

# MambaLiteSR: Image Super-Resolution with Low-Rank Mamba using Knowledge Distillation

Anonymous Submission

**Abstract**—Generative Artificial Intelligence (AI) has gained significant attention in recent years, revolutionizing various applications across industries. Among these, advanced vision models for image super-resolution are in high demand, particularly for deployment on edge devices where real-time processing is crucial. However, deploying such models on edge devices is challenging due to limited computing power and memory. In this paper, we present MambaLiteSR, a novel lightweight image Super-Resolution (SR) model that utilizes the architecture of Vision Mamba. It integrates State Space Blocks and a reconstruction module for efficient feature extraction. To optimize efficiency without affecting performance, MambaLiteSR employs knowledge distillation, transferring essential information from a larger Mamba-based teacher model to a smaller student model through hyperparameter tuning. Through a mathematical analysis of model parameters and their impact on the Peak Signal-to-Noise Ratio (PSNR), we identify key factors and adjust them accordingly. Our comprehensive evaluation shows that MambaLiteSR outperforms state-of-the-art edge SR methods by reducing latency and power consumption while maintaining competitive PSNR and SSIM scores across benchmark datasets such as Set5, Set14, and BSD100. It also reduces the power usage during training by adopting low-rank approximation. Moreover, MambaLiteSR reduces the total number of parameters without degrading performance, enabling the efficient deployment of generative AI models on resource-constrained devices. Deployment on the embedded NVIDIA Jetson Orin Nano confirms the superior balance of MambaLiteSR size, latency, and resource efficiency. The experimental results show that MambaLiteSR achieves performance comparable to both the baseline and other edge models while using 15% fewer parameters than the baseline. It also reduces power consumption by up to 3.4× and offers faster inference compared to state-of-the-art SR edge models, all while maintaining low energy consumption during training.

**Index Terms**—Image Super-Resolution, Mamba, Knowledge Distillation, Low-Rank Approximation, Edge Computing

## I. INTRODUCTION

Generative Artificial Intelligence (AI) models have gained attention in recent years for their ability to generate outputs such as images and text based on input data [1], [2]. These models learn the patterns and structures of training data and use them to produce results. Image Super-Resolution (SR) is one such task in Generative AI and computer vision. The goal is to generate a High-Resolution (HR) image from a Low Resolution (LR) one [3]. An efficient SR model achieves high Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity (SSIM); PSNR measures the reconstruction quality by comparing the pixel-wise difference between the HR output and the ground truth, while SSIM evaluates the perceptual similarity by considering structural information, luminance, and contrast. However, deploying these models on edge devices might be challenging because of their complexity and

high intensiveness. Since edge devices are constrained in several aspects such as computing resources and power, the importance of model size, number of parameters, and FLOPs comes into matter [4], [5].

Early SR approaches relied on convolutional networks [6]–[8]. While small, these CNN-based methods demonstrated relatively low PSNRs. Therefore, efforts have been made to improve the performance [9]–[11] by removing unnecessary modules, bypassing low-frequency information, and integrating denoising as a prior. Recently, research has focused on developing smaller and faster models with novel architectures, attention mechanisms, and hybrid models [12]–[14].

With the introduction of Mamba [15], a revolution happened in optimizing feature extraction. This architecture was quickly adopted across various vision tasks, including image classification, object detection, and semantic segmentation [16], [17]. Having similar results to previous works but with better processing speed, led to its adoption in SR frameworks, where it involved combining the lightweight structure of Mamba with attention mechanisms and transformer-based modules [19]–[21]. However, even though Mamba improved the latency, model deployment on edge devices is still challenging because of its size and power usage.

To address these challenges, we propose MambaLiteSR, which builds on previous studies by enhancing the existing Mamba-based architecture with significant improvements. In this paper, we focus on minimizing the model size and training power usage while maintaining comparable performance by optimizing the loss function, utilizing low-rank knowledge distillation, and carefully fine-tuning hyperparameters. After finalizing the architecture and loss function, we employed low-rank approximation, which is widely used in machine learning studies such as LoRA [22], to decrease the power usage during training. Then, knowledge distillation was applied to reduce the model size. To the best of our knowledge, no existing work has explored the simultaneous use of Mamba, knowledge distillation, and low-rank approximation for optimizing SR generative AI models on edge devices. In this context, we enhanced the Mamba vision model to improve performance, energy usage while training, and compactness, ensuring it fits seamlessly on edge devices. To evaluate the proposed MambaLiteSR, experiments were conducted on key knowledge distillation parameters to achieve optimal results. The experimental results demonstrate that MambaLiteSR achieves comparable performance to baseline and edge models while being 15% smaller than the baseline. For edge-device evaluation, the model was tested on the embedded NVIDIA

Jetson Orin Nano. The results show that our model achieves comparable PSNR and SSIM with the state-of-the-art edge models while consuming less dynamic power and delivering faster latency.

Our contributions are summarized as follows:

- Image Super Resolution model improvement to achieve comparable performance by optimizing the loss function, utilizing knowledge distillation, and carefully fine-tuning its hyperparameter.
- Assessment of the matrix ranks used in Mamba architecture on model size, performance, and training power usage, as well as the distillation parameter  $\alpha$  on model performance.
- Real-world experiment by deployment of the proposed MambaLiteSR on embedded NVIDIA Jetson Orin Nano, with comparisons made against similar models in terms of power usage and latency.

## II. RELATED WORK

Image super-resolution has been a major task in computer vision, with early approaches relying on interpolation techniques such as bicubic scaling [4], [23], [24]. While these methods are computationally efficient, they lack adaptability to image features and often produce artifacts like blurriness and jagged edges in images. Advances like SRCNN [6] and FSRCNN [8] introduced learnable features through CNNs, significantly improving SR performance. However, these models are computationally intensive and might be challenging for real-time deployment on resource-constrained devices.

To address efficiency concerns, lightweight SR models such as ESPCN [25], DVMSR [19], and SRMamba-T [21] have been proposed. ESPCN utilizes sub-pixel convolution layers to reduce computational overhead, but it struggles to maintain high reconstruction quality at larger scales. DVMSR utilizes Vision Mamba [27] modules and state space blocks to balance efficiency and accuracy. SRMamba-T combines Mamba and Transformer architectures to balance computational efficiency with high performance.

Frameworks like ESHP [5] extend these advancements by leveraging heterogeneous hardware to optimize SR tasks. ESHP dynamically allocates CPU, GPU, and NPU resources using deep reinforcement learning. However, its dependence on specialized hardware ecosystems introduces complexity and limits deployment flexibility. Edge-SR [4] proposes lightweight one-layer architectures for real-time applications. Although practical for constrained devices, its performance is often inferior to more advanced multi-layer networks. Similarly, thermal imaging SR pipelines [28] and facial verification systems [29] focus on specific use cases but are not flexible enough for wider SR applications.

The proposed MambaLiteSR further advances these foundations by integrating low-rank Mamba architecture and a knowledge distillation framework, offering a unified solution to the limitations of prior methods. Unlike simpler interpolation-based techniques, MambaLiteSR dynamically adapts to complex image features for improved SR quality. Meanwhile, its

low-rank design addresses the memory and computational constraints typically faced by CNN-based or transformer-heavy approaches, reducing reliance on specialized hardware. By judiciously managing parameter usage and supporting real-time deployment, MambaLiteSR provides an effective, flexible solution for lightweight SR on edge devices.

## III. BACKGROUND

Image super-resolution aims to reconstruct a high-resolution image from a low-resolution input while keeping details. Traditional methods cannot recover high-frequency details, leading to blurry results [30], [31]. Deep learning-based SR models, particularly those utilizing CNN and Transformer-based architectures [6], [8], [21], have significantly improved performance. However, these models often require large computational resources, making deployment on edge devices challenging. In this section, various parts contributing to model size reduction or latency reduction are discussed. In addition, strategies for improving the performance of small models are discussed.

**Mamba and State Space Models** The Mamba [15] architecture introduces Selective State Space Models, a lightweight alternative to Transformers, that reduces computational complexity while maintaining feature extraction capabilities. The state space representation in Mamba is formulated as follows:

$$\mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \quad (1)$$

$$\frac{d}{dt}\mathbf{x}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t) \quad (2)$$

where  $\mathbf{x}(t)$  represents the hidden state,  $\mathbf{u}(t)$  is the input signal and  $\mathbf{A}$ ,  $\mathbf{B}$ , and  $\mathbf{C}$  are learnable matrices. This structure enables Mamba to capture long-range dependencies efficiently while requiring fewer parameters than traditional self-attention mechanisms. As a result, several efforts have been made to apply this method across various tasks. Vision Mamba [16] is such an effort that utilizes the efficient architecture of Mamba for computer vision tasks like image classification. Later, DVMSR [19] was proposed, utilizing Mamba Vision in the image super-resolution task. Figure 1 demonstrates the architecture of DVMSR and Vision Mamba, showing the contribution of Vision Mamba to DVMSR.

**Embedding Dimension** The embedding dimension significantly influences the number of parameters in the model. For an SR model, the number of parameters in a single layer can be estimated as:

$$P = d_{in} \times d_{out} + d_{out} \quad (3)$$

where  $d_{in}$  and  $d_{out}$  are the input and output embedding dimensions, respectively. Reducing the embedding dimension from  $d_{base}$  to  $d_{small}$  leads to a reduction in parameter count by a factor of:

$$\frac{d_{small}^2}{d_{base}^2} \quad (4)$$

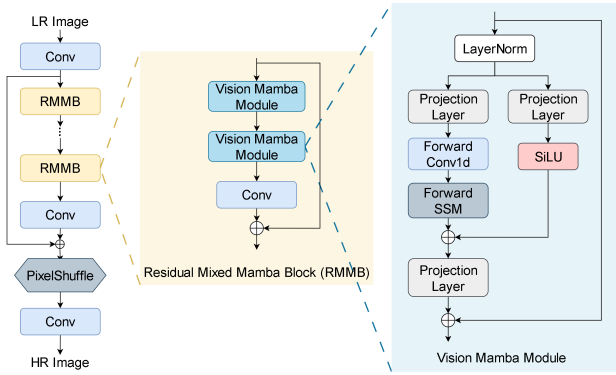


Fig. 1. Architecture of DVMSR [19] and Vision Mamba [16]: Input image is preprocessed and fed into the DVMSR model, which consists of Vision Mamba modules, convolution layers, and a decoder.

**Low-Rank Approximation** Low-rank approximation is used to further compress the model by reducing the rank of weight matrices while preserving essential information. Given a weight matrix  $\mathbf{W} \in R^{m \times n}$ , a low-rank factorization approximates  $\mathbf{W}$  as:

$$\mathbf{W} \approx \mathbf{U}\mathbf{V}^T \quad (5)$$

where  $\mathbf{U} \in R^{m \times r}$  and  $\mathbf{V} \in R^{n \times r}$ , with rank  $r \ll \min(m, n)$ . This leads to a reduction in the number of computations, and FLOPs, and less power usage.

**Knowledge Distillation** Knowledge distillation is employed to transfer knowledge from a larger teacher model to a smaller student model to have a similar performance with fewer parameters. This process introduces a trade-off between the soft targets of the teacher and the ground truth labels, controlled by the parameter  $\alpha$ . The loss function in knowledge distillation is defined as:

$$L = \alpha L_{KD} + (1 - \alpha) L_{GT} \quad (6)$$

where  $L_{KD}$  is the distillation loss calculated using teacher soft predictions, and  $L_{GT}$  is the loss using ground truth labels. A higher  $\alpha$  puts more emphasis on the teacher’s outputs. In contrast, a lower  $\alpha$  puts the student to rely more on the ground truth, which may limit the benefits of distillation. In summary, tuning  $\alpha$  is to balance model efficiency.

#### IV. PROPOSED APPROACH

To design an efficient image super-resolution model, we build upon the Vision Mamba architecture while optimizing its structure and using knowledge distillation. The goal is to maintain the performance of SR models while reducing computational complexity and model size and simplifying it for deployment on edge devices. By combining the efficient modeling of Mamba, embedding dimension adjustments, low-rank approximations, and tuning hyperparameters, MambaLiteSR achieves a balanced trade-off between model size, computational efficiency, and SR performance, making it suited for deployment on resource-constrained devices. In this section,

we go over the main aspects and materials of our work. Experimental results show that this combination of parameters balancing allows for a substantial reduction in model size and energy usage while preserving high reconstruction accuracy.

Figure 2 presents our high-level diagram, which includes the model, knowledge distillation process, and parameters of interest. The flow consists of the following key components:

##### A. Embedding Dimension

The embedding dimension is a crucial factor in model efficiency, which directly affects the number of parameters and computational cost. By reducing the embedding dimension from 192 to 60 for the model, we achieve a substantial reduction in memory usage and inference latency, which according to the equation 4, leads to a 10 times reduction of the model size. Experimental results confirm that this reduction in embedding dimension does not significantly impact reconstruction quality and allows for efficient deployment on edge devices.

In this study, images are cropped into smaller patches. Specifically, from each ground truth image 8 mini-patches of  $64 \times 64$  are extracted randomly. To further preprocess the data, random horizontal and vertical flips and rotations are applied. Then, each of these  $64 \times 64$  patches is converted into a vector of 50 and 32 features, for teacher and student models respectively.

##### B. Large Teacher Model

The input image is first passed through a convolutional layer, which acts as a shallow feature extractor to derive basic feature maps. These features then go through a set of Residual Mixed Mamba Blocks (RMMB), which consist of several Vision Mamba modules designed to capture long-range dependencies. Each RMMB internally contains a basic layer with a configurable number of Blocks, each Block has exactly one Mamba mixer inside and An internal residual convolution and skip connection (residual group).

Therefore, RMMBs perform deep feature extraction and contain multiple Vision Mamba Modules along with a convolution layer and a residual connection. In the proposed large teacher model, the Mamba layer is low-rank to reduce the number of computations. After extracting deep features, a global residual connection is applied to fuse shallow features from the initial feature map with deep features. These features are then up-sampled by a method of pixel-shuffle to create the final HR output. To summarize, there are 16 Mamba layers in the teacher model with an embedding dimension of 60, which will be covered in the following subsections.

##### C. Tiny Student Model

To improve efficiency, a student model is trained with a lower number of RMMBs and a smaller embedding dimension. The student model follows a similar architecture to the teacher but with reduced parameters. Knowledge distillation allows the student to mimic the performance of the teacher while being more computationally efficient. In this work, it has 4 RMMBs,

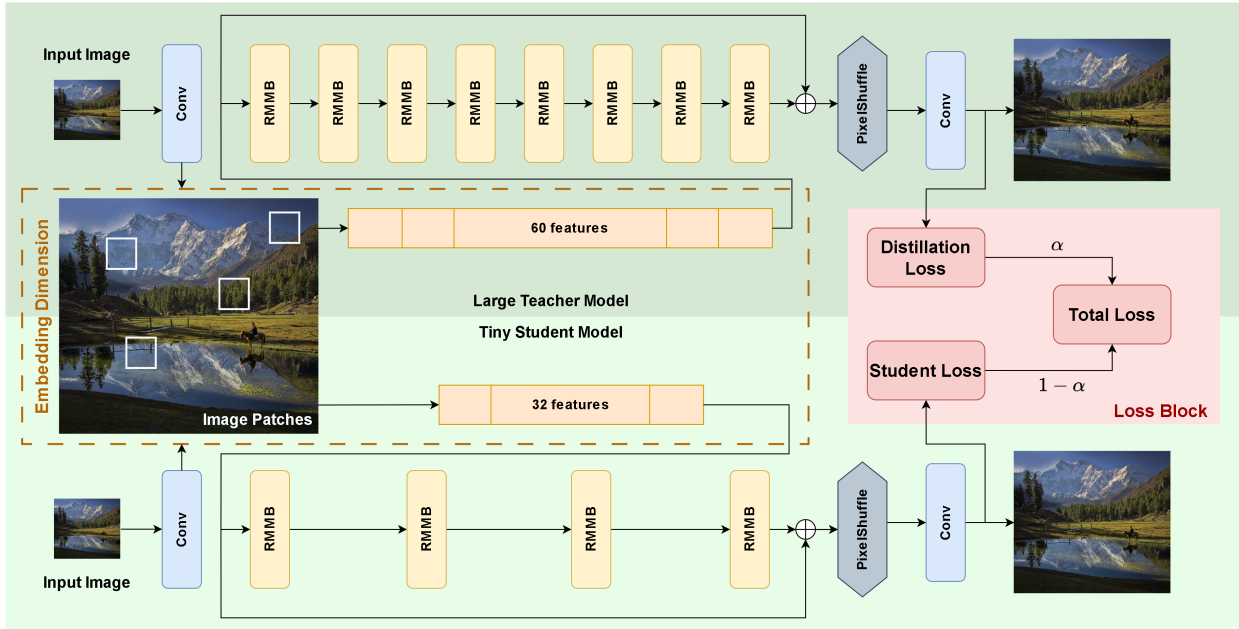


Fig. 2. High-Level overview of MambaLiteSR process: Low-resolution input image ( $64 \times 64$ ) is preprocessed and fed into the knowledge distillation process, generating the high-resolution output ( $256 \times 256$ ). Weighted distillation and student losses enable the student model to learn efficiently under teacher supervision. Proper embedding dimension, which determines the feature vector size resulting from image patches, makes it suitable for edge devices.

resulting in 8 Mamba layers, with an embedding dimension of 32. The student model is optimized to keep essential features while achieving comparable PSNR and SSIM. Note to mention that the input to both teacher and student model is the same and they go through the same flow of preprocessing.

#### D. Loss Block

As stated in the equation 7,  $\alpha$  controls the trade-off between distillation and direct supervision, and tuning it ensures the optimal balance between performance and model efficiency. In the total loss function, L1 loss for both distillation and student loss is considered. L1 loss measures the average difference between predicted and actual values. We write the combined loss as:

$$L = \alpha L_1(y_{student}, y_{teacher}) + (1 - \alpha) L_1(y_{student}, y_{GT}) \quad (7)$$

By assigning a proper value for  $\alpha$ , each of the two loss terms is weighted to optimize the trade-off between performance and model size.

#### E. Low-Rank Mamba

Another key strategy is the low-rank approximation in the Mamba mixers. This reduces complexity and energy usage during training while maintaining model performance by factoring each  $dim \times dim$  weight into  $dim \times rank$  and  $rank \times dim$ . By reducing the rank from  $dim/2$ , where  $dim$  stands for the matrix dimension, we can ensure the reduction in computations. However, the rank layers are such a small fraction of the overall architecture, that changing ranks does

not materially affect the total parameter count. But at runtime, the larger rank requires more FLOPs (more multiplications/additions), so it draws more power under load. Therefore, rank mostly impacts computation rather than overall parameter storage. Therefore, to reduce the energy usage and the number of parameters between layers, a low-rank approximation is applied.

In this work, we fine-tune these two hyperparameters: the distillation weight  $\alpha$  and the rank used in low-rank approximation. These parameters are adjusted based on experimental evaluations to achieve an optimal trade-off between efficiency and performance.

## V. EXPERIMENTAL RESULT

In this section, we begin by presenting the experimental setup, which includes details about the dataset and the implementation. Then, we evaluate the proposed LightMambaSR model, focusing on its size, performance, and power consumption in comparison to state-of-the-art methods.

#### A. Experimental Setup

**Datasets.** In this work, the DF2K (DIV2K + Flickr2K) [32] dataset was used. This dataset consists of 3450 high-resolution images and their corresponding low-resolution versions, which were downscaled by factors of 2, 3, and 4. For this research, since upscaling with higher factors are harder problem for networks to fix and typically requires more parameters to learn, the experiments are done only on the scale of 4. As for the validation set, 100 images exist for each of the low-resolution and high-resolution images of DIV2K. For the test, 4 standard benchmark datasets were used: Set5 [33], Set14 [34], BSD100 [35], and Urban100 [36].

**Implementation Details.** For faster processing during training, images are cropped into smaller patches. Specifically, from each HR image 8 mini-patches of  $64 \times 64$  are extracted randomly. To further preprocess the data, random horizontal and vertical flips and rotations are applied. The model is trained using the Adam optimizer. The batch size is set to 128 and the training process takes 2500 iterations. The initial learning rate is set at  $2 \times 10^{-4}$  and is halved when the training iteration reaches specific milestones. For the embedding dimension of images, it is considered to be 60 to reduce the model size. For evaluation, we calculate PSNR and SSIM metrics on the Y channel in the YCbCr color space. All experiments are conducted on Lambda GPU Server [37], which utilizes NVIDIA GeForce RTX 4090 GPU.

Experiments are conducted with various reduced matrix ranks including ranks 30 and 2. To further reduce the model size, knowledge distillation is applied, the same process as DVMSR [19], but with different values and hyperparameters. For this purpose, the embedding dimension is lowered to 32, which results in a 4-times reduction compared to our teacher model size, which was also 10 times smaller than the baseline model, as well as a reduction in the number of RMMB blocks. To assess the performance of different values of  $\alpha$  parameter in knowledge distillation an experiment was conducted comparing the result of changing this value over 1000 iterations for each of teacher and student models.

For the power consumption and latency measurements, we deploy the models on the embedded Nvidia Jetson Orin Nano board [38] (Figure 3.a) and do the measurements over 1000 samples, and the average was calculated. The Jetson Orin Nano Developer Kit (8GB) features a hexa-core ARM Cortex-A78AE CPU (1.5 GHz) and a 512-core NVIDIA Ampere GPU (625 MHz), optimized for AI tasks like matrix multiplications and deep learning inference, making it ideal for edge AI applications. We converted the software models to ONNX format and optimized them into TensorRT representations for GPU-accelerated inference. During inference, the tegrastats utility was used to measure instantaneous and average dynamic power and GPU latency.

## B. Results

Figure 3.b illustrates the power consumption results on the embedded NVIDIA Jetson Orin Nano during inference. Experiments are done on the tiny student model to evaluate latency, inference speed, dynamic power, and energy efficiency, as summarized in Table I. The idle power consumption was measured at approximately 2.2W, and within 7s (for 1000 samples), the system initiated image pre-processing. Following this, the model execution and image input processing caused a  $2\times$  increase in power consumption before stabilizing for inference. The deployment results of the proposed MambaLiteSR model, compared against state-of-the-art methods on edge devices, are detailed in Table I. Measurements were averaged over 1000 samples. The results demonstrate that the proposed approach achieves up to  $3.4\times$  lower power

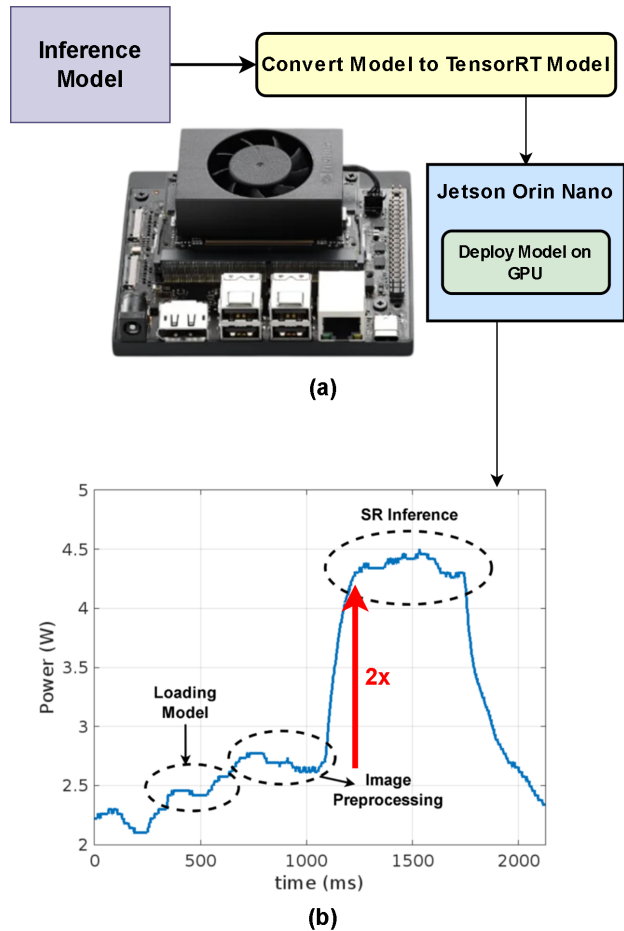


Fig. 3. (a) For measuring dynamic power and latency on the embedded NVIDIA Jetson Orin Nano, the student onnx model is converted to TensorRT format and then the measurement starts. (b) The plot shows the instantaneous power usage over time for 1000 samples, extracted using the tegrastats utility. Note that the command may introduce minor inaccuracies on the time axis.

consumption and reduced latency while maintaining SSIM and PSNR performance comparable to existing methods.

To reduce the number of parameters to make the model suitable for edge deployment the embedding dimension of 60 is applied, which leads to the 15% reduction in the number of parameters while keeping the performance acceptable with comparison with the Baseline [19]. The proposed LightMambaSR includes 370k parameters with a PSNR of 28.28 while the baseline has a 424k parameters with a PSNR of 32.19. There is a difference in PSNRs, which due to the increasing trend of the training we expect to achieve the same baseline PSNR as they did the experiments for 500K II.

Table III presents the rank configurations of MambaLiteSR along with their results. It can be seen that there is a negligible difference between the PSNRs of  $rank = 2$  and  $rank = 30$ . As a result,  $rank = 2$  is chosen to allow for  $1.7\times$  less energy consumption besides good performance.

This result can be seen in figure 4, which shows the validation results during teacher model training for rank values of 2 and 30, along with the moving average of their corre-

TABLE I

COMPARISON OF IMAGE QUALITY AND PERFORMANCE METRICS OF MAMBA LITE SR TO EXISTING METHODS IMPLEMENTED ON EDGE DEVICES (JETSON AGX XAVIER GPU [39]). MAMBA LITE SR REQUIRES UP TO  $3.4\times$  LESS POWER TO IMPLEMENT WHILE PERFORMING SIMILARLY. VALUES OF LATENCY AND POWER OF THE PROPOSED APPROACH ARE SPECIFIC TO THE EMBEDDED NVIDIA JETSON ORIN DEVICE. THE REPORTED LATENCY IS THE AVERAGE OF 1000 SAMPLES. EXPERIMENTS WERE CONDUCTED ON THE SCALE OF 4. NOTE THAT FOR FSRCNN, THE LATENCY IS ACHIEVED BY GETTING THEIR REPORTED AVERAGE OVER MULTIPLE DATASETS.

Algorithm	Latency [ms]	Power [mWatts]	Set5 [33]		Set14 [34]		BSD100 [35]		Urban100 [36]	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
eSR [4]	-	7100	30.62	0.860	27.48	0.751	26.93	0.714	24.42	0.718
eSR - fast [4]	-	3867	28.64	0.806	26.12	0.712	26.13	0.684	23.28	0.668
ESPCN [25]	5	6952	30.57	0.858	27.50	0.752	26.92	0.715	24.42	0.718
FSRCNN [26]	20 (average)	4795	30.16	0.845	27.19	0.742	26.74	0.707	24.09	0.702
MambaLiteSR (Ours)	<b>7</b>	<b>2104</b>	28.28	0.837	25.39	0.727	25.20	0.693	22.57	0.686

TABLE II

COMPARISON OF THE NUMBER OF PARAMETERS AND PSNR USAGE FOR THE BASELINE AND MAMBA LITE SR AT A SCALE FACTOR OF 4. THE PSNR RESULTS ARE EVALUATED ON THE SET5 DATASET.

Model	Parameters	iterations	PSNR
DVMSR [19]	424k	500,000	32.19
MambaLiteSR	<b>370k (15% smaller)</b>	2500	28.28

TABLE III

COMPARISON OF PSNR AND POWER USAGE FOR DIFFERENT RANKS AT A SCALE FACTOR OF 4. CONFIGS 1 AND 2 USE THE SAME CONFIGURATION, VARYING ONLY IN RANK AND BELONGING TO THE TEACHER MODEL. IT CAN BE SEEN THAT THE LOWER-RANK MODEL CONSUMES ABOUT HALF THE POWER OF THE HIGHER-RANK MODEL. THE REPORTED POWER MEASUREMENTS ARE AVERAGED VALUES OBTAINED DURING THE TRAINING OF THE TEACHER MODEL ON THE LAMBDA GPU SERVER, AND THE PSNR RESULTS ARE EVALUATED ON THE VALIDATION DATASET.

Config.	Rank	Parameters	iterations	PSNR	Power (W)
Config 1	<b>30</b>	370k	1500	28.88	<b>103</b>
Config 2	<b>2</b>	370k	1500	28.81	<b>60 (1.7x less)</b>

sponding GPU power usage during training on the Lambda GPU Server [37], with all other settings kept the same. While both models demonstrate similar training behavior and performance, the model with the lower rank demonstrates more efficient training; when  $rank = 2$ , GPU power usage is nearly half that of  $rank = 30$ .

To further reduce the model size, knowledge distillation is applied. The result of these values can be seen in table IV.

TABLE IV

COMPARISON OF PSNR FOR DIFFERENT  $\alpha$  VALUES. THE REPORTED PSNR BELONGS TO THE STUDENT MODEL ON THE VALIDATION SET.

$\alpha$	0.2	0.4	0.6	0.8
<b>PSNR</b>	28.71	26.35	26.69	27.95

As a result,  $\alpha = 0.8$  is chosen to enable both learning from the teacher and ground truth labels. Then, the student model is put to training for 2500 number of iterations, the same as the teacher model. Then, the experiments on  $\alpha$  values and their corresponding PSNRs inspected more with a wider range of values because of the probable formulation it suggested. Figure 5 demonstrates model performance on the validation

set with changes in  $\alpha$ , suggesting a potential inconsistency in gradient values between the learned teacher and the ground truth, as indicated in the equation 7, meaning that whenever the teacher and ground truth have near to similar weights in the loss function, PSNR might degrade. At the same time, when one of these becomes effectless, it means that the model is not learning or starts learning from scratch.

Figure 6 demonstrates the preprocessed input image, the corresponding MambaLiteSR output, and the ground truth image. The same process for a random patch is shown as well for better visualization. It can be seen that MambaLiteSR is successful in generating a super-resolution image similar to the ground truth image.

## VI. CONCLUSION

In this paper, we presented MambaLiteSR, a new image super-resolution model designed for edge devices. It combines a low-rank Mamba mixer with knowledge distillation to deliver outputs with super-resolution, relying on state space representations while checking the embedding dimension and rank settings to minimize parameters and computation, as well as the distillation parameter tuning for better performance. By integrating knowledge distillation, MambaLiteSR inherits the performance of a larger teacher model and achieves competitive PSNR and SSIM on standard benchmarks. Our experiments show that MambaLiteSR can reduce the model size by up to 15% compared to a baseline without sacrificing reconstruction quality significantly. Changes in rank resulted in a 50% reduction in power usage during training. Assessment of the distillation parameter led us to a realization between the teacher model and ground truth, resulting in a balanced knowledge distillation. We also demonstrated its real-time capability on the embedded NVIDIA Jetson Orin Nano, which maintains low power consumption and short inference times. By combining low-rank approximation, knowledge distillation, and efficient embedding strategies, MambaLiteSR offers a practical solution for generative AI applications on edge devices.

## REFERENCES

- [1] S. Feuerriegel, J. Hartmann, C. Janiesch, and P. Zschech, *Generative AI*, Springer, 2023.

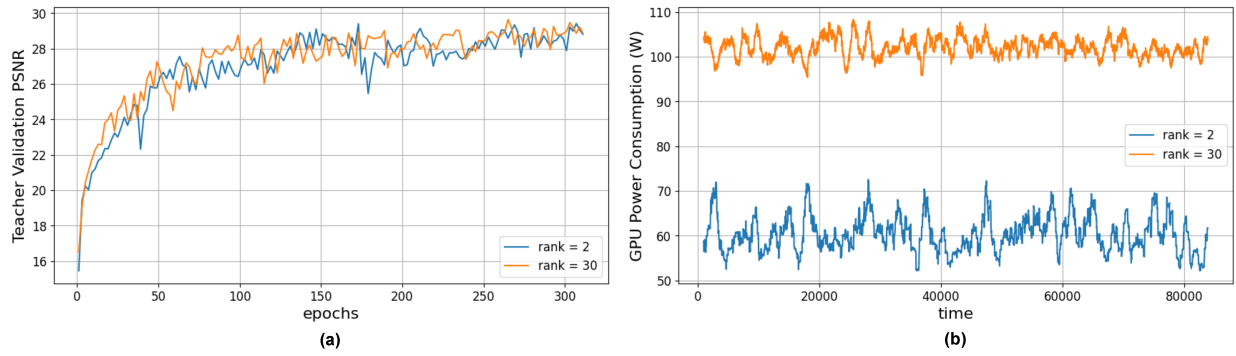


Fig. 4. Comparison between MambaLiteSR teacher model training when  $rank = 2$  and  $rank = 30$  over 1500 iterations: (a) depicts validation PSNR. (b) depicts the moving average of GPU power usage every 100 seconds with a window size of 100. At runtime, the larger rank requires more FLOPs, drawing more power under load. The training outcome and model performance act similar because, as indicated by equation 5, the matrix ultimately remains the same. The reported power measurements correspond to the NVIDIA GeForce RTX 4090 on Lambda GPU Server [37] using wandb [40] dashboard.

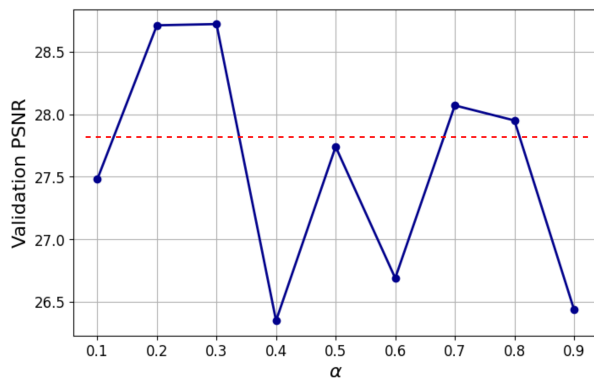


Fig. 5. Model performance on the validations set based on the changes in  $\alpha$ , suggesting a potential inconsistency in gradient values between the learned teacher and the ground truth. The reported PSNRs are after 1000 iterations of training for teacher and student models.

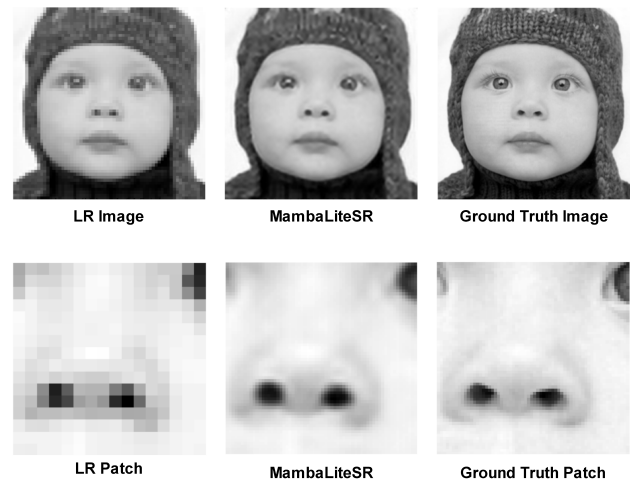


Fig. 6. Demonstration of LR image, MambaLiteSR output, and ground truth image, as well as a random patch cropped from the LR input and its corresponding MambaLiteSR output and ground truth patch.

- [2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, *Attention Is All You Need*, NeurIPS, 2023.
- [3] M. Yu, J. Shi, C. Xue, X. Hao, and G. Yan, *A review of single image super-resolution reconstruction based on deep learning*, Springer, 2023.
- [4] P. Navarrete Michellini, Y. Lu, and X. Jiang, *Edge-SR: Super-Resolution for the Masses*, WACV, 2022.
- [5] Q. Wang et al., *An Intelligent Co-Scheduling Framework for Efficient Super-Resolution on Edge Platforms*, IEEE IoT Journal, 2024.
- [6] C. Dong, C. C. Loy, K. He, and X. Tang, *Image Super-Resolution Using Deep Convolutional Networks*, ECCV, 2014.
- [7] J. Kim, J. K. Lee, and K. M. Lee, *Accurate Image Super-Resolution Using Very Deep Convolutional Networks*, CVPR, 2016.
- [8] C. Dong, C. C. Loy, and X. Tang, *Accelerating the super-resolution convolutional neural network*, ECCV, 2016.
- [9] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, *Image Super-Resolution Using Very Deep Residual Channel Attention Networks*, CVPR, 2018.
- [10] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, *Enhanced Deep Residual Networks for Single Image Super-Resolution*, CVPR, 2017.
- [11] K. Zhang, W. Zuo, S. Gu, and L. Zhang, *Learning Deep CNN Denoiser Prior for Image Restoration*, CVPR, 2017.
- [12] W. Zhang, L. Pan, K. Xu, G. Li, and Y. Lv, *Efficient self-calibrated and hierarchical refinement network for lightweight super-resolution*, ELSEVIER, 2024.
- [13] Z. He, S. Zhang, et al., *ESR-DDLN : Enhanced Single Image Super-Resolution Via Dual-Domain Learning Network*, IEEE International Conference on Multimedia and Expo (ICME), 2024.
- [14] Y. Zhang, X. Lin, H. Yang, J. He, L. Qing, X. He, Y. Li, and Honggang Chen, *A Multi-Attention Feature Distillation Neural Network for Lightweight Single Image Super-Resolution*, International Journal of Intelligent Systems, 2024.
- [15] A. Gu and T. Dao, *Mamba: Linear-Time Sequence Modeling with Selective State Spaces*, arXiv preprint arXiv:2312.00752, 2023.
- [16] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang, *Vision Mamba: Efficient Visual Representation Learning with Bidirectional State Space Model*, CVPR, 2024.
- [17] A. Hatamizadeh and J. Kautz, *MambaVision: A Hybrid Mamba-Transformer Vision Backbone*, CVPR, 2024.
- [18] F. Kong, M. Li, S. Liu, D. Liu, J. He, Y. Bai, F. Chen, and L. Fu, *Residual local feature network for efficient super-resolution*, CVPR, 2022.
- [19] X. Lei, W. Zhang, and W. Cao, *DVMSR: Distilled Vision Mamba for Efficient Super-Resolution*, ICCV, 2024.
- [20] J. Qiao, J. Liao, W. Li, Y. Zhang, Y. Guo, Y. Wen, Z. Qiu, J. Xie, J. Hu, and S. Lin, *Hi-Mamba: Hierarchical Mamba for Efficient Image Super-Resolution*, CVPR, 2024.
- [21] C. Liu, D. Zhang, G. Lu, W. Yin, J. Wang, and G. Luo, *SRMamba-T: Exploring the hybrid Mamba-Transformer network for single image super-resolution*, ELSEVIER, 2025.
- [22] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, *LoRA: Low-Rank Adaptation of Large Language Models*, International Conference on Learning Representations, 2022.
- [23] M. Jahnava, D. R. Raob, A. Sujathac, *A Comparative Study Of*

- Super-Resolution Interpolation Techniques: Insights For Selecting The Most Appropriate Method*, International Conference on Innovative Data Communication Technologies and Application, 2024.
- [24] Y. Chen and T. Pock, *Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration*, arXiv preprint arXiv:1508.02848, 2015.
- [25] W. Shi et al., *Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network*, CVPR, 2016.
- [26] C. Dong, C. Change Loy, and X. Tang, *Accelerating the super-resolution convolutional neural network*, ECCV, 2016.
- [27] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang, *Vision Mamba: Efficient Visual Representation Learning with Bidirectional State Space Model*, arXiv preprint arXiv:2401.09417, 2024.
- [28] P. Mathur et al., *A Real-Time Super-Resolution for Surveillance Thermal Cameras Using Optimized Pipeline on Embedded Edge Device*, IEEE AVSS, 2021.
- [29] F. Perez-Montes, J. Olivares-Mercado, and G. Sanchez-Perez, *An Efficient Facial Verification System for Surveillance That Automatically Selects a Lightweight CNN Method and Utilizes Super-Resolution Images*, MICAI, 2023.
- [30] R. Y. Tsai and T. S. Huang, *Multiframe image restoration and registration*, Advances in Computer Vision and Image Processing, 1984.
- [31] S. Baker and T. Kanade, *Limits on Super-Resolution and How to Break Them*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002.
- [32] R. Timofte, E. Agustsson, L. Van Gool, M. Yang, and L. Zhang, *Ntire 2017 challenge on single image super-resolution: Methods and results*, Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2017.
- [33] M. Bevilacqua, A. Roumy, C. Guillemot, and M. Alberi-Morel, *Low-complexity single-image super-resolution based on nonnegative neighbor embedding*, BMVC, 2012.
- [34] R. Zeyde, M. Elad, and M. Protter, *On single image scale-up using sparse-representations*, Curves and Surfaces, 2010.
- [35] D. R. Martin, C. C. Fowlkes, D. Tal, and J. Malik, *A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics*, ICCV, 2001.
- [36] J. Huang, A. Singh, and N. Ahuja, *Single image super-resolution from transformed self-exemplars*, CVPR, 2015.
- [37] Lambda Labs, *Lambda GPU Server*, <https://lambdalabs.com/>
- [38] NVIDIA, *NVIDIA Jetson Orin*, <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/>
- [39] NVIDIA, *NVIDIA Jetson AGX Xavier GPU*, <https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-agx-xavier/>
- [40] wandb.ai, *Weights & Biases*, <https://wandb.ai/>