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

Romina Aalishah, Mozhgan Navardi, and Tinoosh Mohsenin

Department of Electrical and Computer Engineering

Johns Hopkins University

{raalish1, mnavard1, tinoosh}@jhu.edu

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 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 improves the power consumption by up to 58% 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]–[4]. 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 [5]. 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 [6]–[12]. 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 [13]–[17].

Deep learning models play a crucial role in applications such as image processing [18]. Two commonly used architectures in deep learning are Fully Connected Networks (FCNs) and Convolutional Neural Networks (CNNs). Early SR approaches relied on CNNs [19]–[21]. While small, these CNN-based methods demonstrated relatively low PSNRs. Therefore, efforts have been made to improve the performance [22]–[24] 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 [25]–[27].

With the introduction of Mamba [28], a revolution happened in optimizing feature extraction. This architecture was quickly adopted across various vision tasks, including image classification, object detection, and semantic segmentation [29], [30]. 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 [32]–[34]. 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 power usage during inference and training while maintaining comparable performance by optimizing the loss function, utilizing low-rank Mamba, 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 [35], 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 simulta-

neous use of Mamba, knowledge distillation, and low-rank approximation for optimizing SR generative AI models on edge devices. In this context, we enhanced the Vision Mamba-based 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 [36]. The results show that our model achieves comparable PSNR and SSIM with the state-of-the-art edge models while consuming less dynamic power. Our contributions for the proposed MambaLiteSR are summarized as follows:

- Improvement of the image super-resolution model 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 α 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 performance.

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 [13], [37], [38]. 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 [19] and FSRCNN [21] 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 [39], DVMSR [32], and SRMamba-T [34] 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, which outperformed RLFN [31], the winner of the NTIRE 2023 Efficient Super-Resolution Challenge [41], utilizes Vision Mamba [42] 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 [14] 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
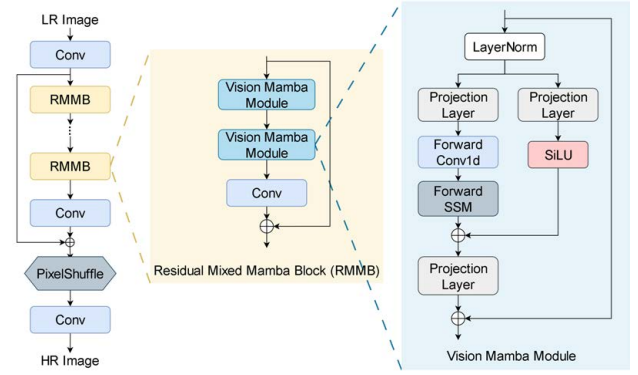

Fig. 1. Architecture of DVMSR [32] and Vision Mamba [29]: Input image is preprocessed and fed into the DVMSR model, which consists of Vision Mamba modules, convolution layers, and a decoder.

and limits deployment flexibility. Edge-SR [13] 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 [43] and facial verification systems [44] 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 design addresses the memory and computational constraints typically faced by CNN-based or transformer-heavy approaches, reducing reliance on specialized hardware. By 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 [45], [46]. Deep learning-based SR models, particularly those utilizing CNN and Transformer-based architectures [19], [21], [34], 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 the reduction in model size and power usage are explored. In addition, strategies for improving the performance of small models are discussed.

Mamba and State Space Models The Mamba [28] 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 [29] is such an effort that utilizes the efficient architecture of Mamba for computer vision tasks like image classification. Later, DVMSR [32] 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:

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

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, 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 α. 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 α puts more emphasis on the teacher's outputs. In contrast, a lower α puts the student to rely more on the ground truth, which may limit the benefits of distillation. In summary, tuning α is to balance model efficiency.

IV. PROPOSED APPROACH

To design an efficient image super-resolution model, we build upon a Vision Mamba-based 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 and power usage, 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 image, eight mini-patches of 64×64 are extracted randomly. To further preprocess the data, random horizontal and vertical flips and rotations are applied. Then, each of these 64×64 patches is converted into a vector of 60 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 as a 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
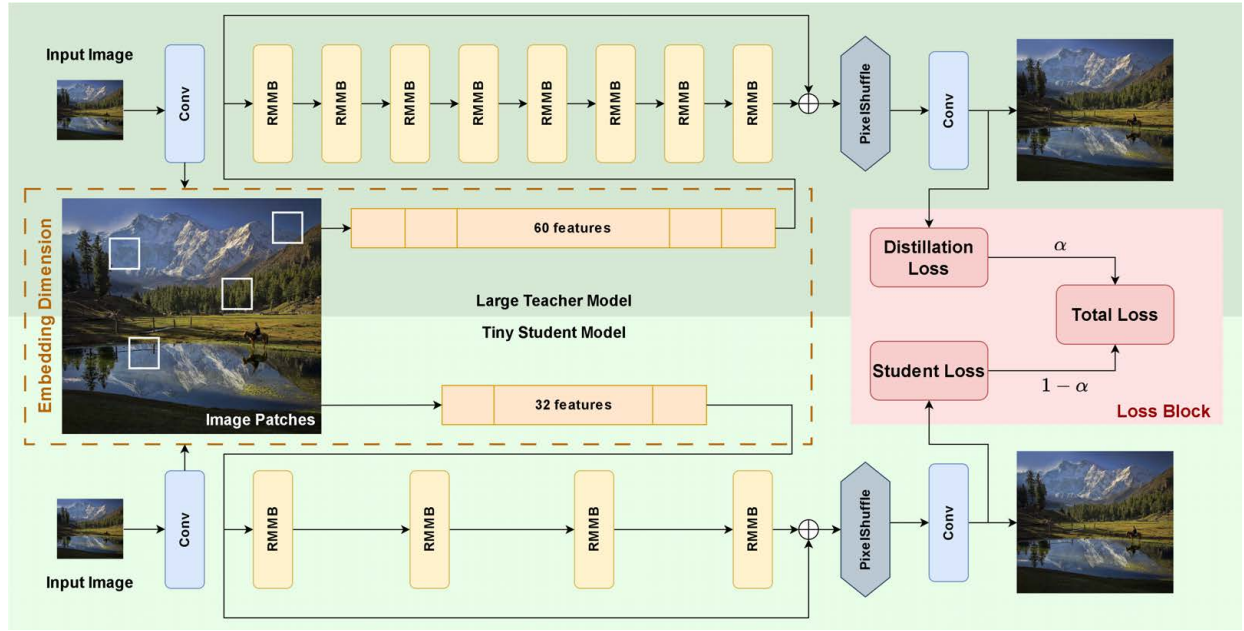
Fig. 2. High-Level overview of MambaLiteSR process: Low-resolution input image (64×64) is preprocessed and fed into the knowledge distillation process, generating the high-resolution output (256×256). Weighted distillation and student losses enable the student model to learn efficiently under teacher supervision. Proper embedding dimension, which determines the feature vector size resulting from the image patches, makes it suitable for edge devices.

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, 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 that the input to both the 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, α 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 α, 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. As a result, 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 α 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, we used the DF2K (DIV2K + Flickr2K) [47] dataset, which consists of 3450 high-resolution

images and their corresponding low-resolution versions. The low-resolution images are generated by downscaling each image by factors of 2, 3, and 4. To train the proposed model, we consider a scale of 4 to do the experiments as it presents a more challenging problem and requires more parameters to learn. Moreover, for the validation set, a subset of the DIV2K dataset consisting of 100 images was used. For testing the accuracy and performance of the proposed approach, we utilized four standard benchmark datasets: Set5 [48], Set14 [49], BSD100 [50], and Urban100 [51].

Implementation Details. For faster processing during training, images are cropped into smaller patches. Specifically, from each LR image eight mini-patches of 64×64 and from each HR image eight corresponding mini-patches of 256×256 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 2×10^{-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 training experiments are conducted on Lambda GPU Server [52], 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 [32], but with different values and hyperparameters. For this purpose, the embedding dimension is reduced to 32. To assess the performance of different values of α parameter in knowledge distillation, an experiment was conducted comparing the result of changing this value over 1000 iterations for each of the teacher and student models.

For the inference power consumption measurements, we deploy the model on the embedded Nvidia Jetson Orin Nano board [36] as shown in Figure 3.(a). Power measurements experiments are repeated over 1000 samples, and the average is calculated. The Jetson Orin Nano Developer Kit with 8GB of memory features a hexa-core ARM Cortex-A78AE CPU with 1.5 GHz frequency and a 512-core NVIDIA Ampere GPU with 625 MHz. This board is optimized for AI tasks like matrix multiplications and deep learning inference, making it ideal for edge AI applications [53], besides boards such as Jetson Orin Nano [54]. We converted the software models to ONNX format and optimized them into TensorRT representations for GPU-accelerated inference. The frequency was set to 612 MHz with the power mode being set to 7 Watt. During inference, the tegrastats utility [55] was used to measure instantaneous power.

B. Results

In this section, we compare the proposed MambaLiteSR model with the state-of-the-art methods, including eSR [26], ESPCN [39], and FSRCNN [40], in terms of performance and power consumption. While ESPCN and FSRCNN were not
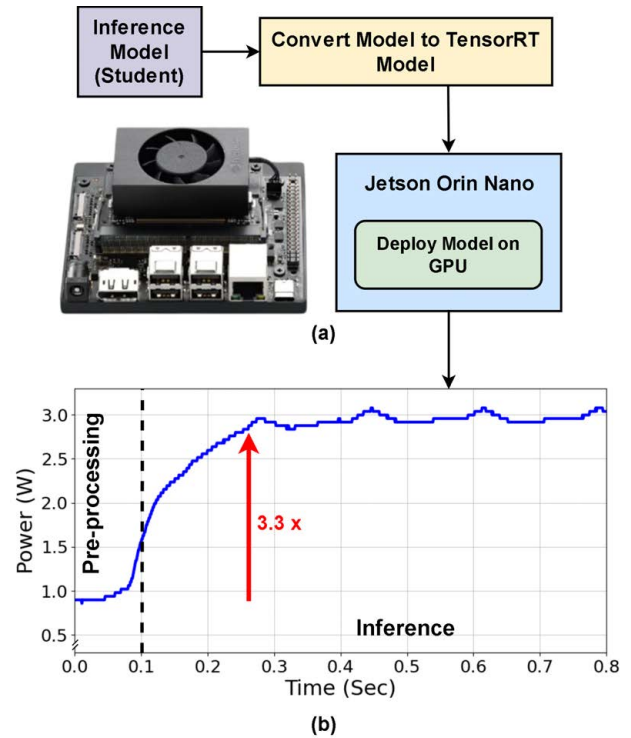

Fig. 3. (a) For measuring dynamic power on the embedded NVIDIA Jetson Orin Nano, the student onnx model is converted to TensorRT format and then the measurement starts. (b) The plot shows the instantaneous power usage over time for 1000 samples, extracted using the tegrastats utility [55].

originally designed for edge deployment, they were evaluated on edge devices in the experiments conducted in eSR. Next, we present the model optimization and training results to analyze the impact of various variables of knowledge distillation and Mamba architecture on training and overall performance. Moreover, we compare the proposed MambaLiteSR with the baseline [32]. Finally, we demonstrate the performance of the proposed tiny student model by comparing the model output with the ground truth image to demonstrate its ability to achieve similar quality.

Figure 3.(b) illustrates the power consumption results on the embedded NVIDIA Jetson Orin Nano during inference. Experiments are done on the proposed MambaLiteSR tiny student model to evaluate the power and PSNR/SSIM results, as summarized in Table I. The idle power consumption was measured at approximately 0.9 W, and then, the system initiated image pre-processing. Following this, the model execution and image input processing caused a $3.3 \times$ increase in power consumption. The deployment results of the proposed MambaLiteSR model, compared against state-of-the-art methods on edge devices, are detailed in Table I. The power measurement was averaged over 1000 samples, resulting in a fast single image latency of 14 ms. The results demonstrate that the proposed approach improved the power consumption by up to 58% while maintaining SSIM and PSNR performance comparable to existing methods. Note that the compared methods are all CNN-based. While CNNs

Algorithm	Power [mWatts]	Set5 [48]		Set14 [49]		BSD100 [50]		Urban100 [51]	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
eSR [13]	7100	30.62	0.860	27.48	0.751	26.93	0.714	24.42	0.718
eSR - fast [13]	3867	28.64	0.806	26.12	0.712	26.13	0.684	23.28	0.668
ESPCN [39]	6952	30.57	0.858	27.50	0.752	26.92	0.715	24.42	0.718
FSRCNN [40]	4795	30.16	0.845	27.19	0.742	26.74	0.707	24.09	0.702
MambaLiteSR (Ours)	**2957**	28.28	0.837	25.39	0.727	25.20	0.693	22.57	0.686

extract structured features, generative AI models such as ours can create more detailed and realistic images by generating missing information. Such models handle complex transformations better, especially for large scaling factors, and produce sharper textures by learning perceptual features instead of just minimizing pixel differences.

To reduce the number of parameters to make the model suitable for edge deployment the embedding dimension of 60 and 32 to the teacher and student model is applied respectively, which leads to the 15% reduction in the number of parameters while keeping the performance acceptable with comparison with the Baseline [32]. 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 iterations as shown in Table II.

Model	Parameters	iterations	PSNR
DVMSR [32]	424k	500,000	32.19
MambaLiteSR	**370k (15% smaller)**	2500	28.28

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 1.7x less energy consumption besides good performance.

The performance of different values of rank 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 corresponding GPU power usage during training on the Lambda GPU Server [52], 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 42% less than the one with $rank = 30$.

To further reduce the model size, knowledge distillation is applied. According to the Equation 7, α is a key parameter

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

in the loss function. Therefore, experiments were made with different values of α to evaluate their performance over 1000 iterations and choose the most appropriate one. The result of different values of α can be seen in table IV with their corresponding PSNR.

α	0.2	0.4	0.6	0.8
PSNR	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. After that, the student model is put to training for 2500 number of iterations, the same as the teacher model. Then, the experiments on α 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 α, 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. Moreover, if one of these weights decreases significantly, it indicates that the model is either not effectively learning or has reverted to learning from scratch. Note that in the baseline, α was set to 0.5, meaning the impact of the teacher model and ground truth is the same, which as was mentioned, results in inferior performance.
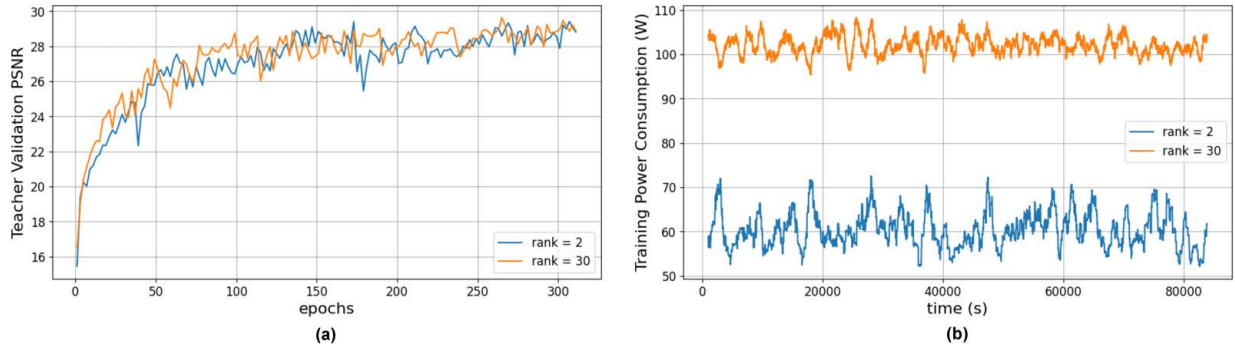
Fig. 4. Comparison between MambaLiteSR teacher model training when $rank = 2$ and $rank = 30$ over 1500 iterations: (a) depicts the validation PSNR. (b) depicts the moving average of GPU power usage every 100 seconds with a window size of 100. At runtime, the larger rank requires more FLOPs, drawing 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 [52] using wandb [57] dashboard.
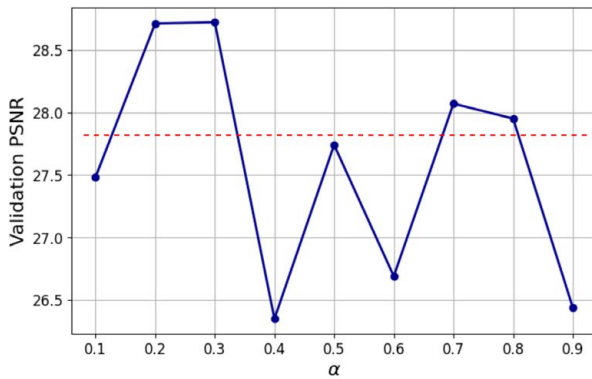

Fig. 5. Model performance on the validations set based on the changes in α, suggesting a potential inconsistency in gradient values between the learned teacher and the ground truth. The reported PSNRs are after 1000 iterations of training for teacher and student models.
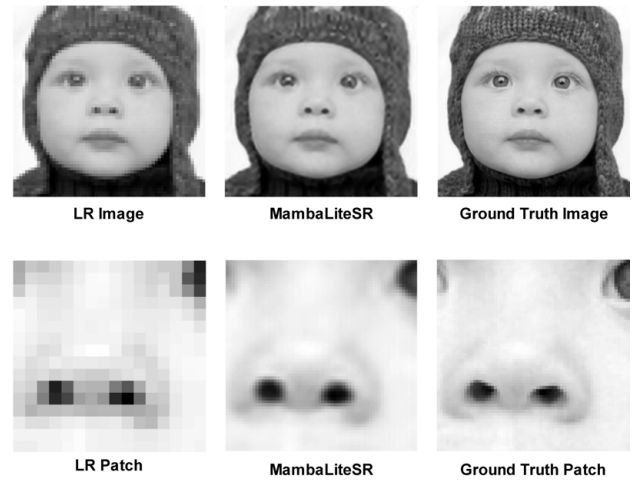

Fig. 6. Demonstration of LR image, MambaLiteSR output, and ground truth image, as well as a random patch cropped from the LR input and its corresponding MambaLiteSR output and ground truth patch.

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; a random patch from the LR image is taken, was given as an input to the model, and the output as well as the ground truth is shown. A complete process consists of dividing the LR image into patches, passing each of them through the model, and stitching these patches together. It can be seen that MambaLiteSR is successful in generating a super-resolution image similar to its 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 42% 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 lower power consumption compared to the state-of-the-art edge super-resolution models, while maintaining similar performance. By combining low-rank approximation, knowledge distillation, and efficient embedding strategies, MambaLiteSR offers a practical solution for generative AI applications on edge devices.

VII. ACKNOWLEDGMENT

R EFERENCES

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

[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] B. Prakash, T. Oates, T. Mohsenin, *Using LLMs for Augmenting Hierarchical Agents with Common Sense Priors*, The International FLAIRS Conference Proceedings. Vol. 37. 2024.

[4] B. Prakash, T. Oates, T. Mohsenin, *LLM Augmented Hierarchical Agents*, arXiv preprint arXiv:2311.05596, 2023.

[5] 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.

[6] H. Pourmehrani, J. Bahrami, P. Nooralinejad, H. Pirsiavash, N. Karimi *FAT-RABBIT: Fault-Aware Training towards Robustness AgainstBit-flip Based Attacks in Deep Neural Networks*, 2024 IEEE International Test Conference (ITC), 2024.

[7] M. Navardi, E. Humes, T. Mohsenin, *MetaTinyML: End-to-End Metareasoning Framework for TinyML Platforms*, IEEE Embedded Systems Letters, 2024.

[8] U. Kallakuri, E. Humes, T. Mohsenin, *Resource-Aware Saliency-Guided Differentiable Pruning for Deep Neural Networks*, Proceedings of the Great Lakes Symposium on VLSI 2024.

[9] H. Rashid, T. Mohsenin, *TinyM2 Net-V3: Memory-Aware Compressed Multimodal Deep Neural Networks for Sustainable Edge Deployment*, arXiv preprint arXiv:2405.12353, 2024.

[10] Z. Bao, E. Kanjo, S. Banerjee, H. Rashid, T. Mohsenin, *Decentralised Resource Sharing in TinyML: Wireless Bilayer Gossip Parallel SGD for Collaborative Learning*, arXiv preprint arXiv:2501.04817, 2025.

[11] T. Manjunath, M. Navardi, P. Dixit, T. Mohsenin *ReProHRL: Towards multi-goal navigation in the real world using hierarchical agents*, arXiv preprint arXiv:2308.08737, 2023.

[12] M. Navardi T. Mohsenin *Mlae2: Metareasoning for latency-aware energy-efficient autonomous nano-drones*, 2023 IEEE International Symposium on Circuits and Systems (ISCAS), 2023.

[13] P. Navarrete Michelini, Y. Lu, and X. Jiang, *Edge-SR: Super-Resolution for the Masses*, WACV, 2022.

[14] Q. Wang et al., *An Intelligent Co-Scheduling Framework for Efficient Super-Resolution on Edge Platforms*, IEEE IoT Journal, 2024.

[15] A. Mazumder, F. Safavi, M. Rahnemoonfar, T. Mohsenin, *Reg-tune: A regression-focused fine-tuning approach for profiling low energy consumption and latency*, IEEE IoT Journal, 2024.

[16] M. Navardi, E. Humes, T. Manjunath, T. Mohsenin *MetaE2RL: Toward Meta-Reasoning for Energy-Efficient Multigoal Reinforcement Learning With Squeezed-Edge You Only Look Once*, IEEE Micro, 2023.

[17] E. Humes, M. Navardi, T. Mohsenin, *Squeezed Edge YOLO: Onboard Object Detection on Edge Devices*, NeurIPS, 2023.

[18] J. Wright, A. Y. Yang, A. Ganesh, S. Shankar Sastry, and Y. Ma, *Robust Face Recognition via Sparse Representation*, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 2009.

[19] C. Dong, C. C. Loy, K. He, and X. Tang, *Image Super-Resolution Using Deep Convolutional Networks*, ECCV, 2014.

[20] J. Kim, J. K. Lee, and K. M. Lee, *Accurate Image Super-Resolution Using Very Deep Convolutional Networks*, CVPR, 2016.

[21] C. Dong, C. C. Loy, and X. Tang, *Accelerating the super-resolution convolutional neural network*, ECCV, 2016.

[22] 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.

[23] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, *Enhanced Deep Residual Networks for Single Image Super-Resolution*, CVPRw, 2017.

[24] K. Zhang, W. Zuo, S. Gu, and L. Zhang, *Learning Deep CNN Denoiser Prior for Image Restoration*, CVPR, 2017.

[25] W. Zhang, L. Pan, K. Xu, G. Li, and Y. Lv, *Efficient self-calibrated and hierarchical refinement network for lightweight super-resolution*, ELSEVIER, 2024.

[26] 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.

[27] 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.

[28] A. Gu and T. Dao, *Mamba: Linear-Time Sequence Modeling with Selective State Spaces*, arXiv preprint arXiv:2312.00752, 2023.

[29] L. Zhu, B. Liao, Q. Zhang, X. Wang, W. Liu, and X. Wang, *Vision Mamba: Efficient Visual Representation Learning with Bidirectional State Space Model*, ICML, 2024.

[30] A. Hatamizadeh and J. Kautz, *MambaVision: A Hybrid Mamba-Transformer Vision Backbone*, arXiv preprint arXiv:2407.08083, 2024.

[31] 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.

[32] X. Lei, W. Zhang, and W. Cao, *DVMSR: Distilled Vision Mamba for Efficient Super-Resolution*, ICCV, 2024.

[33] 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*, arXiv preprint arxiv.org/abs/2410.10140, 2024.

[34] 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.

[35] 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.

[36] NVIDIA, *NVIDIA Jetson Orin*, https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/

[37] M. Jahnavia, 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.

[38] Y. Chen and T. Pock, *Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration*, arXiv preprint arXiv:1508.02848, 2015.

[39] W. Shi et al., *Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network*, CVPR, 2016.

[40] C. Dong, C. Change Loy, and X. Tang, *Accelerating the super–resolution convolutional neural network*, ECCV, 2016.

[41] Y. Li, Y. Zhang, et al. Ntire 2023 challenge on efficient super-resolution: Methods and results, CVPRW, 2023.

[42] 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.

[43] P. Mathur et al., *A Real-Time Super-Resolution for Surveillance Thermal Cameras Using Optimized Pipeline on Embedded Edge Device*, IEEE AVSS, 2021.

[44] 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.

[45] R. Y. Tsai and T. S. Huang, *Multiframe image restoration and registration*, Advances in Computer Vision and Image Processing, 1984.

[46] S. Baker and T. Kanade, *Limits on Super-Resolution and How to Break Them*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002.

[47] R. Timofte, et al., *Ntire 2017 challenge on single image super-resolution: Methods and results*, CVPRW, 2017.

[48] M. Bevilacqua, A. Roumy, C. Guillemot, and M. Alberi-Morel, *Low-complexity single-image super-resolution based on nonnegative neighbor embedding*, BMVC, 2012.

[49] R. Zeyde, M. Elad, and M. Protter, *On single image scale-up using sparse-representations*, Curves and Surfaces, 2010.

[50] 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.

[51] J. Huang, A. Singh, and N. Ahuja, *Single image super-resolution from transformed selfexemplars*, CVPR, 2015.

[52] Lambda Labs, *Lambda GPU Server*, https://lambdalabs.com/

[53] M. R. Shaharear, A. N. Mazumder, T. Mohsenin, *ViT-Reg: Regression-Focused Hardware-Aware Fine-Tuning for ViT on tinyML Platforms*, IEEE Design & Test, 2024.

[54] NVIDIA, *Jetson Orin Nano Super Developer Kit*, https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-orin/nano-super-developer-kit/

[55] Tegrastats Utility, https://docs.nvidia.com/jetson/archives/r34.1/ DeveloperGuide/text/AT/JetsonLinuxDevelopmentTools/TegrastatsUtility.html

[56] NVIDIA, *NVIDIA Jetson AGX Xavier GPU*, https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-agx-xavier/

[57] wandb.ai, *Weights & Biases*, https://wandb.ai/