Abstract

Super-resolution (SR) has achieved great success due to the development of deep convolutional neural networks (CNNs). However, as the depth and width of the networks increase, CNN-based SR methods have been faced with the challenge of computational complexity in practice. Moreover, most of them train a dedicated model for each target resolution, losing generality and increasing memory requirements. To address these limitations we introduce OverNet, a deep but lightweight convolutional network to solve SISR at arbitrary scale factors with a single model. We make the following contributions: first, we introduce a lightweight recursive feature extractor that enforces efficient reuse of information through a novel recursive structure of skip and dense connections. Second, to maximize the performance of the feature extractor we propose a reconstruction module that generates accurate high-resolution images from overscaled feature maps and can be independently used to improve existing architectures. Third, we introduce a multi-scale loss function to achieve generalization across scales. Through extensive experiments, we demonstrate that our network outperforms previous state-of-the-art results in standard benchmarks while using fewer parameters than previous approaches.

1. Introduction

Single image super-resolution (SISR) is the task of reconstructing a high resolution image (HR) from its low-resolution version (LR). Since obtaining a HR image from its LR counterpart is an ill-posed problem, it is necessary to learn the original data distribution to provide the most likely solutions.

Convolutional neural networks (CNNs) have recently become the main workhorse to tackle SISR [5]. Moreover, thanks to the increase in capacity of CNNs in depth [21] and width [43], their performance has greatly improved. Despite their remarkable performance, most deep networks still have some drawbacks. Firstly, the increase of depth and width has dramatically increased the computational demands and memory consumption. This makes modern architectures less applicable in practice, such as mobile and embedded vision applications. Secondly, as the network depth increases, the low-level feature information gradually disappears in the successive non-linear operations to produce the output. However, these low-level features is crucial for the network to reconstruct high quality images.

Aside from the aforementioned problems, another desired ability is to upsample images to arbitrary scales using a single model. Current state-of-the-art SISR models such as RDN [38], ESPCNN [25] and EDSR [21], only consider SR at certain integer scale factors (×2, ×3, ×4) and treat each super-resolution scale as an independent task. They then train a different specialized model for each, which is not practical for mobile applications.

To address these problems, we propose Overscaling Network (OverNet), a novel lightweight method for SISR. OverNet consists of two main parts: a lightweight feature extractor and a reconstruction module called Overscaling module (OSM). The feature extractor follows a novel recursive framework of skip and dense connections to reduce low-level feature degradation. The OSM is a new inductive bias which generates accurate SR image by internally constructing an overscaled intermediate representation of the output features. Finally, to solve the problem of reconstruc-
tion at arbitrary scale factors, we introduce a novel multi-scale loss by downsampling the output at multiple super resolution factors and we minimize the reconstruction error in all of them. Our main contributions can be summarized as follows:

- A lightweight recursive feature extractor, which results in improved performance over other state-of-the-art models, even those having an order of magnitude more parameters.

- An Overscaling Module (OSM) that generates overscaled maps from which HR images can be accurately recovered at arbitrary scales. This module boosts the reconstruction accuracy efficiently with respect to its number of parameters. Additionally, we demonstrate that integrating this module into existing state-of-the-art models improves on their original performance.

- A novel multi-scale loss function for SISR, that allows the simultaneous training of all scale factors using a single model. As a result, the model is able to maintain accurate reconstruction results across scales.

2. Related Work

2.1. Super-Resolution Networks

Recently, deep learning-based models have shown dramatic improvements in the SISR task. Dong et al. [5] first presented SRCNN, a CNN to predict super-resolved images. However, SRCNN has a large number of operations compared to its depth, since the network operates by initially upsampling LR images and subsequently refining them. In contrast to the SRCNN, FSCRNN [6] and ESPCN [25] only upsample images at the output of the network, which leads to a reduction in the number of operations compared to the SRCNN.

Despite the higher capacity of deep neural networks, the aforementioned methods have settled for shallow models because of the difficulty in training. VDSR [13] and IR-CNN [35] improved the performance by increasing the network depth, using stacked convolutions with residual connections. Lim et al. [21] further expanded the network size and improved the residual block by removing batch normalization layers. Moreover, Ahn et al. [11] proposed a cascading residual network (CARN) which used ResNet blocks [8] to learn the relationship between low-resolution input and high-resolution output. Later, Ledig et al. [18] introduced the SRResNet and are further improved in [31] by introducing dense connections. More recently, Zhang et al. [38] and Liu et al. [22] also used dense and residual connections in RDN and RFANet to utilize information from all the feature hierarchy. DBPN [7] and SRFBN [20] proposed architectures comprises of a series of up and down sampling layers that are densely connected with each other. These methods achieved significant improvement over conventional SR methods and indicate the effectiveness of residual learning. In this work, we show that it is possible to increase the effectiveness of these residual connections by using a recursive architecture structure.

Another issue of deep learning-based SR is how to reduce the parameters and number of operations to make it effective in mobile applications. For instance, DRCN [14] was the first to apply recursive algorithm to SISR to reduce the number of parameters by reusing them multiple times. Tai et al. [28] improved DRCN by combining the recursive and residual network schemes in order to achieve better performance with even fewer parameters. They also introduced a deep memory network to solve the problem of long-term dependencies [29]. On the other hand, LapSRN [16] employs a pyramidal framework to increase the image size gradually. By doing so, LapSRN effectively performs SR on extremely low-resolution cases. However, these methods use very deep networks to compensate for the loss of parameters and hence, they require heavy computing resources. Therefore, we aim to build a model that is lightweight in both size and computation.

2.2. Image Reconstruction in SISR

One of the most important stages of SISR is reconstruction, which consists of generating HR images based on high-level features extracted from a low-dimensional space.

Interpolation is a commonly used method in SR networks, such as SRCNN [5], VDSR [13] and DRRN [28], to resize the original LR image to the target size as the input of a CNN model for SR reconstruction. However, the computational operations are greatly increased due to the large size of the input image. Therefore, FSR-CNN [6] and SRDenseNet [18] directly adopted the original LR image without upsampling as input for CNN, in which a transposed convolution layer was added to implement the final upsampling reconstruction [30]. This method greatly reduces unnecessary computational overhead. Furthermore, ESPCN [25] proposed a method called pix2pix [2] to overcome the problem of the checkerboard effect in transposed convolution. pix2pix has been widely used in recently proposed SR models, such as EDSR [21], WDSR [33] and CARN [11]. However, these methods cannot manage multi-scale training.

Few works tackle super-resolution at different scale factors, and those that do treat the problem as independent tasks, i.e. a model is trained for each scale. Lim et al. [21] proposed MDSR, the first multi-scale SISR model, which has different image processing blocks and upscaling modules for each integer scale factor. Later, Li et al. [19] proposed a multi-scale residual network (MSRNet). They use multi-path convolution layers with different ker-
nel sizes to extract multi-scale spatial features. Therefore, these methods require vast amounts of computational resources. Recently, Meta-SR [10] introduced an upsampling module based on meta-learning to solve super-resolution at arbitrary scale factors with a single model through a weight prediction technique. However, this method must predict a large number of convolution weights for each target pixel, this prediction is not efficient, and finally the results may be unstable [32].

3. Proposed Overscaling Network

In this section, we describe the main components of our architecture (as depicted in Figure 1) and the novel loss function used for training.

Algorithm 1 Overscaling network forward step. Given a LR image and a set of output scales, OverNet produces an HR reconstruction for each scale. Learnable parameters are omitted to improve readability

```
function OVERNET(LR image $I^{LR}$, target scales $S$)
    # Compute features with the CNN
    $h = \mathcal{H}(I^{LR})$
    # Overscaling module
    $\hat{I}^{HR} = \mathcal{O}(h)$
    # Output
    for $s$ in $S$ do
        $\hat{I}^{HR}_{s} =$ bicubic($\hat{I}^{HR}$, scale = $s$)
    end for
    return $\{\hat{I}^{HR}_{s}, s \in S\}$
end function
```

Problem formulation. Algorithm 1 formulates the main steps involved in our pipeline. Given a set of high resolution images and their downscaled versions $\{I^{HR}, I^{LR}\}$, the goal of SISR is to find a function $\mathcal{F} : LR \rightarrow HR$ that maps LR images to their original HR version. This is an ill-posed problem, since there are multiple possible HR images corresponding to a single LR image. However, it is possible to learn the most likely reconstruction by parametrizing $\mathcal{F}$ over a set of parameters $\theta$, and finding the most likely $\theta$ given some criterion $\mathcal{L}$:

$$\theta^* = \arg \min_{\theta} \sum_{i} \mathcal{L}(\mathcal{F}(I^{LR}, \theta), I^{HR})$$

We chose $\mathcal{L}$ to be the $L1$ distance, since we empirically obtained superior PSNR results compared to $L2$. In this work $\mathcal{F}$ is composed of two parts: (i) a feature extractor $\mathcal{H}$:

$$h = \mathcal{H}(I^{LR}, \theta_h)$$

with parameters $\theta_h$, and (ii) the overscaling module $\mathcal{O}$ where

$$\hat{I}^{HR} = \mathcal{O}(h, \theta_o)$$

with $\theta_o$ the parameters used in this operation, and $\hat{I}^{HR}$ the reconstructed image. These two parts are described in detail in the following two subsections.

3.1. Feature Extractor

The feature extractor computes useful representations of the LR patch in order to infer its HR version. Concretely, we propose a recursive structure based on Residual Blocks (RBs) assembled into Dense Groups (DGs), see Figure 1. We describe each of these below.

Residual Blocks. We use a modified version of WDSR [33] with wide-low rank convolutions instead of using standard Residual Blocks [33]. These convolutions widen the activation space before the non-linearity to let more information pass through it in order to lose less detail, while using the same amount of computation as standard 3 × 3 Residual Blocks. In order to make the network focus on more informative features, we exploit the inter-dependencies among feature channels using squeeze-and-excitation (SE) operations [12] after these convolutions, see Figure 1.

Inspired by [26, 27], the model learns a scalar multiplier $\lambda$ to balance the amount of information that should be carried by the identity and activation operations within the residual units of the network. Let $x_i$ and $x_o$ be the input and output vectors of the $k$-th RB, and $WA$ the wide activation operation [33]. Then, the RB proceeds as follows:

$$x_o = \lambda_o SE(WA(x_i)) + \lambda_i x_i$$

Dense Groups. RBs are then composed into the so-called dense groups. The input of a DG is concatenated with the output of the first RB, and passed to the following RB after a $1 \times 1$ convolution. This recursion is repeated for all RBs within the DG. In this way, we gather all local information progressively by $1 \times 1$ convolution layers. The use of additional $1 \times 1$ convolution layers results in higher representation power.

To increase the network capacity, a similar recursion is applied to the Dense Groups within the network, but this time incorporating skip connections between the DGs. We repeat this procedure while integrating the recursive concatenations through the DGs into a single output. The output of each DG is concatenated to the input of the next one. In order to facilitate access to local information, the output of the network consists of the concatenation of the outputs of all the DGs. Therefore, the model incorporates features from multiple layers. This strategy makes information propagation efficient due to multi-level representation and many shortcut connections. Inspired by MemNet [29], we then introduce a $1 \times 1$ convolutional layer to adaptively control the output information, as directly using these concatenated features would greatly increase computational complexity. The output of these hierarchical features can be formulated
as

$$f_D = \text{conv}_{1 \times 1}(f_0, ..., f_{D-1})$$

(5)

where $[f_0, ..., f_{D-1}]$ refers to the concatenation of feature maps produced by DGs.

To make sure that no information is lost before the reconstruction step, we incorporate a long-range skip connection to grant access to the original information, and encourage back-propagation of gradients from the output of the feature extractor to the first $3 \times 3$ convolution layer. We also include a global average pooling followed by a $1 \times 1$ convolution, to fully capture channel-wise dependencies from the aggregated information. The final output before the reconstruction step is then,

$$h = \lambda_0 f_D + \lambda_1 \sigma(\text{conv}_{1 \times 1}(\text{GAP}(\text{conv}_{3 \times 3}(I^{LR}))))$$

(6)

where $\sigma$ denotes the ReLU activation, GAP denotes global average pooling, and $\lambda_0$ and $\lambda_1$ are learned parameters.

### 3.2. Overscaling Module

In this work we introduce a new inductive bias in SISR architectures so as to generate images that are more accurate and present fewer artifacts. We hypothesize that, since overscaling produces multiple values for the same pixel, these values act as an ensemble of predictions thus reducing noise when combined to produce the final image.

Let us consider $N$ the maximum scale factor addressed by the network. We first generate an intermediate representation of the final image consisting of overscaled maps $H^{OHR}$, with an overscale factor $(N+1)$ times larger. Thus, given the features $h$ extracted from $I^{LR}$, we use a $3 \times 3$ convolutional layer followed by the strided sub-pixel convolution proposed in [2] to upscale the features $h$ to $H^{OHR}$:

$$H^{OHR} = \text{pixelshuffle}(\text{conv}_{3 \times 3}(h))$$

(7)

To obtain the final output of the overscaling module, we further include a second long-range skip connection from the original $I^{LR}$ image. The final HR image is obtained by adjusting the overscaled maps and incorporating them into the na"ive upscaling of the original LR image:

$$\hat{I}^{HR} = \text{bicubic}_i(\text{conv}_{3 \times 3}(H^{OHR})) + \text{bicubic}_i(\text{bicubic}_\downarrow(I^{LR}))$$

(8)

Hence, we could think of the whole network as learning how to refine or correct a na"ive bicubic upscaling of the low-resolution input, in order to bring it closer to the actual high-resolution counterpart. Since the final $\hat{I}^{HR}$ images are obtained with an efficient non-parametric interpolation, we are able to produce multiple scales with negligible computational cost, and only using differentiable operations.

### 3.3. Multi-Scale Loss

We propose the minimization of a multi-scale loss to optimize the network. We choose a finite set of scale factors $S = \{s_1, ..., s_n\}$, all within the interval of scales targeted by the network. Once the network has reconstructed the HR image, images at the target scales are obtained through a bank of bicubic interpolators, $\hat{I}^{HR}_s = \text{bicubic}_i(I^{HR}, s)$. Then, we minimize the following loss function:

$$\mathcal{L} = \sum_{s \in S} |\hat{I}^{HR}_s - \text{bicubic}_i(I^{HR}, s)|$$

(9)

Training with this multi-scale loss at different target scales simultaneously provides additional supervision to the model, compared to a single-scale training. As a result, the model is enforced to learn how to generate highly representative overscaled maps, from which HR images at arbitrary scales can be recovered accurately, hence enforcing the generalization capability of the network across scales.

### 4. Experimental Results

In this section, we evaluate the performance of our model on a series of benchmark datasets. In addition, we provide...
In this section, we investigate the contribution of each of the different components of the proposed method.

### Effect of the OSM across scales

Here we analyze the benefits of introducing our OSM module in SISR. Additionally, we also explore the influence of the different interpolation methods on the reconstruction. We run the following experiments: (i) directly using pixelsHuffle to generate the images without map overscaling, (ii) downscaling an output generated using pixelsHuffle and overscaled maps with bicubic interpolation, and (iii) doing the same with bicubic interpolation.

### 4.1. Settings

**Datasets and metrics.** In this work, we use the DIV2K dataset for training. The DIV2K dataset is a recently proposed high-quality image dataset, containing 800 images for training, 100 for validation, and 100 for testing. We train all our models with the 800-image training set. Several standard benchmark datasets are used for testing, namely Set5 [4], Set14 [34], B100 [3], and Urban100 [11].

SR results are evaluated with two commonly used metrics: PSNR (peak-to-peak signal-to-noise ratio) and SSIM (structural similarity index), on the Y channel of the transformed YCbCr space.

**Degradation models.** In order to comprehensively illustrate the efficacy of the proposed method, three degradation models are used to simulate LR images, following [35, 36, 38]. The first one, denoted by **BI**, consists of generating LR images by bicubic-downsampling ground truth HR images with scaling factors $2 \times 2$, $3 \times 3$, $4 \times 4$. The second one, denoted by **DB**, first performs bicubic downsampling on HR images with scaling factor $3 \times 3$, and then blurs the images with a Gaussian kernel of size $7 \times 7$ and standard deviation $1.6$. Finally, we further produce LR images in a third challenging way, denoted by **DN**, by carrying out bicubic downsampling followed by additive Gaussian noise, with noise level of 30.

**Implementation details.** We used $64 \times 64$ RGB input patches from the LR images for training. We sampled LR patches randomly and augmented them with random horizontal flips and $90^\circ$ rotation. The number of DGs and RBs was set to 3 in all experiments. We trained our models with the ADAM optimizer [15]. The mini-batch size was set to 64, and the learning rate to the maximum convergent value $10^{-3}$, applying weight normalization in all convolutional layers [33]. The learning rate was decreased by half every $2 \times 10^5$ back-propagation iterations. We implemented our networks using the PyTorch framework [23] and trained them on NVIDIA 1080 Ti GPUs.

### 4.2. Ablation Studies

In this section, we investigate the contribution of each of the different components of the proposed method.

| Experiment          | Scale |
|---------------------|-------|
|                      | $\times 1.1$ | $\times 1.2$ | $\times 1.3$ | $\times 1.4$ | $\times 1.5$ | $\times 1.6$ | $\times 1.7$ | $\times 1.8$ | $\times 1.9$ | $\times 2.0$ |
| Pixelshuffle         | 31.60  | 31.22  | 30.75  | 30.50  | 30.27  | 29.95  | 29.73  | 29.42  | 29.17  | 29.09   |
| OSM-bilinear         | 31.71  | 31.29  | 30.84  | 30.55  | 30.37  | 30.02  | 29.77  | 29.52  | 29.30  | 29.26   |
| OSM-bicubic          | 31.75  | 31.34  | 30.86  | 30.65  | 30.42  | 30.11  | 29.83  | 29.64  | 29.36  | 29.30   |
| Meta-RDN             | 31.82  | 31.41  | 31.06  | 30.62  | 30.45  | 30.13  | 29.82  | 29.67  | 29.40  | 29.30   |
| OSM-RDN              | 31.75  | 31.46  | 31.10  | 30.60  | 30.48  | 30.15  | 29.79  | 29.71  | 29.35  | 29.38   |
|                      | $\times 3.1$ | $\times 3.2$ | $\times 3.3$ | $\times 3.4$ | $\times 3.5$ | $\times 3.6$ | $\times 3.7$ | $\times 3.8$ | $\times 3.9$ | $\times 4.0$ |
| Pixelshuffle         | 28.78  | 28.70  | 28.50  | 28.30  | 28.14  | 28.10  | 28.38  | 28.74  | 27.60  | 27.58   |
| OSM-bilinear         | 28.81  | 28.77  | 28.62  | 28.49  | 28.23  | 28.22  | 28.90  | 27.82  | 27.79  | 27.75   |
| OSM-bicubic          | 28.90  | 28.81  | 28.66  | 28.51  | 28.26  | 28.25  | 28.96  | 27.84  | 27.83  | 27.79   |
| Meta-RDN             | 28.87  | 28.79  | 28.68  | 28.54  | 28.32  | 28.27  | 28.04  | 27.92  | 27.82  | 27.75   |
| OSM-RDN              | 28.96  | 28.70  | 28.80  | 28.64  | 28.41  | 28.23  | 28.00  | 27.97  | 27.89  | 27.83   |

Table 1: PSNR results of different OSM upscaling methods trained for arbitrary scales. The test dataset is B100. The number of parameters for OSM models ranges between 0.9 and 1.0M, whereas Meta-RDN has 22M ($25 \times$ more). Best results are highlighted, second best underlined.
We replaced it with our overscaling module and trained it on pixelshuffle, a module based on a typical upsampling module, that is, CARN [1] as a reference. CARN is a lightweight method for SISR algorithms. To this end, we used the state-of-the-art network DIV2K for all scale factors (CARN with OSM in Table 2). We also trained OverNet by replacing its OSM with a typical upsampling module (OverNet without OSM in Table 2).

We observe that CARN with OSM has significantly higher PSNR than CARN at all scale factors. On the other hand, OverNet without OSM outperforms both CARN and CARN with OSM. This shows that (i) OSM is robust and orthogonal to the feature extractor choice, and (ii) both this module and the proposed feature extractor independently increase the PSNR when compared to CARN. Moreover, it can also be seen that combining the proposed feature extractor and OSM together (OverNet) increases performance.

Generalization across scales. By construction, the overscaling factor in our architecture is always \( N+1 \) when targeting a maximum scale of \( N \), c.f. Section 3.2. The following experiments investigate the generalization capability of models that target a maximum scale \( N \) across lower scales \( M \) \( \leq N \). To this end, we trained models for \( N \in \{2, 3, 4\} \) and evaluated them across scales.

Table 3 illustrates the experimental results. It can be observed that models trained to target larger scales yield better PSNR scores for all scale factors. This demonstrates the generalization capabilities of the proposed architecture across scales, as it is not necessary to train a dedicated model for each scale. Instead training a larger scale seems to be always beneficial for lower scales. Moreover, the cost to pay in terms of additional parameters is low.

Effect of multi-scale loss. In this section we demonstrate the influence of the multi-scale loss. The advantage of this learning strategy is that it can process multiple scales using a single trained model, while most of the state-of-the-art algorithms require to train separate models for each supported scale. This property targets real-world applications, where the output size is usually fixed but the input LR scale can vary. Moreover, the multi-scale loss acts as a regularizer, enforcing the generalization of the network across scales and improving the performance. As a result, the model is able to maintain accurate reconstruction results across scales representational capacity of OverNet.

Table 4 shows the experimental results. It can be observed that the model with the multi-scale loss achieves better performance with a large margin than model without multi-scale loss.
Table 5: Average PSNR/SSIM values for models with the same order of magnitude of parameters (MSRN included as a high-capacity reference model). Performance is shown for scale factors ×2, ×3 and ×4 with BI degradation. The number of parameters and multi-adds of each method are indicated under their name. The best performance is shown highlighted and the second best underlined.

| Dataset | Scale | VDSR [13] | DRCN [14] | LapSRN [17] | DRNN [28] | MemNet [29] | SRFBN [20] | OISR [9] | CARN [11] | OverNet | MSRN [19] |
|---------|-------|-----------|-----------|-------------|-----------|-------------|------------|---------|--------|--------|----------|
| Urban100 | 2     | 28.13 | 29.58 | 30.02 | 31.35 | 33.06 | 33.62 | 31.97 | 32.29 | 33.74 | 33.40 |
|         | 3     | 28.13 | 29.58 | 30.02 | 31.35 | 33.06 | 33.62 | 31.97 | 32.29 | 33.74 | 33.40 |
|         | 4     | 28.13 | 29.58 | 30.02 | 31.35 | 33.06 | 33.62 | 31.97 | 32.29 | 33.74 | 33.40 |

Table 6: Average PSNR/SSIM for models with the same order of magnitude of parameters (RDN included as a high-capacity reference model). Scores shown for scale factor ×3 using BD and DN degradation models. Best performance is highlighted, second best underlined.

| DB Degrad. | Bicubic | SPMSPR [24] | SRCNN [5] | FSRCNN [6] | VDSR [13] | IRCNN [8] | IRCNN+C [35] | SRFBN [20] | OverNet | RDN [38] |
|------------|---------|-------------|-----------|-------------|-----------|------------|--------------|------------|--------|--------|
| Set5       | BD      | 28.34 | 29.70 | 31.05 | 32.40 | 33.80 | 35.12 | 36.53 | 38.07 | 39.70 | 40.90 |
|            | DN      | 28.34 | 29.70 | 31.05 | 32.40 | 33.80 | 35.12 | 36.53 | 38.07 | 39.70 | 40.90 |

4.3. Comparison with State-of-the-Art Methods

4.3.1 Results with BI degradation models.

To test the effectiveness of our model, we compare OverNet with nine lightweight state-of-the-art SISR methods including VDSR [13], DRCN [14], LapSRN [17], DRNN [28], MemNet [29], SRFBN [20], OISR [9], and CARN [11]. We included MSRN [19] high-capacity model for reference. For fair comparison, we train our model individually for each scale factor, including ×2, ×3 and ×4. We test our model on different benchmarks with PSNR and SSIM.

Table 5 shows quantitative evaluation results, including the number of parameters and the number of multiplications and additions (multi-adds), for a more informative comparison (under the method name). Multi-adds were calculated with 1280×720 SR images at all scales. Note that, in this table we only compare models that have roughly similar number of parameters as our. Results show that our architecture outperforms all the networks with less than 2M parameters by a large margin. In the case of networks with a large number of parameters, such as MSRN (6M parameters), OverNet achieves better or competitive results, while using 7× less parameters than MSRN.

In addition, we present qualitative results in Figure 3.

1 Additional analysis and qualitative results can be found as supplementary material.
unable to remove blurring artifacts. In contrast, OverNet could alleviate distortions and generate more accurate details in the SR images. Regarding DN degradation, we observe that it is difficult to recover the details with the other methods. However, our method can deliver good results by removing more noise and enhancing detail.

4.3.3 Memory complexity analysis.

In Figure 2, we compare OverNet against various benchmark algorithms in terms of network parameters and reconstruction PSNR, using the B100 dataset and scale factor $\times 4$. OverNet achieves the best results among the networks with less than 2M parameters. This demonstrates that our method can correctly balance the number of parameters and the reconstruction performance.

4.3.4 Running time comparison.

We compare the running time of the proposed OverNet on Urban100 with five other state-of-the-art networks, namely MemNet [29], EDSR [21], SRFBN [20], D-DBPN [7], and Meta-RDN [10], using a scale factor $\times 4$. The running time of each network is evaluated using its official code, on the same machine with a NVIDIA 1080 Ti GPU. OverNet has the fastest evaluation time compared to the other networks (see Table 7), reflecting the efficiency of the proposed method.

5. Conclusion

We have introduced OverNet, a novel efficient architecture that facilitates image super-resolution at arbitrary scale factors using a single model. OverNet outperforms other state-of-the-art algorithms while keeping a reduced number of parameters and low computational requirements. The efficacy of the proposed algorithm results mainly from the following contributions: (i) a lightweight feature extractor, Table 7: Average running time comparison on Urban100 with scale factor 4.

| Model    | Parameters | Running Time (ms) | PSNR  |
|----------|------------|-------------------|-------|
| MemNet  | 0.6M       | 0.481             | 25.54 |
| EDSR    | 43M        | 1.218             | 26.64 |
| SRFBN   | 0.4M       | 0.006             | 25.71 |
| D-DBPN  | 10M        | 0.015             | 26.38 |
| RDN     | 22M        | 1.268             | 26.61 |
| Meta-RDN | 22M     | 1.350             | 26.65 |
| Ours    | 0.9M       | **0.004**         | **26.30** |

in which the proposed Residual Blocks and Dense Groups enhance the flow of information to preserve details; (ii) an Overscaling Module that helps to generate accurate SR images at different scaling factors, and (iii) a multi-scale loss function that enhances the training compared to dedicated single-scale models. Thanks to the OSM, we can train a single model for super-resolution at arbitrary scale factors. More importantly, we proved that our overscaling head can be flexibly applied to other SR models by simply replacing their upsampling module, thus improving their original performance. The provided empirical evidence suggests that the proposed overscaling method may help with other low-level image restoration tasks, such as denoising and dehazing.

Figure 3: Visual results of BI degredation model with scale factor $\times 4$.

References

[1] N. Ahn, B. Kang, and K.-A. Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In Proceedings of the European Conference on Computer Vision (ECCV), pages 252–268, 2018.

[2] A. Aitken, C. Ledig, L. Theis, J. Caballero, Z. Wang, and W. Shi. Checkerboard artifact free sub-pixel convolution: A
Figure 4: Visual results of BD and DN degradation models with scale factor $\times 3$. The first set of images shows the results obtained from BD degradation model and the second set indicates the results from DN degradation model.

Note on sub-pixel convolution, resize convolution and convolution resize. arXiv preprint arXiv:1707.02937, 2017.

[3] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. IEEE transactions on pattern analysis and machine intelligence, 33(5): 898–916, 2010.

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

[5] C. Dong, C. C. Loy, K. He, and X. Tang. Learning a deep convolutional network for image super-resolution. In European conference on computer vision, pages 184–199. Springer, 2014.

[6] C. Dong, C. C. Loy, and X. Tang. Accelerating the super-resolution convolutional neural network. In European conference on computer vision, pages 391–407. Springer, 2016.

[7] M. Haris, G. Shakhnarovich, and N. Ukita. Deep back-projection networks for super-resolution. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1664–1673, 2018.

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[9] X. He, Z. Mo, P. Wang, Y. Liu, M. Yang, and J. Cheng. Ode-inspired network design for single image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1732–1741, 2019.

[10] X. Hu, H. Mu, X. Zhang, Z. Wang, T. Tan, and J. Sun. Meta-sr: A magnification-arbitrary network for super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 1575–1584, 2019.

[11] J.-B. Huang, A. Singh, and N. Ahuja. Single image super-resolution from transformed self-exemplars. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 5197–5206, 2015.

[12] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer. SqueezeNet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. arXiv preprint arXiv:1602.07360, 2016.

[13] J. Kim, J. Kwon Lee, and K. Mu Lee. Accurate image super-resolution using very deep convolutional networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1646–1654, 2016.

[14] J. Kim, J. Kwon Lee, and K. Mu Lee. Deeply-recursive convolutional network for image super-resolution. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1637–1645, 2016.

[15] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. ICLR, 2014.

[16] W. Lai, J. Huang, N. Ahuja, and M. Yang. Fast and accurate image super-resolution with deep laplacian pyramid networks. corr abs/1710.01992. arXiv preprint arXiv:1710.01992, 2017.

[17] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 624–632, 2017.

[18] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 4681–4690, 2017.

[19] J. Li, F. Fang, K. Mei, and G. Zhang. Multi-scale residual network for image super-resolution. In Proceedings of the European Conference on Computer Vision (ECCV), pages 517–532, 2018.

[20] Z. Li, J. Yang, Z. Liu, X. Yang, G. Jeon, and W. Wu. Feedback network for image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3867–3876, 2019.

[21] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee. Enhanced deep residual networks for single image super-resolution. In Proceedings of the IEEE conference on computer vision and pattern recognition workshops, pages 136–144, 2017.
[22] J. Liu, W. Zhang, Y. Tang, J. Tang, and G. Wu. Residual feature aggregation network for image super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2359–2368, 2020.

[23] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017.

[24] T. Peleg and M. Elad. A statistical prediction model based on sparse representations for single image super-resolution. IEEE transactions on image processing, 23(6):2569–2582, 2014.

[25] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1874–1883, 2016.

[26] R. K. Srivastava, K. Greff, and J. Schmidhuber. Highway networks. arXiv preprint arXiv:1505.00387, 2015.

[27] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[28] Y. Tai, J. Yang, and X. Liu. Image super-resolution via deep recursive residual network. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 3147–3155, 2017.

[29] Y. Tai, J. Yang, X. Liu, and C. Xu. Memnet: A persistent memory network for image restoration. In Proceedings of the IEEE international conference on computer vision, pages 4539–4547, 2017.

[30] T. Tong, G. Li, X. Liu, and Q. Gao. Image super-resolution using dense skip connections. In Proceedings of the IEEE International Conference on Computer Vision, pages 4799–4807, 2017.

[31] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In Proceedings of the European Conference on Computer Vision (ECCV), pages 0–0, 2018.

[32] Z. Wang, J. Chen, and S. C. Hoi. Deep learning for image super-resolution: A survey. arXiv preprint arXiv:1902.06068, 2019.

[33] J. Yu, Y. Fan, J. Yang, N. Xu, Z. Wang, X. Wang, and T. Huang. Wide activation for efficient and accurate image super-resolution. arXiv preprint arXiv:1808.08718, 2018.

[34] R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse-representations. In International conference on curves and surfaces, pages 711–730. Springer, 2010.

[35] K. Zhang, W. Zuo, S. Gu, and L. Zhang. Learning deep cnn denoiser prior for image restoration. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3929–3938, 2017.

[36] K. Zhang, W. Zuo, and L. Zhang. Learning a single convolutional super-resolution network for multiple degradations. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3262–3271, 2018.

[37] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu. Image super-resolution using very deep residual channel attention networks. In Proceedings of the European Conference on Computer Vision (ECCV), pages 286–301, 2018.

[38] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu. Residual dense network for image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2472–2481, 2018.