Pruning Overparameterized Multi-Task Networks for Degraded Web Image Restoration

Thomas Katraouras
Dept. of Electrical and Computer Engineering
University of Thessaly
Volos, Greece
tkatraouras@uth.gr

Dimitrios Rafailidis
Dept. of Electrical and Computer Engineering
University of Thessaly
Volos, Greece
draf@uth.gr

***Abstract*—Image quality is a critical factor in delivering visually appealing content on web platforms. However, images often suffer from degradation due to lossy operations applied by online social networks (OSNs), negatively affecting user experience. Image restoration is the process of recovering a clean high-quality image from a given degraded input. Recently, multi-task (all-in-one) image restoration models have gained significant attention, due to their ability to simultaneously handle different types of image degradations. However, these models often come with an excessively high number of trainable parameters, making them computationally inefficient. In this paper, we propose a strategy for compressing multi-task image restoration models. We aim to discover highly sparse subnetworks within overparameterized deep models that can match or even surpass the performance of their dense counterparts. The proposed model, namely MIR-L, utilizes an iterative pruning strategy that removes low-magnitude weights across multiple rounds, while resetting the remaining weights to their original initialization. This iterative process is important for the multi-task image restoration model's optimization, effectively uncovering "winning tickets" that maintain or exceed state-of-the-art performance at high sparsity levels. Experimental evaluation on benchmark datasets for the deraining, dehazing, and denoising tasks shows that MIR-L retains only 10% of the trainable parameters while maintaining high image restoration performance. Our code, datasets and pre-trained models are made publicly available at https://github.com/Thomkat/MIR-L.**

***Index Terms*—Multi-task, Web Image Restoration, Pruning.**

I. Introduction

Image quality is a critical factor in delivering visually appealing content across web platforms, where images are essential to user engagement and experience. However, images on the web frequently undergo lossy operations applied by online social networks (OSNs), such as JPEG compression and format conversion [1]–[4]. These operations result in noticeable degradation, with higher compression ratios correlated with greater degradation [5]. Such reductions in image quality can negatively impact user experience, as lower visual quality reduces the perceived value of online content [6]. Overcoming the problem of degraded images is important for improving user experience on the web.

Image restoration is a fundamental task in computer vision that aims to recover high-quality images from degraded versions. This degradation can be caused by factors such as noise [7], rain [8], haze [9], motion blur [10], low resolution [11], or compression artifacts [12]. Image restoration seeks to enhance the visual quality and clarity of images, making them more suitable for various applications. Recent image restoration models utilize deep learning techniques [8]–[10], [13]–[21] to reconstruct clean images by learning complex mappings from degraded inputs to their high-quality equivalents. These models offer breakthrough performance compared to conventional restoration methods [22]–[25] and are widely used in fields such as medical imaging [26], astronomy [27] and aerial imaging [28], where accurate and visually enhanced images are essential for analysis and decision-making. In addition, image restoration not only improves visual fidelity, but it also promotes high performance in tasks such as object detection [29], [30].

Image restoration models have been designed to handle specific tasks, such as denoising [7], [17], [20], deraining [8], [18], [31], dehazing [9], [16], [32], deblurring [10], [13], [33] and super-resolution [11], [15], [21]. However, real-world images often suffer from multiple types of degradation. To address this, the focus has shifted towards multi-task (all-in-one) image restoration models [14], [34]–[36], handling various types of degradation within a single framework, without requiring any prior knowledge of the degradation. Multi-task models provide an efficient and unified solution for real-world image restoration problems, as they reduce the overhead of deploying separate networks for individual degradations.

However, despite their effectiveness, multi-task image restoration models often require a significantly high number of trainable parameters, as demonstrated in our experiments in Section IV. This leads to substantial computational costs and memory demands, making running these models feasible only on high-end machines, rather than consumer-grade devices. Additionally, this limits their usability in real-time applications, such as client-side web image restoration. To address the complexity issue, researchers have explored several techniques to reduce the size and computational requirements of deep learning models, while maintaining their performances. Model compression methods such as one-shot pruning [37]–[39], knowledge distillation [40]–[42] and parameter sharing [43], [44] have been applied to deep neural networks. One-shot pruning removes redundant parameters in a single step, knowledge distillation transfers knowledge from a large teacher

Accepted at WI-IAT 2025

model to a smaller student model and parameter sharing reduces redundancy by reusing parameters across different tasks or layers, effectively lowering the model size. These techniques have shown promising results in reducing the size of complex deep learning models. However, achieving a balance between preserving the model's ability to handle diverse degradations in the image restoration problem and minimizing redundant parameters still remains a challenge.

One promising approach to model compression is the Lottery Ticket Hypothesis (LTH), which suggests that within a large, overparameterized neural network, there are small subnetworks—referred to as "winning tickets"—that can be trained in isolation to achieve comparable performance to the original model [45]. It has been studied in image classification [46], [47] and natural language processing [48], [49]. The effectiveness of lottery tickets in reducing the size of multi-task image restoration models has not been thoroughly explored. Investigating LTH in this context could reveal whether certain subnetworks consistently perform well across multiple image restoration tasks, potentially enabling more efficient all-in-one solutions for handling diverse image degradations.

In this paper we propose the MIR-L model based on lottery tickets for compressing multi-task image restoration models while maintaining the performance high. Specifically, we make the following contributions:

- We propose a LTH-based pruning algorithm designed for multi-task image restoration models, focusing on deraining, dehazing and denoising tasks. The algorithm iteratively removes the smallest-magnitude weights and resets the remaining weights to their original initialization, seamlessly integrating to the multi-task image restoration models' optimization process.
- We explore both layer-wise and global pruning strategies to assess their effectiveness in discovering sparse networks. We show that global pruning is capable of finding very sparse winning tickets, while layer-wise pruning diminishes performance.

We conduct experiments on benchmark datasets for the deraining, dehazing, and denoising tasks, comparing our proposed MIR-L with baseline pruning methods and state-of-the-art multi-task image restoration models. Our results demonstrate that the sparse networks of MIR-L reduce the number of trainable parameters by up to 90% compared to the original dense models and outperform baseline pruning methods. In many cases, these sparse networks match or even exceed the performance of dense, state-of-the-art multi-task image restoration models, confirming that our approach effectively discovers efficient and highly sparse subnetworks—winning tickets—for multi-task image restoration.

The remainder of this paper is structured as follows: Section II provides an overview of pruning techniques. Section III presents our proposed method, the architecture of the multi-task image restoration model, the pruning strategy and the MIR-L optimization algorithm. Section IV provides the experimental evaluation, showcasing results on various datasets. Finally, Section V concludes the paper, summarizing key findings and discussing potential future directions.

II. Preliminaries

Pruning is a technique in deep learning used to reduce the number of parameters in a neural network by removing certain connections. The goal is to create an efficient model with reduced memory and computational costs, while preserving the performance high. Formally, given a dense network $f(x;\theta)$, pruning identifies and removes a subset of parameters $\theta_p \subset \theta$, yielding a sparse network $f(x;\theta \setminus \theta_p)$ [50]. Below, we outline preliminaries of existing pruning strategies.

1) Magnitude Pruning.: Magnitude-based pruning is a widely used pruning strategy that removes parameters having the smallest absolute values, assuming they contribute less to the network's performance and can be removed with minimal impact. Formally, given a trained dense network $f(x;\theta)$ and a threshold τ, a parameter θ_i is pruned if $|\theta_i| < \tau$, setting $\theta_i = 0$ for such parameters. The resulting pruned network is represented as $f(x;\theta')$, where $\theta' = \theta \setminus \{\theta_i : |\theta_i| < \tau\}$ [51], [52].

2) One-shot Pruning.: One-shot pruning is a pruning strategy where the network parameters are pruned once after the initial training phase. A fixed percentage $p\%$ of the parameters are removed based on a pruning criterion, e.g., magnitude-based, resulting in a sparse network with pruned parameters set to zero [37], [38].

3) Iterative Pruning: Iterative pruning is an approach to network sparsification, where pruning is performed in multiple rounds rather than in a single step. This method iteratively prunes a percentage $p\%$ of the parameters and optimizes the network after each pruning step [53], [54].

III. Proposed Method

A. Multi-Task Image Restoration Model

1) Tasks: A multi-task (all-in-one) blind image restoration model is designed to recover clean images from degraded inputs without prior knowledge of the degradation type. Specifically, it handles the following image restoration tasks: I. *Deraining*: removes rain streaks and artifacts; II. *Dehazing*: removes haze and fog; III. *Denoising*: reduces unwanted noise caused by low-light conditions, sensor imperfections, or compression artifacts.

2) Architecture: A multi-task image restoration model takes a degraded image $\tilde{\mathbf{I}} \in \mathbb{R}^{H\times W\times C}$ as input, where $H \times W$ is the spatial resolution, and $C = 3$ represents the RGB color channels [36]. This image has undergone an unknown degradation $\mathbf{D}$. The model produces a restored image $\mathbf{I} \in \mathbb{R}^{H\times W\times C}$. The model follows a UNet-style network architecture [19] with transformer blocks [55] in both the encoding and decoding stages. Initially, low-level features $\mathbf{F}_0 \in \mathbb{R}^{H\times W\times C}$ are extracted from $\tilde{\mathbf{I}}$ by applying a 3×3 convolution: $\mathbf{F}_0 = \mathrm{Conv}_{3\times 3}(\tilde{\mathbf{I}})$. These features go through a four-level hierarchical encoder-decoder, where each level increases channel capacity while reducing spatial resolution, ultimately generating low-resolution latent features $\mathbf{F}_l \in \mathbb{R}^{\frac{H}{8}\times\frac{W}{8}\times 8C}$ [36].

The decoder gradually upsamples and refines $\mathbf{F}_l$, leading to the final clean output image $\mathbf{I}$. During decoding, the model incorporates sequential prompt blocks at multiple levels to inject degradation-aware information. Each prompt block consists of two components: a Prompt Generation Module (PGM) and a Prompt Interaction Module (PIM). Given N learnable prompt components $\mathbf{P}_c \in \mathbb{R}^{N\times\hat{H}\times\hat{W}\times\hat{C}}$ and input features $\mathbf{F}_l \in \mathbb{R}^{\hat{H}\times\hat{W}\times\hat{C}}$, the prompt block produces refined features $\hat{\mathbf{F}}_\mathbf{l} = \text{PIM}(\text{PGM}(\mathbf{P_c}, \mathbf{F_l}), \mathbf{F_l})$. The PGM learns an adaptive prompt $\mathbf{P}$ conditioned on both $\mathbf{F}_l$ and $\mathbf{P}_c$. In particular, the PGM aggregates spatial information from $\mathbf{F}_l$ using global average pooling, followed by a 1×1 convolution and softmax to produce prompt weights: $\mathbf{w} = \text{Softmax}(\text{Conv}_{1\times1}(\text{GAP}(\mathbf{F}_l)))$, where $\mathbf{w} \in \mathbb{R}^{1\times1\times N}$. These weights $\mathbf{w}$ determine the contribution of each prompt component $\{\mathbf{P}_{c_1}, \ldots, \mathbf{P}_{c_N}\}$ in a weighted sum. The resulting combination is then refined by a 3×3 convolution: $\mathbf{P} = \text{Conv}_{3\times3}(\sum_{i=1}^{N} w_i\, \mathbf{P}_{c_i})$. The PIM fuses $\mathbf{P}$ with $\mathbf{F}_l$ by concatenating along the channel dimension: $\mathbf{F}_{\text{concat}} = \text{Concat}(\mathbf{F}_l, \mathbf{P})$. A transformer block $\mathbf{T}$ processes $\mathbf{F}_{\text{concat}}$ to incorporate degradation-specific information, followed by two consecutive 1×1 and 3×3 convolutions: $\hat{\mathbf{F}}_\mathbf{l} = \text{Conv}_{3\times3}\big(\text{Conv}_{1\times1}(\mathbf{T}(\mathbf{F}_{\text{concat}}))\big)$. Finally, $\hat{\mathbf{F}}_\mathbf{l}$ propagate through the decoder, leading to the reconstructed image $\mathbf{I}$. The L_1 loss function is used to minimize the absolute differences between the restored and ground truth images, defined as $L_1 = \frac{1}{HWC}\sum_{i=1}^{HWC}|\mathbf{I}_{\text{GT}i} - \mathbf{I}_i|$, where $\mathbf{I}_{\text{GT}} \in \mathbb{R}^{H\times W\times C}$ is the ground truth image [36]. The optimization is performed using the Adam optimizer.

B. Lottery Ticket Hypothesis

The LTH proposes that within a dense, randomly-initialized neural network, there is a sparse subnetwork—referred to as a winning ticket—that can be trained in isolation to achieve performance comparable to the original network [45].

Definition 1 (Winning Ticket)**.** A winning ticket, denoted as $f_w(x;\theta_w)$, is a sparse subnetwork within a dense, randomly-initialized neural network $f(x;\theta)$ with initial parameters $\theta_0 \sim \mathcal{D}_\theta$, such that when trained in isolation from its original initialization θ_0, it satisfies $a_{f_w} \geq a_f$, $j_{f_w} \leq j_f$, and $|\theta_w| \ll |\theta|$; where a_{f_w} and a_f denote the test accuracies achieved by f_w and f, respectively, j_{f_w} and j_f denote the number of training iterations required to reach minimum validation loss, and $|\theta_w|$ and $|\theta|$ denote the number of parameters in the winning ticket and the original network, respectively.

Proposition 1. *Consider a dense feed-forward neural network $f(x;\theta)$ with initial parameters $\theta_0 \sim \mathcal{D}_\theta$. Let $m \in \{0,1\}^{|\theta|}$ be a binary mask that identifies the active connections in the subnetwork. The Lottery Ticket Hypothesis predicts that a mask m does exist such that training $f(x; m\odot\theta_0)$, where $\odot$ denotes element-wise multiplication, results in a winning ticket $f_w(x;\theta_w)$; where $\theta_w \equiv m\odot\theta_0$.*

1) Layer-wise Pruning: Layer-wise pruning is a strategy where pruning is applied independently to each layer of the network. A fixed percentage $p\%$ of the smallest-magnitude weights within each layer are pruned, ensuring that sparsity is uniformly distributed across all layers. The output layer is pruned at half the rate, $\frac{p}{2}\%$, since it typically contains far fewer parameters compared to other layers. Pruning it too aggressively can lead to diminishing returns much earlier.

2) Global Pruning: Global pruning is a strategy where a fixed percentage $p\%$ of the smallest-magnitude weights are pruned across the entire network, rather than on a per-layer basis. This approach is particularly effective in deeper networks, where layers can have significantly different numbers of parameters. By pruning globally, bottlenecks caused by uniformly pruning smaller layers are avoided. As a result, global pruning can identify smaller winning tickets compared to layer-wise pruning, especially in networks with imbalanced layer sizes.

Algorithm 1 MIR-L Optimization Algorithm

Input: 1. Initial dense network $f(\tilde{\mathbf{I}};\theta_0)$, 2. Pruning rate p, 3. Number of training epochs j, 4. Number of warmup epochs j_w, 5. Batch size B, 6. Training samples $\mathcal{X}_{train}$, 7. Initial learning rate η_{start}, 8. Base learning rate η_{base}, 9. Minimum learning rate η_{min}, 10. Target sparsity level S

Output: Trained sparse network $f(\tilde{\mathbf{I}};\theta)$

$m \leftarrow \mathbf{1}^{|\theta_0|}$ ▷ Initialize binary mask
$\theta \leftarrow m \odot \theta_0$ ▷ Set initial parameters
while $\frac{\|m\|_0}{|\theta_0|} < S$ **do**
  for epoch = $1 \rightarrow j$ **do**
    Compute learning rate η_t using Linear Warmup Cosine Annealing:

$$\eta_t = \begin{cases} \eta_{\text{start}} + \frac{t}{j_w - 1}(\eta_{\text{base}} - \eta_{\text{start}}), & 0 \leq t < j_w \\ \eta_{\text{min}} + \frac{1}{2}(\eta_{\text{base}} - \eta_{\text{min}})\left(1 + \cos\left(\frac{(t-j_w)\pi}{j-j_w}\right)\right), & j_w \leq t \leq j \end{cases}$$

    for step = $1 \rightarrow \frac{|\mathcal{X}_{train}|}{B}$ **do**
      $\mathbf{I}_B = \{f(\tilde{\mathbf{I}}_i;\theta)\}_{i=1}^{B}, \quad \tilde{\mathbf{I}}_i \sim X_{\text{train}}$ ▷ Forward pass
      $\mathcal{L}_{L1} \leftarrow \frac{1}{HWC}\sum_{i=1}^{HWC}|\mathbf{I}_{\text{GT}i} - \mathbf{I}_{B_i}|$ ▷ Compute the reconstruction loss
      $\nabla_\theta \mathcal{L}_{L1} \leftarrow \frac{\partial \mathcal{L}_{L1}}{\partial \theta}$ ▷ Backward pass
      $\nabla_\theta \mathcal{L}_{L1} \leftarrow m \odot \nabla_\theta \mathcal{L}_{L1}$ ▷ Mask gradients of pruned weights
      $\theta \leftarrow \theta - \eta_t \nabla_\theta \mathcal{L}_{L1}$ ▷ Parameter update via Adam
      $\theta \leftarrow m \odot \theta$ ▷ Apply sparsity mask to updated weights
    end for
  end for
  Determine pruning threshold τ as the p-th percentile of $|m\odot\theta|$, i.e., $\tau = \text{Quantile}_p\big(|m \odot \theta|\big)$
  $m' \leftarrow \mathbb{1}(|m \odot \theta| \geq \tau)$ ▷ Calculate new mask
  $\theta \leftarrow m' \odot \theta_0$ ▷ Prune and reset to initial values
  $m \leftarrow m'$ ▷ Update mask
end while
return Final sparse model $f(\tilde{\mathbf{I}};\theta)$

C. MIR-L Optimization Algorithm

The proposed MIR-L model optimizes the multi-task image restoration model (Section III-A) and prunes it with the LTH to obtain a sparse yet equally or more effective multi-task image restoration model. MIR-L, optimized with an L_1

reconstruction loss, is iteratively trained and pruned, until the target sparsity level is reached. Algorithm 1 provides a formal outline of the model optimization and pruning process. Firstly, the dense network parameters and a binary mask are initialized (lines 1–2). Next, the network is trained for j epochs with a learning rate schedule that includes linear warmup followed by cosine annealing (lines 4–14). After each training step's backward pass, the gradients and weights are masked, to ensure they remain zeroed. After each training cycle, a pruning threshold τ is determined based on a pruning rate p, the mask is updated, the network is pruned and remaining weights are reset to their initial values (lines 15–18). This procedure is repeated until the target sparsity S is reached, resulting in a final sparse model.

Note that for layer-wise pruning, the threshold τ is determined for each layer independently, whereas for global pruning τ is determined across all layers.

IV. Experimental Evaluation

A. Datasets

We evaluate our MIR-L model, as well as the baselines following the evaluation protocol of [34]–[36]. Specifcially, for image denoising, we use a combination of the BSD400 [56] and WED [57] datasets for training. BSD400 consists of 400 training images and the WED dataset consists of 4,744 images. Due to training resourse constraints, we randomly selected 5% of the images of each dataset for training. From the clean images, we generate the noisy images by adding Gaussian noise with different noise levels $\sigma \in \{15, 25, 50\}$. Testing is performed on the Color BSD68 [58] and Urban100 [59] datasets consisting of 68 and 100 images, respectively. For image deraining, we use the Rain100L [60] dataset, which consists of 200 rainy-clean image pairs for training and 100 rainy-clean image pairs for testing. We randomly selected 10% of the original pairs for training. For image dehazing, we use the OTS [61] dataset for training, which consists of 72,135 images. We randomly selected 3% of the original pairs. Testing is performed on the SOTS [61] dataset, consisting of 500 hazy-clean image pairs. In the all-in-one setting (covering both training and testing), we combine the aforementioned datasets across denoising, deraining, and dehazing. This approach enables a unified evaluation of our method under a single model across multiple restoration tasks. All the datasets are publicly available for reproducibility purposes at https://github.com/Thomkat/MIR-L.

B. Evaluation Protocol

To evaluate the performance of our model, we need to specify appropriate metrics that objectively compare different models. In image restoration tasks, Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) are commonly used to assess the quality of restored images [8], [16], [19], [20], [35], [36], [62]. These metrics provide insight into the reconstruction fidelity and perceptual similarity of the restored images.

1) Peak Signal-to-Noise Ratio (PSNR): measures the ratio between the maximum possible power of a signal and the power of the noise that affects its representation. A higher PSNR value indicates better image quality, as it implies a lower level of distortion in the restored image. The PSNR is calculated as follows:

$$PSNR = 10 \cdot \log \left(\frac{MAX^2}{MSE} \right) \tag{1}$$

where *MAX* is the maximum possible pixel value, i.e., 255 for an 8-bit image and *MSE* (Mean Squared Error) represents the average squared differences between corresponding pixels of the original and restored images.

2) Structural Similarity Index Measure (SSIM): quantifies the perceived visual quality of an image by considering structural information, luminance, and contrast. A higher SSIM value indicates better perceptual quality and structural similarity to the reference image. The SSIM is calculated as follows:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{2}$$

where μ_x and μ_y are the mean intensities of images x and y, σ_x^2 and σ_y^2 are their variances, σ_{xy} is the covariance, and C_1 and C_2 are small constants to avoid instability.

While PSNR is useful for measuring absolute reconstruction fidelity, SSIM aligns better with human visual perception. Therefore, both PSNR and SSIM provide complementary insights into the performance of our model.

C. Experimental Setup

1) Implementation Details: All the experiments were performed on the NVIDIA A40 GPU, using PyTorch version 2.5.1. The model was trained for 120 epochs (15 warmup epochs) with a batch size of 8. Optimization was performed using the Adam optimizer with an L_1 loss function and a learning rate of 2×10^{-4}. The target sparsity level S is 90%, which corresponds to 15 pruning steps, and pruning rate p was set to 20%. During training, the input images were randomly cropped into patches of size 64×64. To improve generalization, random horizontal and vertical flips were applied to the training data. Smaller datasets were artificially expanded by duplicating their images, while random augmentations ensured variation, allowing the model to perceive them as distinct and maintain a balanced training process.

2) Examined Models:

- **MSPFN**[1] [8]: A multi-scale progressive fusion network for image deraining, using cross-scale and intra-scale information with recurrent refinement.
- **EPDN**[2] [16]: An enhanced Pix2pix Dehazing Network reframing dehazing as image-to-image translation, with a GAN-based enhancer module.
- **AirNet**[3] [35]: An all-in-one image restoration network

[1]https://github.com/kuijiang94/MSPFN
[2]https://github.com/ErinChen1/EPDN
[3]https://github.com/XLearning-SCU/2022-CVPR-AirNet

TABLE I
EXAMINED IMAGE RESTORATION MODELS

Model	Single-Task	Multi-Task	Task			Sparse
			Deraining	Dehazing	Denoising	
MSPFN [8]	✓		✓			
EPDN [16]	✓			✓		
FFDNet [20]	✓				✓	
AirNet [35]		✓	✓	✓	✓	
Restormer [19]		✓	✓	✓	✓	
MPRNet [62]		✓	✓	✓	✓	
AdaIR [34]		✓	✓	✓	✓	
PromptIR [36]		✓	✓	✓	✓	
PIR-OSM I		✓	✓	✓	✓	✓
PIR-OSM II		✓	✓	✓	✓	✓
PIR-OSR I		✓	✓	✓	✓	✓
PIR-OSR II		✓	✓	✓	✓	✓
MIR-L-LW		✓	✓	✓	✓	✓
MIR-L-G		✓	✓	✓	✓	✓

for unknown degradations via contrastive-based encoding and degradation-guided recovery.

- **Restormer**[4] [19]: A transformer-based restoration network for high-resolution images, utilizing attention for long-range dependencies.
- **FFDNet**[5] [20]: A convolutional neural network (CNN) for image denoising using downsampled sub-images and a tunable noise-level map for spatially varying noise.
- **MPRNet**[6] [62]: A multi-stage all-in-one image restoration network that progressively refines spatial details.
- **AdaIR**[7] [34]: An adaptive all-in-one image restoration network that mines low- and high-frequency features and modulates them bidirectionally for progressive correction.
- **PromptIR**[8] [36]: An all-in-one blind image restoration model that generalizes to various unknown degradation types and levels by using prompt-based learning to encode degradation-specific information, dynamically guiding the restoration network.
- **PIR-OSM I**: A pruned version of the model described in Section III-A (one-shot, magnitude-based), obtained by removing 30% of the smallest weights post-training and fine-tuning for an additional 5% of training epochs.
- **PIR-OSM II**: A variant of PIR-OSM, removing 70% of the smallest weights.
- **PIR-OSR I**: A pruned version of the model described in Section III-A (one-shot, random), obtained by randomly removing 30% of weights post-training and fine-tuning for an additional 5% of training epochs.
- **PIR-OSR II**: A variant of PIR-OSR, randomly removing 70% of the weights.
- **MIR-L-LW (Layer-wise Pruning)**: Our proposed model based on the LTH with layer-wise pruning.
- **MIR-L-G (Global Pruning)**: Our proposed model based on the LTH with global pruning.

In Table I we present an overview of the examined image restoration models. To ensure a fair comparison, we retrain all the aforementioned models using their publicly available implementations and the datasets described in Section IV-A. All the models are trained with an input patch size of 64×64.

[4]https://github.com/swz30/Restormer
[5]https://github.com/cszn/FFDNet
[6]https://github.com/swz30/MPRNet
[7]https://github.com/c-yn/AdaIR
[8]https://github.com/va1shn9v/PromptIR

D. Experimental Results

Table II compares our MIR-L against conventional one-shot pruning baselines and model baselines on the following single-task settings: Table IIa reports deraining results on the Rain100L dataset, Table IIb reports dehazing results on the SOTS dataset and Table IIc reports denoising results on the BSD68 and Urban100 datasets. In the single task setting, separate models are trained for each individual degradation (Table I). Multi-task models have a higher number of trainable parameters than single-task models but they achieve better restoration performance. Although one-shot magnitude (PIR-OSM I. & II.) and random pruning (PIR-OSR I. & II.) reduce the trainable parameters, they show a steep drop in performance at high sparsity levels, expressed by fewer trainable parameters. The proposed MIR-L-LW and MIR-L-G drastically reduce the parameters, down to 4.7M, while preserving the performance high. Our strategy achieves superior performance by gradually pruning the model and resetting the remaining weights to their original values. This process allows the optimization to relearn the weights and recover any lost performance by modifying the relationships between the surviving weights. In subsequent rounds, less important weights are pruned again, ensuring that the most critical parts of the network are preserved. By contrast, conventional one-shot pruning methods remove a large portion of weights all at once, leaving little opportunity for the model to adjust and fully recover the lost performance. MIR-L-G consistently outperforms MIR-L-LW in all settings, demonstrating that global pruning more effectively discovers winning tickets in large networks.

Table III compares our MIR-L against conventional one-shot pruning baselines and model baselines on the multi-task setting: Table IIIa reports deraining results on the Rain100L dataset, Table IIIb reports dehazing results on the SOTS dataset and Table IIIc reports denoising results on the BSD68 and Urban100 datasets. In the multi-task (all-in-one) setting, a model is trained to simultaneously handle multiple degradations. Similarly to the single-task settings, one-shot magnitude (PIR-OSM I. & II.) and random pruning (PIR-OSR I. & II.) reduce the parameters, while their performance degrades significantly as sparsity increases. The proposed MIR-L-LW and MIR-L-G achieve greater performance than PIR-OSM and PIR-OSR, using only 4.7M parameters, an approximate 87% reduction compared to the dense model's 35.6M parameters on average, corresponding to a compression rate of x7.57. Similarly to the single-task setting, MIR-L-G outperforms MIR-L-LW, with the former achieving restoration performance that reaches or exceeds state-of-the-art both in terms of PSNR and SSIM.

Figure 1 reports PSNR when varying the number of trainable parameters for layer-wise and global pruning. The x-axis indicates the number of trainable parameters, where a

TABLE II

COMPARISON OF SINGLE-TASK RESULTS FOR (A) DERAINING, (B) DEHAZING, AND (C) DENOISING. THE BEST RESULTS ARE SHOWN IN BOLD, AND THE SECOND-BEST ARE UNDERLINED. OUR MIR-L-LW AND MIR-L-G MODELS DRASTICALLY REDUCE TRAINABLE PARAMETERS WHILE REACHING PERFORMANCE SIMILAR TO DENSE BASELINE MODELS.

(a) Derain Model (Rain100L)

Method	PSNR/SSIM	Trainable Parameters
MSPFN [8]	25.85/0.8118	21M
AirNet [35]	28.77/0.8867	7.6M
Restormer [19]	30.09/0.9114	26.1M
PromptIR [36]	**35.13/0.9683**	35.6M
PIR-OSM I.	34.75/0.9640	25.6M
PIR-OSM II.	25.52/0.8140	12.4M
PIR-OSR I.	25.74/0.8158	25.6M
PIR-OSR II.	25.69/0.8142	12.4M
MIR-L-LW	32.14/0.9395	**4.7M**
MIR-L-G	34.72/0.9652	**4.7M**

(b) Dehaze Model (SOTS)

Method	PSNR/SSIM	Trainable Parameters
EPDN [16]	24.57/0.9367	22.9M
AirNet [35]	22.13/0.9228	7.6M
Restormer [19]	25.32/0.9432	26.1M
PromptIR [36]	26.76/0.9556	35.6M
PIR-OSM I.	26.55/0.9525	25.6M
PIR-OSM II.	18.75/0.8612	12.4M
PIR-OSR I.	20.70/0.8838	25.6M
PIR-OSR II.	17.25/0.8282	12.4M
MIR-L-LW	26.53/0.9533	**4.7M**
MIR-L-G	**27.62/0.9609**	**4.7M**

(c) Denoise Model (BSD68 & Urban100)

Dataset	Method	$\sigma = 15$ PSNR/SSIM	$\sigma = 25$ PSNR/SSIM	$\sigma = 50$ PSNR/SSIM	Trainable Parameters
BSD68	FFDNet [20]	33.42/0.9240	30.93/0.8768	27.81/0.7838	**494K**
	AirNet [35]	33.89/0.9324	31.28/**0.8883**	28.09/**0.7997**	7.6M
	Restormer [19]	33.64/0.9243	31.22/0.8796	**28.12**/0.7896	26.1M
	PromptIR [36]	**33.97/0.9330**	**31.32**/0.8876	28.08/0.7961	35.6M
	PIR-OSM I.	33.72/0.9272	31.07/0.8777	27.97/0.7873	25.6M
	PIR-OSM II.	27.73/0.7058	23.54/0.5287	17.88/0.2980	12.4M
	PIR-OSR I.	29.14/0.839	28.06/0.7848	24.61/0.5993	25.6M
	PIR-OSR II.	25.44/0.6159	21.15/0.4397	15.59/0.2344	12.4M
	MIR-L-LW	33.01/0.9208	30.40/0.8688	27.06/0.7554	4.7M
	MIR-L-G	33.85/0.9298	31.30/0.8862	28.07/0.7962	4.7M
Urban100	FFDNet [20]	32.65/0.9316	30.57/0.9017	27.51/0.8367	**494K**
	AirNet [35]	34.30/**0.9476**	31.99/**0.9219**	28.72/**0.8661**	7.6M
	Restormer [19]	**34.36**/0.9449	**32.05**/0.9183	**28.83**/0.8608	26.1M
	PromptIR [36]	33.90/0.9433	31.51/0.9139	28.23/0.8522	35.6M
	PIR-OSM I.	33.46/0.9363	31.10/0.9049	28.04/0.8442	25.6M
	PIR-OSM II.	27.61/0.7262	23.54/0.5750	17.95/0.3696	12.4M
	PIR-OSR I.	27.08/0.8389	26.31/0.7858	23.57/0.616	25.6M
	PIR-OSR II.	25.52/0.6500	21.28/0.4998	15.75/0.3080	12.4M
	MIR-L-LW	32.00/0.9250	29.54/0.8846	26.04/0.7910	4.7M
	MIR-L-G	33.71/0.9392	31.50/0.9131	28.26/0.8529	4.7M

larger pruning step corresponds to fewer trainable parameters. We observe that MIR-L-G consistently outperforms MIR-L-LW as sparsity increases, primarily because global pruning selectively removes redundant weights across all layers, avoiding bottlenecks in thinner layers and thus preserving the subnetwork's overall representational capacity. In the single-task settings, both pruning strategies initially maintain high PSNR values, but as pruning becomes more aggressive, layer-wise pruning shows a significant performance drop compared to global pruning. An exception is deraining, where global pruning shows a large drop at higher sparsity levels compared to layer-wise pruning. This occurs because weights essential for deraining performance are pruned by the global magnitude pruning criterion during these steps. In the multi-task (all-in-one) setting, we observe a similar trend: global pruning not only maintains a higher PSNR across all tasks, but performance improves in all tasks as parameters are reduced, whereas layer-wise pruning shows a steep performance drop at higher sparsity levels.

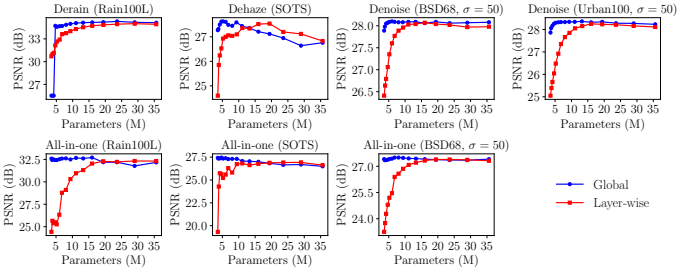

Fig. 1. PSNR vs. trainable parameter count across progressive pruning steps. The x-axis denotes the number of trainable parameters, where a larger pruning step corresponds to fewer trainable parameters. MIR-L-G consistently maintains high performance as step (sparsity) increases, while MIR-L-LW experiences a sharp drop at higher pruning levels.

V. CONCLUSION

This paper proposes a pruning strategy for multi-task image restoration models based on lottery tickets (MIR-L), focusing

TABLE III

COMPARISON OF MULTI-TASK (ALL-IN-ONE) RESULTS FOR (A) DERAINING, (B) DEHAZING, AND (C) DENOISING. THE BEST RESULTS ARE SHOWN IN BOLD, AND THE SECOND-BEST ARE UNDERLINED. OUR MIR-LW AND MIR-L-G MODELS ACHIEVE PERFORMANCE SIMILAR TO OR HIGHER THAN STATE-OF-THE-ART, WITH SUBSTANTIALLY FEWER TRAINABLE PARAMETERS THAN DENSE MODELS.

(a) Rain100L Dataset

Method	PSNR/SSIM	Trainable Parameters
MPRNet [62]	27.64/0.8477	39.5M
AirNet [35]	27.83/0.8809	<u>7.6M</u>
PromptIR [36]	<u>32.17/0.9372</u>	35.6M
AdaIR [34]	25.90/0.8409	28.8M
PIR-OSM I.	31.85/0.9272	25.6M
PIR-OSM II.	26.59/0.8365	12.4M
PIR-OSR I.	25.39/0.8101	25.6M
PIR-OSR II.	26.02/0.8171	12.4M
MIR-L-LW	25.49/0.8125	**4.7M**
MIR-L-G	**32.43/0.9425**	**4.7M**

(b) SOTS Dataset

Method	PSNR/SSIM	Trainable Parameters
MPRNet [62]	24.34/0.9350	39.5M
AirNet [35]	22.41/0.8738	<u>7.6M</u>
PromptIR [36]	26.49/0.9535	35.6M
AdaIR [34]	<u>27.09/0.9575</u>	28.8M
PIR-OSM I.	26.43/0.9524	25.6M
PIR-OSM II.	17.54/0.8399	12.4M
PIR-OSR I.	18.02/0.8404	25.6M
PIR-OSR II.	16.54/0.8180	12.4M
MIR-L-LW	25.63/0.9446	**4.7M**
MIR-L-G	**27.45/0.9591**	**4.7M**

(c) BSD68 Dataset

Method	$\sigma = 15$ PSNR/SSIM	$\sigma = 25$ PSNR/SSIM	$\sigma = 50$ PSNR/SSIM	Trainable Parameters
MPRNet [62]	32.15/0.8976	30.10/0.8552	27.36/0.7647	39.5M
AirNet [35]	32.79/0.9167	30.30/0.8602	27.07/0.7437	<u>7.6M</u>
PromptIR [36]	33.50/0.9247	30.79/0.8734	27.41/0.7667	35.6M
AdaIR [34]	<u>33.52/0.9250</u>	<u>30.82/0.8747</u>	**27.48**/<u>0.7695</u>	28.8M
PIR-OSM I.	32.97/0.9148	30.29/0.8556	26.76/0.7202	25.6M
PIR-OSM II.	25.91/0.6374	21.66/0.4605	16.06/0.2475	12.4M
PIR-OSR I.	25.38/0.6382	21.47/0.4628	16.00/0.2504	25.6M
PIR-OSR II.	24.62/0.5965	20.56/0.4218	15.14/0.2220	12.4M
MIR-L-LW	31.22/0.8731	28.55/0.7946	24.80/0.6166	**4.7M**
MIR-L-G	**33.53/0.9269**	**30.83/0.8772**	**27.48/0.7736**	**4.7M**

on the deraining, dehazing, and denoising tasks. To deal with the overparameterization of multi-task image restoration models, we presented an iterative pruning strategy that removes low-magnitude weights in multiple rounds, while resetting the surviving weights to their initial values. The proposed MIR-L optimization algorithm discovers sparse "winning tickets" capable of matching or surpassing the performance of their dense counterparts, at a fraction of trainable parameters. Our experiments demonstrated that MIR-L effectively reduces the number of trainable parameters by up to 90% across both single-task and multi-task settings, while maintaining high performance on benchmark datasets. This model size reduction and low computational requirements are beneficial for web platforms, allowing faster delivery of high-quality images and improved user experience even on less powerful client devices. In future work, exploring more sophisticated pruning criteria, such as SynFlow [63], or expanding the implementation to image restoration tasks commonly used in real-time applications, such as super-resolution [15], [21], may offer further improvements in both efficiency and image restoration accuracy.

REFERENCES

[1] M.-R. Ra, R. Govindan, and A. Ortega, "P3: Toward {Privacy-Preserving} photo sharing," in *10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, 2013, pp. 515–528.

[2] J. Ning, I. Singh, H. V. Madhyastha, S. V. Krishnamurthy, G. Cao, and P. Mohapatra, "Secret message sharing using online social media," in *2014 IEEE Conference on Communications and Network Security*. IEEE, 2014, pp. 319–327.

[3] W. Sun, J. Zhou, R. Lyu, and S. Zhu, "Processing-aware privacy-preserving photo sharing over online social networks," in *Proceedings of the 24th ACM international conference on Multimedia*, 2016, pp. 581–585.

[4] W. Sun, J. Zhou, S. Zhu, and Y. Y. Tang, "Robust privacy-preserving image sharing over online social networks (osns)," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 14, no. 1, pp. 1–22, 2018.

[5] J. Hu, S. Song, and Y. Gong, "Comparative performance analysis of web image compression," in *2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, 2017, pp. 1–5.

[6] A. Asif, H. He, A. Khan, and M. Shafiq, "Assessment of quality of experience (qoe) of image compression in social cloud computing," *Multiagent and Grid Systems*, vol. 14, no. 2, pp. 125–143, 2018.

[7] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE TIP*, 2017.

[8] K. Jiang, Z. Wang, P. Yi, C. Chen, B. Huang, Y. Luo, J. Ma, and J. Jiang, "Multi-scale progressive fusion network for single image deraining," in *CVPR*, 2020.

[9] B. Li, X. Peng, Z. Wang, J.-Z. Xu, and D. Feng, "Aod-net: All-in-one dehazing network," in *ICCV*, 2017.

[10] K. Kim, S. Lee, and S. Cho, "Mssnet: Multi-scale-stage network for single image deblurring," in *ECCV*, 2022, pp. 524–539.

[11] H. Li, Y. Yang, M. Chang, S. Chen, H. Feng, Z. Xu, Q. Li, and Y. Chen, "Srdiff: Single image super-resolution with diffusion probabilistic models," *Neurocomputing*, vol. 479, pp. 47–59, 2022.

[12] M. Ehrlich, L. Davis, S.-N. Lim, and A. Shrivastava, "Quantization guided jpeg artifact correction," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*. Springer, 2020, pp. 293–309.

[13] S.-J. Cho, S.-W. Ji, J.-P. Hong, S.-W. Jung, and S.-J. Ko, "Rethinking

coarse-to-fine approach in single image deblurring," in *ICCV*, 2021, pp. 4641–4650.
[14] Q. Fan, D. Chen, L. Yuan, G. Hua, N. Yu, and B. Chen, "A general decoupled learning framework for parameterized image operators," *IEEE TPAMI*, 2021.
[15] Z. Lu, J. Li, H. Liu, C. Huang, L. Zhang, and T. Zeng, "Transformer for single image super-resolution," in *CVPR*, 2022, pp. 457–466.
[16] Y. Qu, Y. Chen, J. Huang, and Y. Xie, "Enhanced pix2pix dehazing network," in *CVPR*, 2019.
[17] C. Tian, Y. Xu, and W. Zuo, "Image denoising using deep cnn with batch renormalization," *Neural Networks*, vol. 121, pp. 461–473, 2020.
[18] W. Wei, D. Meng, Q. Zhao, Z. Xu, and Y. Wu, "Semi-supervised transfer learning for image rain removal," in *CVPR*, 2019.
[19] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, "Restormer: Efficient transformer for high-resolution image restoration," in *CVPR*, 2022.
[20] K. Zhang, W. Zuo, and L. Zhang, "Ffdnet: Toward a fast and flexible solution for cnn based image denoising," *IEEE TIP*, 2018.
[21] W. Zhang, W. Zhao, J. Li, P. Zhuang, H. Sun, Y. Xu, and C. Li, "Cvanet: Cascaded visual attention network for single image super-resolution," *Neural Networks*, vol. 170, pp. 622–634, 2024.
[22] W. Dong, L. Zhang, G. Shi, and X. Wu, "Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization," *IEEE TIP*, vol. 20, no. 7, pp. 1838–1857, 2011.
[23] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," in *CVPR*, 2009.
[24] C. Liu, R. Szeliski, S. B. Kang, C. L. Zitnick, and W. T. Freeman, "Automatic estimation and removal of noise from a single image," *IEEE TPAMI*, vol. 30, no. 2, pp. 299–314, 2007.
[25] R. Timofte, V. De, and L. Van Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *ICCV*, 2013.
[26] Z. Yang, H. Chen, Z. Qian, Y. Yang, H. Zhang, D. Zhao, B. Wei, and Y. Xu, "All-in-one medical image restoration via task-adaptive routing," in *MICCAI*, 2024.
[27] P. Jia, R. Ning, R. Sun, X. Yang, and D. Cai, "Data-driven image restoration with option-driven learning for big and small astronomical image data sets," *MNRAS*, vol. 501, no. 1, pp. 291–301, 2021.
[28] M. Y. Hossain, M. M. H. Rakib, S. Rajit, I. R. Nijhum, and R. M. Rahman, "Adaptive and automatic aerial image restoration pipeline leveraging pre-trained image restorer with lightweight fully convolutional network," *ESWA*, vol. 259, p. 125210, 2025.
[29] S. Sun, W. Ren, T. Wang, and X. Cao, "Rethinking image restoration for object detection," *Advances in Neural Information Processing Systems*, vol. 35, pp. 4461–4474, 2022.
[30] J. Wang, M. Xu, H. Xue, Z. Huo, and F. Luo, "Joint image restoration for object detection in snowy weather," *IET Computer Vision*, 2024.
[31] R. Yasarla and V. M. Patel, "Uncertainty guided multi-scale residual learning-using a cycle spinning cnn for single image de-raining," in *CVPR*, 2019.
[32] Y. Dong, Y. Liu, H. Zhang, S. Chen, and Y. Qiao, "Fd-gan: Generative adversarial networks with fusion-discriminator for single image dehazing," *AAAI*, vol. 34, pp. 10 729–10 736, 2020.
[33] X. Ji, Z. Wang, S. Satoh, and Y. Zheng, "Single image deblurring with row-dependent blur magnitude," in *ICCV*, 2023, pp. 12 269–12 280.
[34] Y. Cui, S. W. Zamir, S. Khan, A. Knoll, M. Shah, and F. S. Khan, "Adair: Adaptive all-in-one image restoration via frequency mining and modulation," in *ICLR*, 2025.
[35] B. Li, X. Liu, P. Hu, Z. Wu, J. Lv, and X. Peng, "All-in-one image restoration for unknown corruption," in *CVPR*, 2022.
[36] V. Potlapalli, S. W. Zamir, S. H. Khan, and F. Shahbaz Khan, "Promptir: Prompting for all-in-one image restoration," in *NeurIPS*, 2023.
[37] T. Chen, B. Ji, T. Ding, B. Fang, G. Wang, Z. Zhu, L. Liang, Y. Shi, S. Yi, and X. Tu, "Only train once: A one-shot neural network training and pruning framework," *Advances in Neural Information Processing Systems*, vol. 34, pp. 19 637–19 651, 2021.
[38] P. Hu, X. Peng, H. Zhu, M. M. S. Aly, and J. Lin, "Opq: Compressing deep neural networks with one-shot pruning-quantization," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 35, no. 9, 2021, pp. 7780–7788.
[39] S. Khaki and K. N. Plataniotis, "The need for speed: Pruning transformers with one recipe," *arXiv preprint arXiv:2403.17921*, 2024.
[40] Y. Jiang, J. Nawala, F. Zhang, and D. R. Bull, "Compressing deep image super-resolution models," *PCS*, pp. 1–5, 2023.
[41] B. Murugesan, S. Vijayarangan, K. Sarveswaran, K. Ram, and M. Sivaprakasam, "Kd-mri: A knowledge distillation framework for image reconstruction and image restoration in mri workflow," *arXiv*, vol. abs/2004.05319, 2020.
[42] P. Wang, H. Huang, X. Luo, and Y. Qu, "Data-free learning for lightweight multi-weather image restoration," in *ISCAS*, 2024.
[43] A. Dudhane, O. Thawakar, S. W. Zamir, S. Khan, F. Khan, and M.-H. Yang, "Dynamic pre-training: Towards efficient and scalable all-in-one image restoration," *arXiv*, vol. abs/2404.02154, 2024.
[44] X. Zhou, H. Huang, Z. Wang, and R. He, "Ristra: Recursive image super-resolution transformer with relativistic assessment," *IEEE TMM*, vol. 26, pp. 6475–6487, 2024.
[45] J. Frankle and M. Carbin, "The lottery ticket hypothesis: Finding sparse, trainable neural networks," in *ICLR*, 2019.
[46] T. Chen, J. Frankle, S. Chang, S. Liu, Y. Zhang, M. Carbin, and Z. Wang, "The lottery tickets hypothesis for supervised and self-supervised pre-training in computer vision models," in *CVPR*, 2021, pp. 16 306–16 316.
[47] C. Ma, J. Jia, J. Huang, and X. Wang, "Exploration and optimization of lottery ticket hypothesis for few-shot image classification task," in *IPEC*, 2024, pp. 221–227.
[48] T. Chen, J. Frankle, S. Chang, S. Liu, Y. Zhang, Z. Wang, and M. Carbin, "The lottery ticket hypothesis for pre-trained bert networks," *NeurIPS*, vol. 33, pp. 15 834–15 846, 2020.
[49] H. Yu, S. Edunov, Y. Tian, and A. S. Morcos, "Playing the lottery with rewards and multiple languages: lottery tickets in rl and nlp," *arXiv preprint arXiv:1906.02768*, 2019.
[50] Y. LeCun, J. Denker, and S. Solla, "Optimal brain damage," in *NeurIPS*, 1989.
[51] K. Belay, "Gradient and mangitude based pruning for sparse deep neural networks," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 36, no. 11, 2022, pp. 13 126–13 127.
[52] G. Li, C. Qian, C. Jiang, X. Lu, and K. Tang, "Optimization based layer-wise magnitude-based pruning for dnn compression." in *IJCAI*, vol. 330, 2018, pp. 2383–2389.
[53] B. Geng, M. Yang, F. Yuan, S. Wang, X. Ao, and R. Xu, "Iterative network pruning with uncertainty regularization for lifelong sentiment classification," in *Proceedings of the 44th International ACM SIGIR conference on Research and Development in Information Retrieval*, 2021, pp. 1229–1238.
[54] L. Yu, X. Li, Y. Li, T. Jiang, Q. Wu, H. Fan, and S. Liu, "Dipnet: Efficiency distillation and iterative pruning for image super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 1692–1701.
[55] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *NeurIPS*, 2017.
[56] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2011.
[57] K. Ma, Z. Duanmu, Q. Wu, Z. Wang, H. Yong, H. Li, and L. Zhang, "Waterloo exploration database: New challenges for image quality assessment models," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 1004–1016, 2017.
[58] D. Martin, 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," in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 2, 2001, pp. 416–423 vol.2.
[59] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 5197–5206.
[60] W. Yang, R. T. Tan, J. Feng, Z. Guo, S. Yan, and J. Liu, "Joint rain detection and removal from a single image with contextualized deep networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 6, pp. 1377–1393, 2020.
[61] B. Li, W. Ren, D. Fu, D. Tao, D. Feng, W. Zeng, and Z. Wang, "Benchmarking single-image dehazing and beyond," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 492–505, 2019.
[62] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, "Multi-stage progressive image restoration," in *CVPR*, 2021.
[63] H. Tanaka, D. Kunin, D. L. Yamins, and S. Ganguli, "Pruning neural networks without any data by iteratively conserving synaptic flow," *Advances in neural information processing systems*, vol. 33, pp. 6377–6389, 2020.