Journal of Computer Science

Research Article

CUDA-Powered EDSR x4: Super-Resolution for Real-Time Video Enhancement

Ganeshayya Shidaganti, Shubeeksh Kumaran and Vishwachetan D

Department of Computer Science, Ramaiah Institute of Technology, Bangalore, Karnataka, India

Article history
Received: 27-01-2025
Revised: 29-03-2025
Accepted: 08-04-2025

Corresponding Author:
Ganeshayya Shidaganti
Department of Computer Science,
Ramaiah Institute of Technology,
Bangalore, Karnataka, India
Email:
ganeshayyashidaganti@msrit.edu

Abstract: Super-Resolution enhances image or video quality by upscaling frames to higher resolutions, which is essential for applications like investigative analysis demanding higher quality. However, this process is resource-intensive. GPUs, with their thousands of CUDA cores, offer significant parallel processing advantages over CPUs, enabling faster performance. This paper presents an optimized approach for real-time video enhancement using the EDSR x4 model with Nvidia-CuDNN acceleration. We employ dynamic CPU-GPU load balancing to distribute computational tasks based on resource availability, reducing processing time by 18% and achieving real-time upscaling with a processing time of 205 ms per 10 frames. EDSR, originally designed for Single Image Super-Resolution (SISR), is chosen over Video Super-Resolution (VSR) methods due to its superior frame-level clarity, making it ideal for scenarios where individual frame quality is critical. A notable discovery is the wave-like behavior in normalized PSNR, SSIM, and VIF metrics across resolutions, revealing a periodic relationship between resolution and perceived quality. This insight further informs optimal resolution selection for various applications. The proposed system efficiently handles 480p to 4K video, maintaining high image quality and GPU utilization between 60%-80%, making it suitable for real-time applications that require both speed and high fidelity.

Keywords: EDSR x4, Dynamic Load Balancing, Real-Time Video Upscaling, Super-Resolution, CUDA

Introduction

In the modern world, where the demand for high-quality videos is rapidly increasing, the demand for high-quality video upscaling has significantly increased. This has led to the development of significant technologies like deep learning models, with the Enhanced Deep Super-Resolution (EDSR) x4 model emerging as a potent tool for achieving remarkable image quality enhancement (Kim *et al.*, 2016). Despite the exceptional up-scaling capabilities of such models, the challenge of attaining real-time super-resolution persists due to processing limitations, making it necessary for approaches that leverage cutting-edge technologies to solve this issue.

This paper embarks on a journey of optimization and acceleration by delving into the realm of GPU technology and CUDA optimization to bolster EDSR x4 upscaling to enable real-time video quality enhancement. Recent advancements in deep learning techniques for super-resolution, as highlighted in "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network" (Ledig *et al.*, 2017) and "FastDVDnet: Towards Real-Time Video Denoising Without Explicit Motion Estimation" (Agustsson & Timofte, 2017), serve as pivotal inspirations. These works unravel the intricacies of deep learning architectures and GPU optimization, laying the groundwork for our exploration into real-time video enhancement methodologies.

Furthermore, our approach draws insights from seminal works such as "Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution" (Dong *et al.*, 2016b), "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network" (Shi *et al.*, 2016), "Accelerating the Super-Resolution Convolutional Neural Network" (Dong *et al.*, 2016a), and "Image Super-Resolution Using Deep Convolutional Networks" (Zhang *et al.*, 2018a). These papers contribute essential perspectives on different aspects of super-resolution, encompassing network architectures, real-time processing, and acceleration techniques.

As we confront the computational challenges associated with super-resolution, we draw insights from "Deep Residual Learning for Image Recognition" (He *et* *al.*, 2016), which introduces adaptive super-resolution techniques dynamically adjusting the upscaling process based on input frame complexity. Simultaneously, "Dynamic Load Balancing on Heterogeneous Systems" (Xiang *et al.*, 2022) offers a glimpse into innovative load-balancing strategies in parallel processing environments, a concept central to our methodology.

Guided by the adaptive strategies outlined by (He *et al.*, 2016) and the load-balancing insights from (Xiang *et al.*, 2022), our approach unfolds with a focus on achieving optimal resource utilization. Inspired by the divide-and-conquer methodology proposed by (Tovar & Kwon, 2021), a strategic technique that pre-emptively splits the input frameset, akin to data-prefetching in Multi-Core Architecture methodologies, we aim to strike a balance between CPU and GPU resources. This methodology not only optimizes resource utilization but also ensures a real-time output without a buffered wait lag in video delivery.

In essence, this research positions itself at the intersection of deep learning, GPU acceleration, load-balancing strategies, and a comprehensive understanding of super-resolution methodologies drawn from a rich tapestry of foundational papers.

Related Work

Super-resolution (SR) is a critical technique in image and video processing aimed at enhancing the resolution of visual content to improve detail and overall quality. The field has seen significant advancements, particularly with the advent of deep learning models like EDSR (Enhanced Deep Residual Networks), which have set new benchmarks in Single Image Super-Resolution (SISR). Video Super-Resolution (VSR) builds on these advancements but introduces additional challenges, such as maintaining temporal consistency across frames. Furthermore, the integration of CUDA (Compute Unified Device Architecture) has revolutionized the implementation of these models, enabling real-time processing capabilities. This literature review examines the key developments in SISR and VSR, the rationale for choosing EDSR for video frame upscaling, and the role of CUDA in optimizing these processes.

Single Image Super-Resolution (SISR)

SISR has evolved significantly from traditional interpolation methods to sophisticated deep-learning models. Early methods, such as bicubic interpolation, often resulted in loss of detail and smoothness, particularly at high magnification levels. The introduction of deep learning models like SRCNN and later EDSR marked a significant leap in SISR quality.

Enhanced Deep Residual Networks (EDSR), introduced by Lim *et al.*, is a high-performance model that improves upon earlier approaches like SRCNN by removing batch normalization layers, expanding the model's capacity, and stabilizing training, achieving state-of-the-art performance with PSNR values of 26.96 dB on the Ur-ban100 dataset for 4× upscaling (Lim *et al.*, 2017). EDSR leverages deep residual blocks without batch normalization, enabling it to handle large-scale images effectively. Ahn *et al.* explored compressing deep models, introducing techniques that maintain high PSNR and SSIM values while reducing the model size by 56%, achieving a PSNR of 27.24 dB on Urban100 (Ahn *et al.*, 2018). Zhang *et al.* built on EDSR with Residual Dense Networks (RDN), incorporating dense connections within residual blocks to improve feature reuse and gradient flow, resulting in a PSNR of 32.47 dB on the DIV2K dataset for 2× upscaling, outperforming EDSR by 0.2 dB (Zhang *et al.*, 2018b). Additionally, Zhou *et al.* explored adaptive dictionary learning, where CUDA acceleration led to a speedup of up to 30× compared to CPU implementations, enabling real-time processing of multispectral images (Barman *et al.*, 2021).

Video Super-Resolution (VSR)

VSR extends the principles of SISR to video sequences, introducing challenges related to maintaining temporal coherence across frames. These challenges have led to the development of specialized models that can handle the unique demands of video data.

FRVSR (Frame-Recurrent Video Super-Resolution), proposed by Sajjadi *et al.*, integrates a recurrent mechanism where the output of the previous frame is fed into the network to super-resolve the next frame, maintaining temporal consistency. FRVSR achieved a PSNR of 29.6 dB on the Vid4 dataset for 4× upscaling, demonstrating superior temporal consistency compared to traditional methods that process frames independently (Sajjadi *et al.*, 2018). Yang *et al.* introduced RealVSR, focusing on real-world video data rather than synthetic benchmarks. Trained on a dual-camera system dataset, RealVSR improved generalization and showed PSNR gains of 0.5 dB on real-world test videos compared to models trained solely on synthetic data (Yang *et al.*, 2021). Isobe *et al.* developed Temporal Group Attention Networks (TGAN), which use a temporal attention mechanism to focus on frames contributing the most to perceived video quality, achieving a PSNR of 30.02 dB on the REDS dataset by managing complex motion and reducing artifacts (Isobe *et al.*, 2020). Kim *et al.* introduced AnimeSR, a model tailored for animation videos with sharp edges and flat regions, maintaining a PSNR of 33.1 dB on a custom anime dataset and preserving stylistic elements crucial for high-quality animation upscaling (Wu *et al.*, 2022).

Why Choose EDSR for Video Frames?

Given the complexity and computational demands of specialized Video Super-Resolution (VSR) models, EDSR offers a practical alternative for video frame upscaling. Although it does not address temporal

DOI: 10.3844/jcssp.2025.1283.1292

coherence explicitly, EDSR excels in enhancing spatial resolution, making it suitable for applications where individual frame quality is prioritized and temporal artifacts are minimal or manageable through post-processing. Its simpler architecture also enables faster inference, which is critical for real-time video enhancement.

While EDSR x4 provides superior frame-level clarity and efficiency, it lacks mechanisms to maintain temporal consistency—which is essential for smooth transitions in video. VSR models such as FRVSR, RealVSR, and TGAN address this using recurrent or attention-based designs, but their higher computational overhead limits their applicability in real-time scenarios. EDSR thus represents a trade-off, favoring speed and clarity over inter-frame consistency, and serves well in contexts like archival restoration or low-latency streaming.

CUDA Enhancement for Super-Resolution

CUDA has become indispensable in accelerating deep learning tasks, particularly in super-resolution. The ability to parallelize operations across thousands of GPU cores allows for significant speedups, making real-time super-resolution feasible.

Kim *et al.* evaluated the performance of TensorFlow and PyTorch on GPUs using super-resolution workloads, demonstrating that CUDA optimizations can reduce inference times by up to 50%, which is crucial for deploying models like EDSR in real-time applications where latency is a major concern (Tian & Wei, 2024). Ignatov *et al.* explored implementing super-resolution models on embedded GPUs, focusing on balancing performance and power consumption. Their approach enabled real-time processing on mobile devices, achieving a PSNR of 25.5 dB for 2× upscaling on 1080p video, highlighting the practicality of deploying SR models in portable applications (Ignatov *et al.*, 2021). Zhou *et al.* emphasized the importance of optimizing memory access patterns and utilizing CUDA streams to maximize throughput in super-resolution tasks, achieving real-time performance on multispectral image datasets. These techniques are directly applicable to optimizing EDSR for video applications, ensuring high-resolution outputs are generated efficiently (Barman *et al.*, 2021).

The literature review highlights EDSR's superior balance of performance, simplicity, and computational efficiency compared to models like FRVSR and RealVSR. While FRVSR achieves better temporal coherence in video (e.g., a PSNR of 29.6 dB on the Vid4 dataset), its recurrent architecture introduces higher computational complexity, making it less suitable for real-time applications. In contrast, EDSR offers competitive spatial resolution (e.g., 26.96 dB on Urban100) with lower computational demands, especially when optimized with CUDA, reducing inference times by up to 50%. Models like RealVSR, while effective for real-world video degradation, often require specialized datasets, whereas EDSR's scalability and efficiency make it more practical for real-time applications on embedded devices, such as in video surveillance and autonomous systems.

The Proposed Architecture

The proposed architecture for real-time video upscaling integrates dynamic resource management and deep learning-based super-resolution to efficiently enhance video quality. Figure (1) outlines the architecture, showcasing the flow of video data through the system, from pre-processing to dynamic load balancing, parallel processing across CPU and GPU, and post-processing. This system ensures optimal performance through synchronized merging and load monitoring, enabling real-time video streaming with minimized latency and maximized resolution.

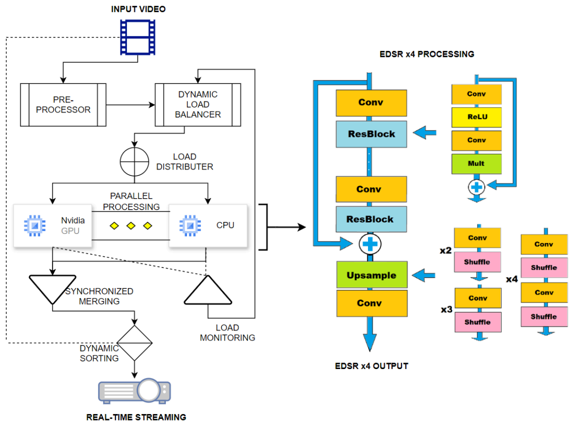

Fig. 1: System architecture for real-time video upscaling with dynamic cpu-gpu load balancing and edsr x4 super-resolution

The system architecture distributes video frames between the CPU and GPU based on real-time utilization analysis. Frames are processed in parallel, with each processing unit handling its assigned frames independently to maximize throughput and reduce latency. The distribution is dynamic—if the GPU is underutilized, more frames are allocated to it, and vice versa. This ensures both processors are efficiently engaged, enabling real-time performance without introducing frame lag or processing bottlenecks. Synchronization is maintained in the post-processing stage to preserve the original sequence of the video stream.

Pre-Processing Module

Before upscaling occurs, the input data is pre-processed, including frame normalization and format conversion, to prepare the data for super-resolution scaling. This step ensures that the input frames are optimized for efficient processing, reducing computational overhead during the subsequent upscaling phase.

Dynamic Load Balancing

This technique effectively divides duties between the CPU and GPU. It regularly checks CPU and GPU use to make sure that both are operating below-set load criteria. Jobs are prioritized based on computational complexity, with CPU-intensive tasks sent to the CPU queue and GPU-heavy tasks to the GPU queue. Tasks from the other queue are intelligently shifted from one queue to the other if one processor is not being used to its full potential. As a result, real-time video upscaling with balanced CPU-GPU processing is made possible while also ensuring that the system maintains responsiveness, maximizes performance, and effortlessly adapts to changing workloads.

The proposed dynamic load-balancing algorithm uses a priority-based allocation strategy guided by real-time CPU and GPU utilization. Frames are assigned to the processor with available capacity, based on predefined thresholds, to avoid overloading either resource. Unlike round-robin scheduling, this approach adapts to system load fluctuations, ensuring efficient parallelism.

EDSR x4 Model

The CUDA-enabled GPU plays a major role in the architecture as it reduces the computational time by a factor of over 100 times. However, this may vary based on the resolution of the input frame. Hence, in such cases the load is shared with the CPU to compensate for this loss in framerate. This ensures the most optimal upscaling of frames in real-time scenarios.

Post-Processing Module

Following the super-resolution, it is crucial to order the upscaled frames from the CPU and GPU in the same order as the video input. This is ensured by dynamically sorting them aided by comparing the upscaled image with the corresponding video input frame in that timeline. Hence, we now have achieved a sorted array of frames used for real-time streaming.

Materials and Methods

The proposed methodology outlines a systematic approach for real-time video enhancement using the EDSR x4 model, emphasizing dynamic CPU-GPU resource allocation and is depicted in Figure (2). The process begins with the acquisition of the input video, which is then systematically buffered for efficient frame management. Concurrently, CPU and GPU utilization are monitored to intelligently divide video frames based on the processing capabilities of each resource. Subsequently, the frames undergo EDSR x4 upscaling, leveraging parallelized computations on both CPU and GPU. The methodology adapts its processing strategy based on real-time utilization analysis, ensuring optimal resource usage. Finally, the upscaled frames are seamlessly integrated to generate an enhanced video output.

```
frame_buffer = [ ]
while True:
    ret, frame = cap.read()
    if not ret:
        break
    frame_buffer.append(frame)
```

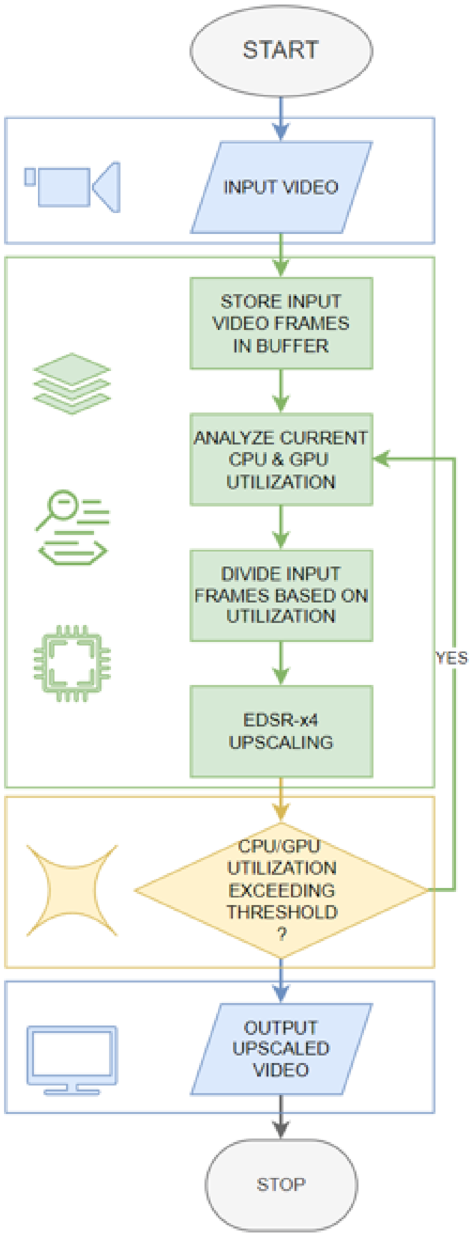

Fig. 2: Flowchart of the real-time video upscaling process

Input Video Acquisition

The first step is loading the input video using OpenCV. Frames are sequentially read from the video source using cv2.VideoCapture() is stored in a buffer for efficient processing.

Utilization Analysis

Utilization analysis dynamically assesses the current resource usage of both the CPU and GPU, providing real-time insights crucial for informed decision-making.

```
def analyze_cpu_utilization():
    cpu_utilization = psutil.cpu_percent(interval=1)
    return cpu_utilization

def analyze_gpu_utilization():
    gpu_utilization = GPUtil.getGPUs()[0].load * 100
    return gpu_utilization

cpu_utilization = analyze_cpu_utilization()
gpu_utilization = analyze_gpu_utilization()
```

Using the `psutil` library, CPU utilization is measured, while the `GPUtil` library is employed for GPU utilization analysis. These insights guide subsequent processing steps, ensuring adaptive strategies based on the dynamically changing system conditions.

Frame Division Based on Utilization

Frames are categorized based on dynamically assessed CPU and GPU utilization thresholds. This adaptive division ensures an optimal processing strategy, distributing frames between CPU and GPU paths based on the current system's resource usage. The utilization-driven frame division paves the way for efficient parallel processing and real-time video enhancement.

```
def divide_frames(frame_buffer, cpu_threshold,
gpu_threshold):
    cpu_frames, gpu_frames = [ ], [ ]
    for frame in frame_buffer:
        cpu_utilization = analyze_cpu_utilization(frame)
        gpu_utilization = analyze_gpu_utilization(frame)
        if cpu_utilization < cpu_threshold:
            cpu_frames.append(frame)
        elif gpu_utilization < gpu_threshold:
            gpu_frames.append(frame)
    return cpu_frames, gpu_frames
```

EDSR x4 Upscaling

In this phase, the EDSR x4 model is applied to upscale frames for both CPU and GPU processing paths. Dedicated instances of the model leverage parallelized computations to enhance each frame's quality. By dynamically utilizing both the CPU and GPU, the system reduces bottlenecks in the video frame processing pipeline. This dynamic CPU-GPU allocation not only distributes the computational workload efficiently but also minimizes the time spent on each frame, improving real-time processing capabilities. The combined power of both resources significantly lowers time per frame, as evidenced by an 18% reduction in overall processing time, resulting in more efficient upscaling at higher resolutions.

```
sr_cpu = cv2.dnn_superres.DnnSuperResImpl_create()
result_cpu = sr_cpu.upsample(cpu_frame)
sr_gpu = cv2.dnn_superres.DnnSuperResImpl_create()
sr_gpu.readModel(path)
sr_gpu.setPreferableBackend(cv2.dnn.DNN_BACKEND
    _CUDA)
sr_gpu.setPreferableTarget(cv2.dnn.DNN_TARGET
    _CUDA)
sr_gpu.setModel("edsr", 4)
result_gpu = sr_gpu.upsample(gpu_frame)
```

Utilization-Driven Processing

Utilization-driven processing involves conditional checks to determine whether CPU or GPU utilization surpasses predefined thresholds. This dynamic decision-making guides the subsequent processing strategy, ensuring optimal resource allocation for real-time enhancement.

```
def utilization_driven_processing(cpu_utilization,
gpu_utilization, cpu_threshold, gpu_threshold,
result_cpu, result_gpu):
    if cpu_utilization > cpu_threshold:
        process_on_cpu(result_cpu)
    elif gpu_utilization > gpu_threshold:
        process_on_gpu(result_gpu)
```

If CPU utilization exceeds a set threshold, the processing continues on the CPU path, and similarly for the GPU. This adaptability optimizes resource usage for enhanced real-time performance.

Output Generation

The implementation yielded impressive outcomes, demonstrating significant advancements in video quality, processing speed, and efficiency. Initial results were obtained on a system equipped with an RTX 3070 GPU and a Ryzen 7 5800H CPU. The Resource Utilization section further compares this setup with an RTX 4070, highlighting the improvements in computational efficiency and reduced system load offered by newer hardware.

Qualitative Inspection of Upscaled Images

The final upscaled frames undergo the generation of an enhanced video output. De-pending on the specific application requirements, the enhanced video can either be saved for later use or displayed in real-time. This step finalizes the real-time video enhancement process, providing a tangible output reflecting the improvements achieved through the EDSR x4 model and adaptive CPU-GPU utilization.

Results

A crucial aspect of evaluating the performance of an image upscaling model is to visually inspect the output images across different resolutions. While numerical

DOI: 10.3844/jcssp.2025.1283.1292

metrics can provide a sense of how well an image is preserved during upscaling, they do not always convey the full picture of how an image might be perceived by the human eye.

The images were upscaled to several key resolutions, such as from 144p. 360p, 480p, 720p to 600p, 1440p, 1920p and 2880p respectively. For each resolution, the original image is displayed alongside its upscaled version as in Figure (3). This side-by-side comparison allows for a direct assessment of how well the upscaling model preserves the quality of the image as the resolution increases.

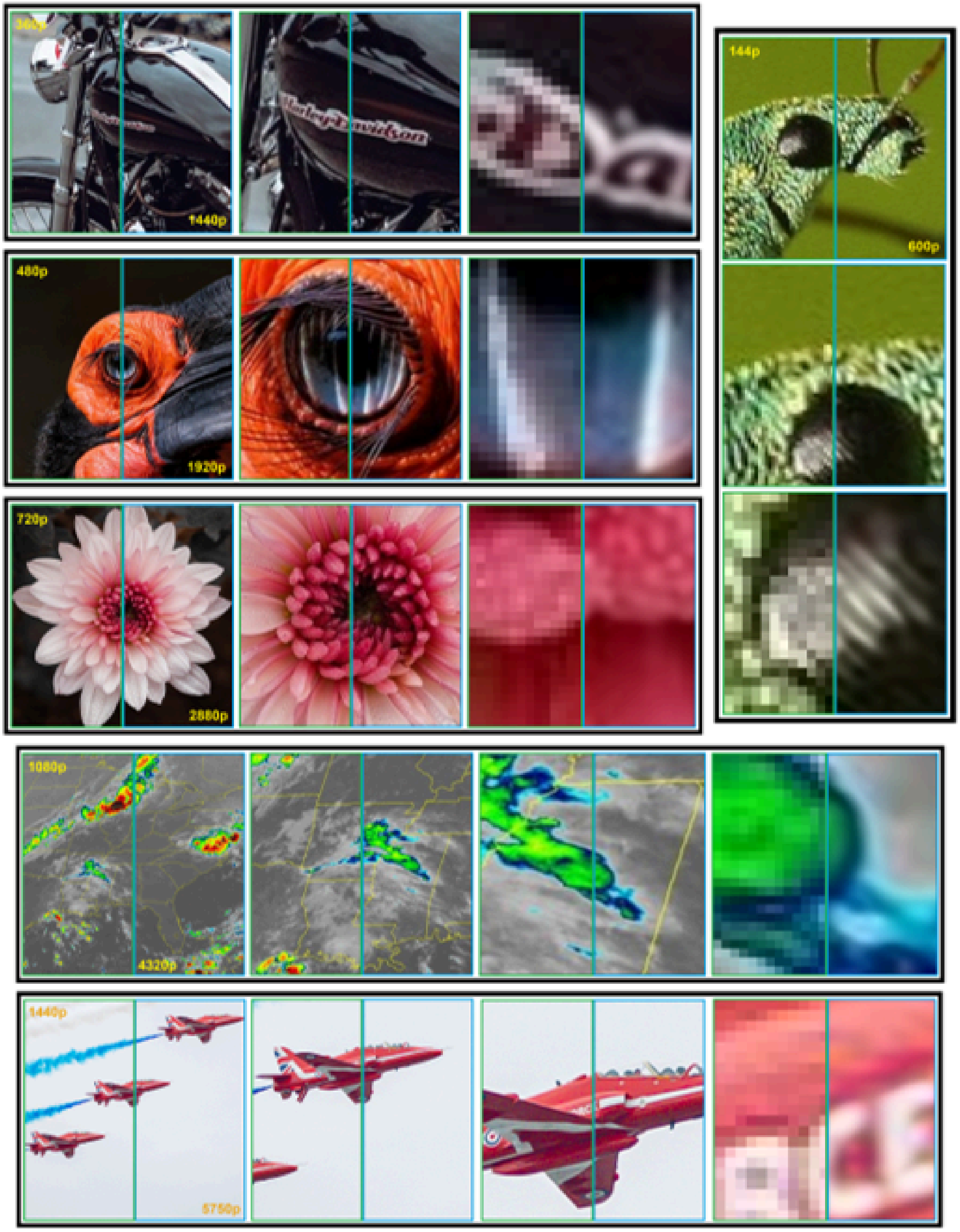

Fig. 3: Visual Comparison of Upscaled Images at Different Resolutions Using EDSR x4

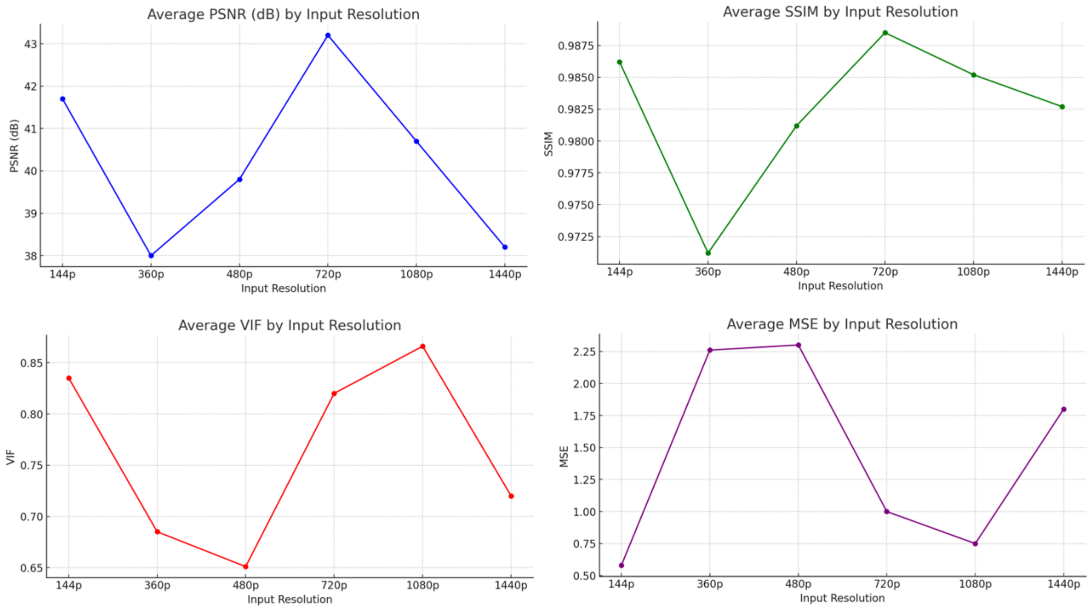

Fig. 4: Comparison of average psnr, ssim, vif, and mse across different resolutions

Quantitative Inspection of Upscaled Images

The effectiveness of the EDSR x4 model was evaluated using four widely recognized image quality metrics: Peak Signal-to-Noise Ratio, Structural Similarity Index Measure, Visual Information Fidelity, and Mean Squared Error and plotted in Figure (4).

Normalized Comparison of PSNR, SSIM, VIF, and MSE by Resolution. To provide a comprehensive comparison, the PSNR, SSIM, VIF, and MSE values were normalized and plotted together. For MSE, since lower values indicate better performance, the normalized MSE values were inverted (1 - normalized MSE) to ensure that higher values consistently represent better image quality across all metrics. The resulting graph, presented in Figure (5), reveals a distinct wave-like or cosine-like pattern across different resolutions. This pattern indicates that the relationship between resolution and image quality is not linear but fluctuates periodically. Understanding these patterns allows for more informed decisions when selecting resolutions for specific applications.

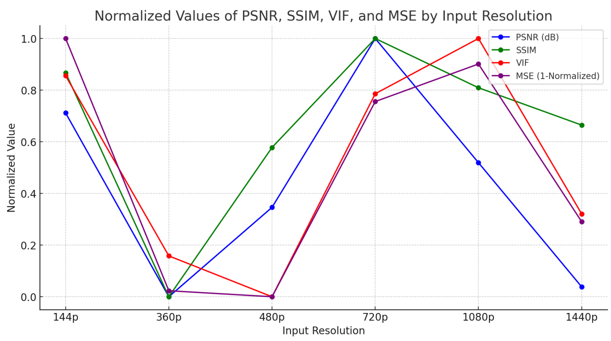

Fig. 5: Wave-like behavior in normalized values of PSNR, SSIM, VIF, and MSE across input resolutions

The wave-like or cosine-like pattern observed in the normalized PSNR, SSIM, and VIF metrics across varying input resolutions suggests a non-linear relationship between resolution and perceived image quality. While this behavior enhances our ability to select optimal resolutions for different applications, its underlying causes warrant deeper investigation. Potential contributing factors include interpolation artifacts, aliasing, or GPU memory and processing limitations at specific frame sizes. To further understand this periodicity, future studies could apply Fourier analysis or spectral decomposition techniques to the metric curves, potentially uncovering frequency-domain characteristics or hardware-level interactions influencing these fluctuations.

Comparative Analysis: EDSR x4 vs VSR Models

To better understand the trade-offs between real-time performance, spatial quality, and temporal consistency, we present a comparative analysis of EDSR x4 against leading Video Super-Resolution (VSR) models—FRVSR, RealVSR, and TGAN. These models are known for their ability to maintain temporal coherence across frames, a feature that EDSR inherently lacks due to its single-image super-resolution (SISR) architecture.

As shown in Figure (6), VSR models achieve higher temporal consistency and PSNR, with RealVSR and TGAN offering smoother transitions across frames. However, these gains come at the cost of significantly increased runtime, with processing times reaching up to 800 ms per 10 frames—substantially higher than EDSR's 205 ms. This increase in computational overhead limits their practicality for real-time applications, particularly in resource-constrained or latency-sensitive environments.

DOI: 10.3844/jcssp.2025.1283.1292

In contrast, EDSR x4 delivers competitive spatial resolution quality while maintaining a much lower computational cost, making it highly suitable for use cases like archival restoration, video surveillance, or live streaming, where real-time output is prioritized and minor temporal artifacts are acceptable. This analysis highlights the performance-consistency trade-off, reinforcing the rationale behind selecting EDSR for this work.

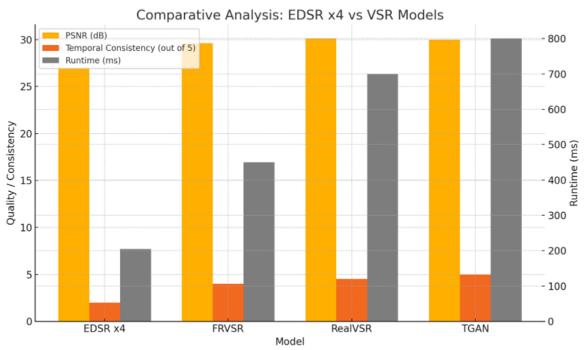

Fig. 6: Comparative analysis of EDSR x4 and VSR models (FRVSR, RealVSR, TGAN)

Real-Time Processing

When running on CPU alone, EDSR exhibits longer processing times compared to other methods, such as ZTE VIP and Diggers (Bhat *et al.*, 2021; Zhou *et al.*, 2022-2023; Park *et al.*, 2023; Yue *et al.*, 2023). However, the introduction of GPU acceleration significantly reduces EDSR's runtime, achieving a speedup of nearly 8.9x lowering the total processing time to 205 ms per 10 frames.

Comparatively, other methods, such as Diggers and Rainbow, perform well in terms of GPU runtime, with processing times of 199 ms and 180 ms per 10 frames, respectively. However, these methods fall short in image quality, as indicated by their PSNR values in previous evaluations. Table (1) shows the comparative runtimes of various Video Upscaling Methods against our proposed methodology.

Table 1: Utilization of RAM, GPU, and CPU for various input video resolutions

Method	Runtime per 10 frames (CPU, ms)	Runtime per 10 frames (GPU, ms)	Speedup (CPU to GPU)	Runtime with Load Balancing (ms)
EDSR	2231	250	8.9x	205
Diggers	916	199	4.6x	N/A
Rainbow	429	180	2.4x	N/A
TerminalVision	448	Error	N/A	N/A
ZTE VIP	163	113	1.4x	N/A

This balance of PSNR and runtime makes EDSR with load balancing a standout method for scenarios where both image fidelity and computational efficiency are critical. The results indicate that, while other methods may offer faster runtimes, they do not deliver the same high level of image reconstruction quality, making EDSR the optimal choice for applications requiring both speed and precision.

The proposed CPU-GPU load-balancing mechanism has demonstrated an 18% reduction in processing time, lowering the upscaling time per frame to 205 ms. This efficiency gain is achieved by dynamically distributing the frame processing workload between the CPU and GPU, preventing bottlenecks and ensuring both resources are used to their maximum capacity.

CPU, RAM, and GPU utilization

The utilization rates of CPU, GPU, and RAM were monitored across various input video resolutions using both RTX 3070 and RTX 4070, as shown in Figure (7). With the RTX 3070, CPU usage approached the 80% threshold, and RAM utilization peaked at 100% for high resolutions, indicating potential memory constraints. GPU usage ranged from 60% to 80% as resolution increased. In contrast, the RTX 4070 demonstrated more efficient performance, maintaining lower utilization across all components—CPU usage remained below 80%, GPU usage stayed between 50 and 77%, and RAM usage showed a more gradual increase, peaking at 95% for 4K input. This highlights the benefits of newer hardware in reducing system strain during real-time video upscaling.

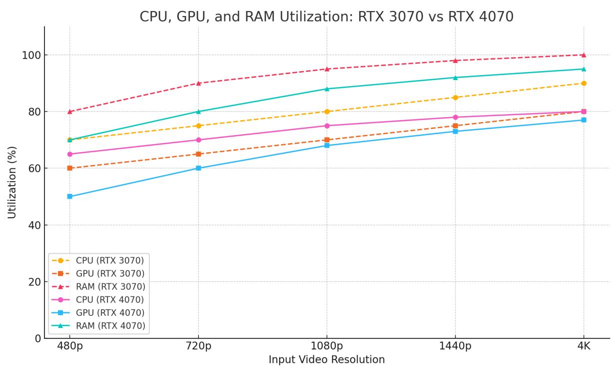

Fig. 7: CPU, GPU, RAM Resource utilization of RTX 3070 and RTX 4070

Discussion

Despite the improvements demonstrated by the proposed CPU-GPU load-balancing method for EDSR x4 upscaling, there are several areas where future work can enhance the system's scalability and efficiency. One limitation observed during testing is the 100% RAM utilization at higher resolutions, which may hinder performance in memory-constrained environments.

To address the high GPU RAM usage observed at higher resolutions, future work could explore memory optimization techniques such as model pruning, weight quantization, and tensor compression. These methods can significantly reduce memory footprint without substantial loss in performance. Pruning removes redundant parameters, quantization lowers precision to reduce storage requirements, and tensor compression optimizes intermediate data representations during inference. Integrating such strategies can make the system more scalable and efficient for high-resolution or memory-constrained environments.

Additionally, while the current implementation focuses on single GPU usage, extending the system to support multi-GPU architectures could significantly boost performance, especially when processing higher frame rates or larger batches. Another area for future exploration is the dynamic adjustment of CPU-GPU load-balancing thresholds based on real-time system conditions, allowing for more flexible and adaptive resource allocation.

Temporal coherence, which is not addressed by EDSR, could also be considered for future work, particularly for applications requiring smooth transitions between frames. Finally, investigating the wave-like behavior observed in quality metrics across different resolutions could uncover opportunities to optimize the super-resolution process further, improving both runtime and image quality.

Conclusion

This work explores the optimization of the EDSR x4 super-resolution model for real-time video enhancement, utilizing Nvidia-CuDNN acceleration and a CUDA-based load-balancing approach. By dynamically distributing computational workloads between the CPU and GPU, we demonstrated a significant 18% reduction in processing time, achieving 205 ms per 10 frames, which sets a new benchmark for real-time upscaling at multiple resolutions. This hybrid CPU-GPU approach not only re-duces bottlenecks but ensures consistent performance across diverse resolutions.

The quantitative metrics including PSNR, SSIM, and VIF illustrated that EDSR offers superior image quality while maintaining computational efficiency, making it an excellent choice for high-resolution video processing. Our approach also capitalized on efficient GPU utilization, maintaining usage between 60% to 80% on RTX 3070 and 50% to 77% on RTX 4070.

A central finding was the discovery of a wave-like pattern in the normalized PSNR, SSIM, and VIF metrics across different resolutions. This cosine-like fluctuation indicates that the relationship between image resolution and perceived quality is periodic rather than linear. Understanding this behavior provides valuable insights into selecting optimal resolutions for different applications.

Looking ahead, memory consumption, especially at higher resolutions, presents an area for further optimization. Enhancing memory efficiency could push the scalability of this system even further, making it suitable for more demanding real-time video applications.

Acknowledgment

We would like to thank Ramaiah Institute of Technology for the conditions and opportunities for research provided.

Funding Information

The authors have not received any financial support or funding to report.

Author's Contributions

Shubeeksh Kumaran: Implementation of the EDSR x4 model, optimization of dynamic CPU-GPU load balancing, acquisition of data, and original draft preparation.

Ganeshayya Shidaganti: Experimental evaluations, analysis of wave-like behavior in PSNR, SSIM, and VIF metrics, and interpretation of data; reviewed, edited.

Vishwachetan D: System integration, GPU utilization analysis, and methodology; reviewed, edited, and approved the version to be submitted.

Ethics

This study is an original research work and the lead the author confirms that all co-authors have reviewed and endorsed the manuscript without any ethical concerns.

References

Agustsson, E., & Timofte, R. (2017). NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1122-1131. https://doi.org/10.1109/cvprw.2017.150

Ahn, N., Kang, B., & Sohn, K.-A. (2018). Fast, Accurate, and Lightweight Super-Resolution with Cascading Residual Network. *Computer Vision - ECCV 2018*, *11214*, 256-272. https://doi.org/10.1007/978-3-030-01249-6_16

Barman, T., Deka, B., & Prasad, A. V. V. (2021). GPU-Accelerated Adaptive Dictionary Learning and Sparse Representations for Multispectral Image Super-resolution. *2021 IEEE 18th India Council International Conference (INDICON)*, 1-7. https://doi.org/10.1109/indicon52576.2021.9691521

Bhat, G., Danelljan, M., Van Gool, L., & Timofte, R. (2021). Deep Burst Super-Resolution. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 9205-9214. https://doi.org/10.1109/cvpr46437.2021.00909

DOI: 10.3844/jcssp.2025.1283.1292

Dong, C., Loy, C. C., He, K., & Tang, X. (2016). Image Super-Resolution Using Deep Convolutional Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *38*(2), 295-307. https://doi.org/10.1109/tpami.2015.2439281

Dong, C., Loy, C. C., & Tang, X. (2016). Accelerating the Super-Resolution Convolutional Neural Network. *Computer Vision - ECCV 2016*, *9906*, 391-407. https://doi.org/10.1007/978-3-319-46475-6_25

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778). https://doi.org/10.1109/cvpr.2016.90

Ignatov, A., Romero, A., Kim, H., Timofte, R., Ho, C. M., Meng, Z., Lee, K. M., Chen, Y., Wang, Y., Long, Z., Wang, C., Chen, Y., Xu, B., Gu, S., Duan, L., Li, W., Bofei, W., Diankai, Z., Chengjian, Z., … Yan, Y. (2021). Real-Time Video Super-Resolution on Smartphones with Deep Learning, Mobile AI 2021 Challenge: Report. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2535-2544. https://doi.org/10.1109/cvprw53098.2021.00287

Isobe, T., Li, S., Jia, X., Yuan, S., Slabaugh, G., Xu, C., Li, Y.-L., Wang, S., & Tian, Q. (2020). Video Super-Resolution With Temporal Group Attention. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8005-8014. https://doi.org/10.1109/cvpr42600.2020.00803

Kim, J., Lee, J. K., & Lee, K. M. (2016). Accurate Image Super-Resolution Using Very Deep Convolutional Networks. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1646-1654. https://doi.org/10.1109/cvpr.2016.182

Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., & Shi, W. (2017). Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 105-114. https://doi.org/10.1109/cvpr.2017.19

Lim, B., Son, S., Kim, H., Nah, S., & Lee, K. M. (2017). Enhanced Deep Residual Networks for Single Image Super-Resolution. *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1132-1140. https://doi.org/10.1109/cvprw.2017.151

Park, S., Cho, Y., Jun, H., Lee, J., & Cha, H. (2023). OmniLive: Super-Resolution Enhanced 360° Video Live Streaming for Mobile Devices. *Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services*, 261-274. https://doi.org/10.1145/3581791.3596851

Sajjadi, M. S. M., Vemulapalli, R., & Brown, M. (2018). Frame-Recurrent Video Super-Resolution. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6626-6634. https://doi.org/10.1109/cvpr.2018.00693

Shi, W., Caballero, J., Huszar, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D., & Wang, Z. (2016). Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1874-1883. https://doi.org/10.1109/cvpr.2016.207

Tian, J., & Wei, Y. (2024). An Inference Performance Evaluation of TensorFlow and PyTorch on GPU Platform Using Image Super-Resolution Workloads. *2024 16th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 1-6. https://doi.org/10.1109/ecai61503.2024.10607564

Tovar, N., & Kwon, S. S.-C. (2021). Deep Learning based Real-Time Image Upscaling for Limited Data Rate and Prevalent Resources. *2021 IEEE Green Energy and Smart Systems Conference (IGESSC)*, 1-5. https://doi.org/10.1109/igessc53124.2021.9618687

Wu, Y., Wang, X., Li, G., & Shan, Y. (2022). AnimeSR: Learning Real-World Super-Resolution Models for Animation Videos. *Advances in Neural Information Processing Systems*, *35*, 11241-11252.

Xiang, X., Tian, Y., Rengarajan, V., Young, L. D., Zhu, B., & Ranjan, R. (2022). Learning Spatio-Temporal Downsampling for Effective Video Upscaling. *Computer Vision - ECCV 2022*, *13678*, 162-181. https://doi.org/10.1007/978-3-031-19797-0_10

Yang, X., Xiang, W., Zeng, H., & Zhang, L. (2021). Real-world Video Super-Resolution: A Benchmark Dataset and A Decomposition based Learning Scheme. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, 4761-4770. https://doi.org/10.1109/iccv48922.2021.00474

Yue, S., Li, C., Zhuge, Z., & Song, R. (2023). EESRNet: A Network for Energy Efficient Super-Resolution. *Computer Vision - ECCV 2022 Workshops*, *13802*, 602-618. https://doi.org/10.1007/978-3-031-25063-7_38

Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., & Fu, Y. (2018). Image Super-Resolution Using Very Deep Residual Channel Attention Networks. *Computer Vision - ECCV 2018*, *11211*, 294-310. https://doi.org/10.1007/978-3-030-01234-2_18

Zhang, Y., Tian, Y., Kong, Y., Zhong, B., & Fu, Y. (2018). Residual Dense Network for Image Super-Resolution. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2472-2481. https://doi.org/10.1109/cvpr.2018.00262

Zhou, L., Zhang, N., Hsu, C. C., Singer, M., Zeng, X., Li, Y., Song, H., Jornet, J., Wu, Y., & Gan, Q. (2022). Super-Resolution Displacement Spectroscopic Sensing over a Surface "Rainbow". *Engineering*, *17*, 75-81. https://doi.org/10.1016/j.eng.2022.03.018

Zhou, L., Zhang, N., Hsu, C. C., Singer, M., Zeng, X., Li, Y., Song, H., Jornet, J., Wu, Y., & Gan, Q. (2023). Plasmonic "Rainbow" for Super-Resolution Displacement Spectroscopic Analysis and Surface Biosensing. *CLEO 2023*, FF1C.3.