern transformer based architectures. The Restomer, which represents the architecture present in figure x, introduces two key changes leading to a significant reduction in computational complexity: Multi-Dconv Head transposed Attention, and Gated-Dconv Feed-Forward Networks.

Unlike conventional self-attention (SA) that grows quadratically based on spatial resolution, Multi-Dconv Head transposed Attention applies SA across the channels rather than spatial dimension leading to a linear time complexity. With regards to Gated-Dconv Feed-Forward Networks, unlike normal FFNs that apply two 1x1 convolutions to expand and reduce feature channels, this architecture implements gating mechanisms and depthwise convolutions. The gating mechanism is the product of two parallel linear transformation paths, while the depth-wise convolutions learn local image structure by encoding neighboring pixel positions.

Varying image resolutions and the convention of training a CNN model on fixed-size images can lead to training on cropped image sections (to maintain resolution) which may not perform well on full-resolution during inference. To avoid this issue, the model is trained starting with smaller image patches in the early epochs, progressively increasing the patch size (and hence, resolution) in the later epochs. This ensures that the model learns to run inference on varying image resolutions.

Overall, Restormer promises amazing results as it rivals other state-of-the-art deblurring models such as NAFNet and other transformers with PSNR scores between 30-35dB and SSIM scores between 0.9 and 0.95.

2.4 Multi-Temporal Recurrent Neural Network (MT-RNN) for progressive non-uniform single image deblurring

Park et al. [2020] propose a multi-temporal (MT) approach that progressively refines blur removal over multiple iterations, prioritizing stepwise refinement over speed, despite its low computational cost. Park et al. [2020] used an incremental temporal training strategy, where blurred images are reconstructed step by step from severe to mild blur before reaching the final sharp output. The MT-RNN utilizes recurrent feature maps, allowing the network to retain information across iterations, making it more effective at handling non-uniform and large motion blur. This method outperforms DeblurGAN and state-of-the-art MS-based deblurring techniques on the GoPro dataset, achieving superior PSNR with fewer parameters, demonstrating its efficiency in handling real-world motion blur.

2.5 Building on Vision Transformers

Liang et al. [2024] introduces a new approach to image deblurring by leveraging pre-trained Vision Transformers (ViTs) Dosovitskiy et al. [2020] instead of traditional CNN-based feature extraction methods like VGG (also used in DeblurGAN-v2). The key idea is to leverage pre-trained transformer model, specifically mask autoencoders (MAE), to capture rich, high-resolution features that are more sensitive to blur. Liang et al. [2024] improves deblurring by introducing two perceptual losses: one that directly compares feature differences (local loss) and another that aligns feature distributions using Wasserstein distance (global loss). This approach helps recover sharper, more realistic images without sacrificing standard quality metrics like PSNR, and performs fast inference as compared to MT-RNN, making it better-suited for real-time applications. Compared to existing methods like Uformer, Restormer, and NAFNet, it achieves better perceptual quality while maintaining strong numerical performance on metrics.

3 Preprocessing and Model Architecture

3.1 Preprocessing Techniques

The images used to train the model contain inherent noise that would pose a threat to the features learned by a model and in turn affect the overall performance of the model on the deblurring task. The optimizer could perceive more variance in the loss landscape and may take longer to settle on good weights. The random pixel fluctuations caused by the noise present in the image may cause early convolutional filters to learn spurious patterns rather than meaningful features.

In order to denoise the frames, we first compute the image's spatial gradients in both the horizontal and vertical directions using a Sobel operator. By reducing low-amplitude fluctuations that are characteristic of random noise, the Sobel kernel efficiently functions as a high-pass filter, emphasizing real edges, where pixel intensities change abruptly. By taking the magnitude of these gradients, spurious fluctuations are attenuated and only the salient structural information is retained, yielding a

cleaner edge map for downstream processing. We then normalize this gradient magnitude back to an 8-bit range, which both preserves important edge details and filters out residual noise before feeding the frames into our deblurring model.

3.2 Loss functions

SSIM loss: The structural similarity and perceived difference in picture quality between the output deblurred image and the ground truth sharp image are measured using the Structural Similarity Index Measure (SSIM) loss. Over small windows, SSIM directly assesses covariance, brightness, and local contrast. We add SSIM to our loss with a configurable weight after computing it over a Gaussian-weighted patch for every frame. This motivates the model to output deblurred images that preserve the consistent visual patterns necessary for deblurring in addition to matching pixel values.

L1 loss: To achieve precise pixel-level reconstruction, we utilize an L1 loss term, which reduces the average absolute difference between our deblurred output and the sharp image from the ground truth. In order to prevent excessively smooth outcomes, L1 promotes a sparser, sharper error distribution and is less sensitive to huge outliers because it penalizes all deviations linearly.

By weighting and summing these two losses, we leverage SSIM's structural sensitivity and L1's intensity accuracy in tandem, yielding deblurred images and video frames that are both visually sharp and quantitatively faithful.

Laplacian score: Filter based on second-order derivation that is able to capture regions with sharp intensity changes and commonly used to detect object edges. By monitoring the Laplacian scores throughout training we are able to monitor and determine an increase in edge sharpness, highlighting progress in image deblurring.

3.3 Model Architecture

To ensure real-time inference across all devices, regardless of their computational resources/limitations, we developed three models of increasing size and complexity. Model one contains 1.7 Million parameters, Model 2 contains 5 Million, and Model 3 contains 14 Million. All models contain a similar architecture inspired by DeepDeblur (Nah et al. [2017a]) that implements a multi-scale convolutional neural network processing the image at ¼, ½, and full resolution. While all models contained a similar architecture, each varied slightly in the number of residual blocks present at each scale. Model 1 contained 6 at each scale, Model 2 contained 4-10 at each scale, and Model 3 contained 16.

4 Results

Table 1: Model Results

Model	**N/A (Blur Image)**	**Small**	**Medium**	**Large**	**NAFNet**	**(Clear Image)**
Size (parameters)	N/A	1.7 Million	5 Million	14 Million	68.7 Million	N/A
SSIM (avg)	77.45	78.97 (↑ 1.52%)	80.36 (↑ 2.91%)	79.44 (↑ 1.99%)	91.50 (↑ 14.05%)	N/A
Laplacian (avg)	1709	2359 (↑ 31%)	2006 (↑ 17.3%)	1905 (↑ 11.4%)	3231 (↑ 189%)	4962
FPS	N/A	100+ (A100, 3080, T4, L4)	35+ (A100, 3080, T4, L4)	20+ (A100, 3080)	1–2 (T4, L4)	N/A

4.1 Realistic and Dynamic Scenes (REDS)

After training all the models on the REDS dataset, their results are depicted in figure 2. Compared to the original blurred image, all models were able to achieve an increase in both the Laplacian and SSIM scores. Model 1 had a 31% increase in Laplacian and 1.3% in SSIM, while the other two models had a lower percentage increase in Laplacian (17.3% and 11.4%), but a 1-2% higher increase in SSIM. These results show that model 1 was able to better detect and highlight edges, higher Laplacian score, but started to lack the complexity to solve more complex blurs as indicated by a lower SSIM compared to Model 2 and 3. Additionally, Table 1 also displays the results for a state

of the art NAFNet model achieving Laplacian scores increasing by 189% and a SSIM increase of 14.05%. While this model delivers impressive results, its main limitation is its complex architecture, 68.7 Million parameters, ultimately making real-time inference impossible. On the other hand, due to our model's significantly smaller architecture, we were able to achieve real time-inference (30+ FPS) across all GPUs, and even upwards of 100+ FPS with Model 1.

Figure 2: Canny Filters (Blurred Image, Model 1, Model 2, Model 3, Clear Image)

Figure 2 contains five columns where each column displays a model's deblurred image output above its corresponding canny filter displaying all edges. When analyzing the canny filters across all columns, you can clearly see the leftmost image contains the least amount of edges, while the remaining columns contain more and sharper edges with our trained models being columns 2-4 and the ground truth clear image being column 5.

4.2 Drone Fine-tuning

Figure 3: Fine-tuning results

While our models showed promising results on the REDS dataset, our goal was to initially train models capable of real-time inference on drone/aerial data. Therefore, Model 1 was selected and fine-tuned on a portion of the VisDrone dataset (Kupyn et al. [2019]). The VisDrone dataset contained the aerial footage we desired, but artificial blur was added to the data for training. The results are

depicted in figure 3 with our model achieving an average Laplacian score around 850, with the blurred image being around 30 and the original image around 1600.

5 Future work

Although the current work provides a solid foundation in real-time image deblurring, there are several directions in which future research and development can be extended to improve model performance. In particular, we aim to refine the preprocessing stage of the pipeline, which plays a critical role in preparing input data for optimal deblurring. Excessive noise in input images can be especially problematic, as deblurring models are typically not trained to handle noise and blur simultaneously, which may cause the model to hallucinate. Further considerations are also required when selecting a noise reduction technique, as it is important to ensure that edges and other salient features are preserved during the denoising process to enable accurate reconstruction of the sharp image. This issue has compelled us to explore several approaches to noise reduction during the preprocessing stage.

Perhaps the most straightforward approach to noise reduction is the application of a filter to the input image. The advantage of this approach is that processing times are fast, and the hyperparameters of the filter can be easily tuned to achieve optimal, model-specific performance. We have explored two different filters for this use case: the bilateral filter and the non-local means filter. The bilateral filter reduces noise by averaging local pixels while preserving edges by considering both spatial proximity and color similarity. It smooths regions of similar intensity while avoiding blurring across edges where intensity changes rapidly. In this way, it effectively reduces noise while keeping image details intact.

The non-local means filter reduces noise by averaging image patches across the entire image. It searches for similar patches and computes weighted averages based on patch similarity. This allows the filter to incorporate long-range dependencies during denoising, making it potentially more effective at preserving edges and image features than the bilateral filter.

Total variation (TV) denoising is a more involved approach to noise reduction that aims to minimize the total variation within the image. Essentially, it penalizes rapid intensity changes while preserving edges. TV denoising is an optimization-based method, in contrast to the filtering approaches, and has the potential to preserve features in images more effectively than filters. Unfortunately, TV denoising is also more computationally intensive, which may make it impractical for real-time applications.

From our preliminary tests on the GoPro dataset [Nah et al., 2017b], the average processing time per image was 0.025, 0.8628, and 2.2559 seconds for the bilateral filter, non-local means filter, and TV denoising, respectively (see Table 2). These processing times were measured on a system with an Intel Core i9-12900H CPU (14 cores, 20 threads) and 32 GB RAM, without GPU acceleration.

From these results, we conclude that the bilateral filter could be readily implemented as a preprocessing step in our real-time model, while further optimizations would be necessary for the non-local means filter and TV denoising to be viable options. Further work is needed to determine whether the inclusion of the bilateral filter improves model performance and to identify optimal, model-specific filter parameters.

Table 2: Average processing time per image measured on the GoPro dataset [Nah et al., 2017b].

Denoising Method	**Average Time (s)**
Bilateral Filter	0.0250
Non-local Means Filter	0.8628
TV Denoising	2.2559

Implementing selective deblurring is another promising direction for enhancing the model's real-time applicability. This approach involves determining whether a frame requires deblurring during the preprocessing stage, prior to performing inference. A metric such as Laplacian variance can be used to quantify the blurriness of a frame, and thresholds would be defined to determine whether the frame needs to be deblurred. This allows the model to bypass unnecessary inference on already sharp images, thus reducing computational load and improving overall processing speed (FPS).

Furthermore, we would like to explore the possibility of implementing an object detection module at the end of our pipeline for extension to more real-world applications. Although model outputs frequently appeared visually similar compared to their blurry counterparts, they often performed much better when used with object detection models. Preliminary testing with the YOLO object detection model [Redmon et al., 2016] showed consistently higher object detection accuracy on our output images compared to the original blurry inputs. Figure 4 shows a comparison of YOLO outputs on a blurry input image and its corresponding sharp output. In this example, YOLO was able to accurately detect 70 more objects in the sharp output compared to the blurry input. These results are promising, suggesting that integrating an object detention model into our pipeline may improve the real-world computer vision applications of our system.

Figure 4: Yolo comparison on input(left) and output(right) images.

6 Conclusion

In this paper, we presented a lightweight deblurring architecture designed to meet the real-time performance demands of modern applications while maintaining competitive image restoration quality. Current state-of-the-art models often achieve high deblurring performance at the cost of excessive computational overhead. To address these limitations, we proposed a series of compact models that strike a balance between visual quality and inference speed, making them suitable for deployment in latency-sensitive environments. We reviewed two existing state-of-the-art approaches to deblurring, namely the transformer-based and GAN-based models NAFNet and DeblurGAN-v2. While these models deliver impressive deblurring results, we highlighted their limitations in terms of model size and inference speed.

In order to support real-time inference across a range of hardware constraints, we developed three progressively larger models, with our smallest model containing only 1.7 million parameters. These are based on a multi-scale convolutional architecture inspired by DeepDeblur. Evaluation on the REDS dataset showed that all three models achieved significant improvements in terms of sharpness and structural similarity over the blurry inputs. While our smallest model excelled in edge enhancement, the larger models performed better in terms of overall reconstruction quality. Most importantly, despite their small size, all models were able to achieve real-time inference speeds across a range of GPUs, outperforming heavier models like NAFNet in terms of efficiency while maintaining competitive deblurring performance.

To demonstrate the real-world applicability of our models, we fine-tuned the smallest model on drone footage from the VisDrone dataset. The model was able to maintain strong deblurring performance, achieving noticeable improvements in image sharpness while preserving its real-time inference speeds.

Finally, we discussed several promising avenues for future research and development that could enhance the model's robustness and applicability in real-world environments. We have explored several approaches to noise-reduction in input images during the pre-processing stage and presented preliminary results from processing the GoPro dataset. From these results, the bilateral filter was identified as a strong candidate for integration into our model. We also discussed the potential for implementing selective deblurring to improve our model's inference speed as well as integrating an object detection model for real-world use cases.

References

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

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2019.

Pengwei Liang, Junjun Jiang, Xianming Liu, and Jiayi Ma. Image deblurring by exploring in-depth properties of transformer. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017a.

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 (CVPR)*, pages 3883–3891, 2017b.

Dongwon Park, Dong Un Kang, Jisoo Kim, and Se Young Chun. Multi-temporal recurrent neural networks for progressive non-uniform single image deblurring with incremental temporal training. In *European conference on computer vision*, pages 327–343. Springer, 2020.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, pages 779–788, 2016.

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 *CVPR*, 2022.