Single Image Super-Resolution via a Dual Interactive Neural Network

Snir Koska
Electrical Engineering Department
Tel Aviv University
Tel Aviv, Israel
snirkoska@mail.tau.ac.il

Dor Arviv
Electrical Engineering Department
Tel Aviv University
Tel Aviv, Israel
dorarviv@mail.tau.ac.il

Abstract

In this work, we deal with an innovative architecture designed to enhance images quality, specifically addressing the challenge of single image super-resolution across arbitrary scales. The approach employs a novel concept: conceptualizing an image as a "decoding function", which effectively correlates image coordinates with their corresponding attributes, resulting in a refined pixel representation. Given the continuous nature of this coordinate representation, this methodology remains adaptable to diverse image resolutions, enabling referencing of any position within images of varying scales. To reconstruct an image with a desired resolution, the decoding function is applied across a grid of coordinates, where each grid point is aligned with a pixel's center in the output image. A distinction from conventional methodologies lies in the architecture of the dual interactive implicit neural network (DIINN), which effectively disentangles content-related features from positional attributes. Consequently, achieving a fully depiction of the image, providing a resolution enhancement solution that accommodates arbitrary real-valued scale factors. We discuss the influence and implications of several different architectural modifications on SISR results, while preserving the core structure of encoder-decoder DIINN structure. Our goal is to test how these changes affect the performance of the network on different common datasets used for SISR.

1 Introduction

Single Image Super-Resolution (SISR) remains one of the most intriguing and challenging pursuits within the domain of computer vision. At its core, SISR is an endeavor to reconstruct a high-definition image from a singular low-resolution counterpart. This pursuit is not merely a technical challenge but holds profound implications in real-world scenarios, with its implications ranging from the enhancement of everyday photos to more mission-critical applications such as medical diagnostics, satellite imagery, and surveillance. The success in SISR can be a game-changer. Moreover, a persistent challenge in this realm has been the scale rigidity of many deep learning models. While they may excel in a specific scale factor, their performance wanes when introduced to diverse scales. Historically, the SISR problem has been approached from various angles, including classic interpolation techniques. However, these traditional methods often yield images that, while higher in resolution, lack in detail and often introduce artifacts. The advent of deep learning, and in particular convolutional neural networks (CNNs), has revolutionized the SISR landscape. Neural networks, with their capacity to learn intricate patterns and represent complex functions, have shown immense promise in producing super-resolved images that are not only higher in resolution but also closer in detail to the original high-definition image. It's in this evolving landscape that the work of Nguyen et al. emerged, introducing a paradigm shift with their Dual Interactive Implicit Neural Network. Their approach, utilizing implicit neural representations, stood out for its ability to produce outputs at arbitrary resolutions, addressing one of the long-standing challenges in SISR – adaptability to various scales. Our work builds upon Nguyen et al.'s work and refines its foundation. Recognizing the potential of Nguyen et al.'s encoder-decoder architecture, we introduced modifications to both components which will be elaborated in this article. These changes enhance the robustness of the model, allowing it to concentrate on the most informative parts of the image in order to regenerate intricate image details, which is crucial for SISR tasks. We also demonstrate the model's robustness as well as its capability to perform well across arbitrary real valued scale factors with little training.

2 Related Work

We base our work on the original article in of Nguyen et al. in [1]. This article proposes a novel approach to Single Image Super-Resolution (SISR) using a Dual Interactive Implicit Neural Network (DIINN) architecture. This unique architecture can handle arbitrary real values scale factor for the task of SISR (with respect to older neural networks architectures that perform well only on the scale factor on which it were trained on). Older models include techniques such as:

- Upscaling & refinement, which first uses interpolation of the LR image and then apply a CNN to enhance the interpolated image. (VDSR [2], DRCN [3]).
- Learning features & upscaling, which first input the LR image through a CNN, to obtain a deep feature map and then apply upscaling (Meta-SR [4]).
- Imlicit neural representations, which are a way to parameterize signals continuously. Learning implicit neural representations of images is useful for SISR since it enables to sample the pixel signals at any location in the spatial domain [5].

The DIINN architecture presented in the article, decouples content and positional features of the input low resolution (LR) image. The DIINN handles the image content features in a modulation branch and the positional features in a synthesis branch, while allowing for interactions between the two. DIINN uses an implicit neural representation (also known as coordinate-based representation) which is a pixel-level representation of the image that allows for

locally continuous super-resolution synthesis with respect to the nearest LR pixel. The proposed method achieves state-of-the-art results on publicly available benchmark datasets. The proposed network consists of the following two parts: an encoder and a dual interactive implicit decoder. The encoder learns the content of the LR image and produces a deep feature map. The implicit decoder predicts the SR image at any query location within the image space, conditioned on the associated features provided by the encoder. The encoder is the RDN architecture [6] without the upsampling module. The encoder produces a deep feature map of the same spatial size of the input image with 64 channels. Interpolation is performed on the feature maps, thus effectively increasing the spatial size of them to be the same as the target resolution. The feature maps are then fed into the decoder. The decoder is based on a dual MLP architecture which utilizes periodic activations (as presented in [7]). It consists of two 4-layer MLPs (each with 256 neurons) implemented as convolutional layers with 256 kernels of size 1. The two MLP branches interact with each other via concatenation of each previous layer's output of the synthesis branch in the following input to the modulation branch.

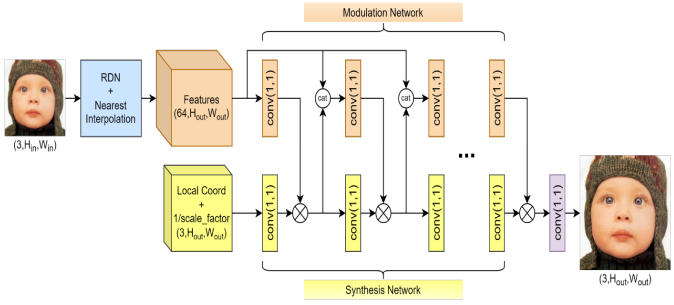

Figure 1: Original DIINN Architecture

3 Data

The datasets we used are the same used in the original article. Those are publicly available datasets used for training and benchmark of image super resolution tasks.
The DIV2K dataset [8], consists of 1000 HR images each of which has a height or width equal to 2040.
Our models are trained using a split of 800 HR images. 100 different images from the set are used on the original DIINN model for validation, we have only used 10 images due to memory limitations.
Our models are tested on the following four standard benchmark datasets:

- B100[9]
- set5
- set14
- Urban100[10]

4 Methods

In our endeavor to enhance the super-resolution process, we have revisited and modified both the encoder and the decoder architectures from Nguyen et al.'s work. In this section, we will present the methods used to modify the original DIINN network as well as methods used to evaluate the results of the models tested.

4.1 Modified encoder architecture

The original encoder is built on the Residual Dense Network (RDN) paradigm, which leans heavily on Residual Dense Blocks (RDBs). These blocks, layered with multiple convolutional layers, perform the robust feature extraction. Following this, the Local Feature Fusion (LFF) and Global Feature Fusion (GFF) processes added subsequent layers of refinement. In this section we present our new encoder architecture based on this core idea with some and modifications.
The choice of the VGGSuperResEncoder (new encoder function) architecture as an encoder for Single-Image Super-Resolution (SISR) tasks is grounded in a combination of factors.
VGG architecture as encoder foundation and initial feature extraction. The new encoder architecture employs the VGG architecture, as its foundation. VGG network, known for their simplicity and effectiveness in capturing image features, considered great, robust feature extractor. By inheriting this structure as an initial, shallow feature extraction mechanism, the encoder benefits from its ability to extract hierarchical features effectively: both low and high level details semantics, which is crucial for super resolution tasks.
Employing **SE Blocks** in the deeper feature extraction stages. A significant enhancement is in introducing of Squeeze and Excitation (SE) block within the RDBs for channel-wise feature recalibration, and the Global Feature Fusion process. SE blocks recalibrate channel-wise feature responses by explicitly modeling interdependencies between channel, providing a more robust feature extraction mechanism and enabling the encoder to amplify informative and important feature channels and suppress the less useful ones, leading to a more expressive feature representation, and allowing the model to learn richer feature representations. Feature quality is further enhanced in the global feature fusion step when combining information from different RDBs.

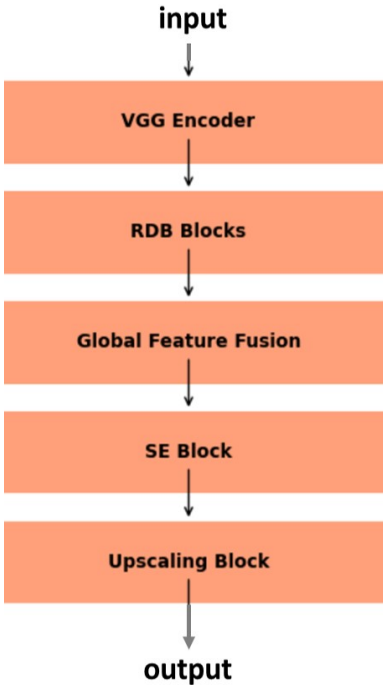

Figure 2: Encoder Architecture Diagram

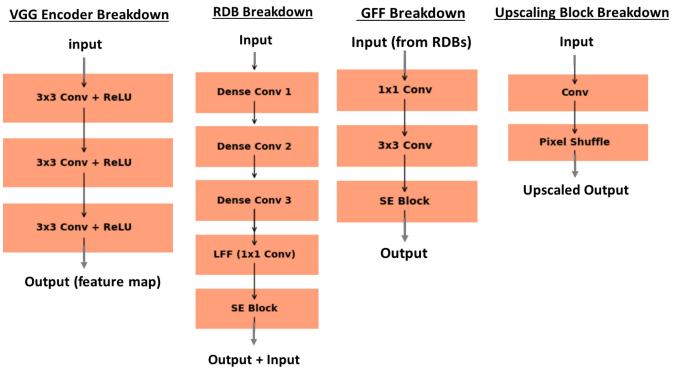

Figure 3: Encoder Blocks Breakdown

4.2 Modified decoder architecture

In a decoder architecture, especially in generative models, the interactions between the modulation and synthesis branches are critical for generating meaningful and high-quality output. We explore a modification to the decoder using **adaptive modulation**, which can lead to more context-aware synthesis. The integration of adaptive modulation into the decoder is done by introducing a new branch which allows the model to dynamically scale the activations of the synthesis branch based on the activations of the modulation branch. The branch consists of convolutional layers that learn modulation factors to scale the activations of the synthesis branch based on the modulation branch inputs. These modulation factors are then applied with element-wise multiplication to the activations of the synthesis branch. This means that the output of the Q branch is multiplied by the modulation factor before passing it through the final layers of the decoder. The adaptive modulation allows the model to dynamically emphasize or de-emphasize certain features in the synthesis branch based on the information contained in the modulation branch. This can improve the model's ability to capture and represent complex patterns in the data. Such dynamic modulation can be handy for super-resolution tasks, where the relevance of certain feature maps can vary across different image regions, enabling the model to weigh feature relevance in diverse image contexts.

Additionally, we test in our architecture a regularization method using **dropout**. After including adaptive modulation and therefore increasing the model size, we want to maintain the model's robustness and prevent overfitting. We test several probabilities for the dropout: 0, 0.1, 0.3 & 0.5.

Usage of **dilated convolution**: A task we want to explore which was suggested by the authors of the original article as future work is to increase the receptive field of the model. On such way is to use dilated convolution which avoids increasing the number of parameters in the model. Dilated kernel enables the decoder to grasp broader contextual information of the image. By diversifying dilation factors, the architecture captures multi-scale contexts with precision. Dilation factors used: 1, 2, 4 & 8.

A shift to **SiLU activation**: the transition to SiLU (also known as Swish) from ReLU and the periodic Sin activation functions. Unlike ReLU, which simply thresholds negative values to zero or Sin that introduces periodic patterns, SiLU offers a smooth, differentiable curve that self-regulates values both above and below zero, based on the input. This non-monotonic, smooth, differentiable behavior allows it to adaptively adjust activations based on the input value. Furthermore, SiLU can mitigate the vanishing gradient problem, a challenge often encountered with ReLU, especially in deeper networks.

4.3 Evaluation methods

Quantitative evaluation for the super-resolved images with respect to the ground truth high resolution images is done by two commonly used metrics: PSNR & SSIM.

4.3.1 Peak Signal to Noise Ratio (PSNR)

- PSNR is a measure of the quality of a reconstructed or super-resolved image compared to the original or ground truth image.
- It quantifies the level of noise or distortion present in the super-resolved image by calculating the ratio of the peak signal (maximum possible pixel value) to the mean squared error (MSE) between the super-resolved and ground truth images.
- Higher PSNR values indicate better quality, as they imply lower distortion and closer resemblance to the original image.

4.3.2 Structural Similarity Index Measure (SSIM)

- SSIM is a metric that assesses the structural similarity between two images, one being the super-resolved image and the other the ground truth image.
- It considers three components of image quality: luminance (brightness), contrast, and structure, and computes a similarity index based on these components.
- SSIM values range from -1 to 1, where 1 indicates a perfect match and -1 indicates a completely dissimilar pair of images.
- Higher SSIM values indicate better similarity and, hence, higher quality super-resolved images
- SSIM is often favored over PSNR for image quality assessment because it takes into account more aspects of human visual perception and tends to better reflect perceived image quality.

5 Experiments

In this section we present the different experiments, results and evaluation for the models we have trained and tested. In section 5.1 we discuss a procedure of progressively upscaling the image instead of using a large scaling factor upfront, in Section 5.2 we give the training details, in section 5.3 we perform qualitative evaluation for the results of different models and in section 5.4 we compare the benchmark results quantitatively.

5.1 Progressive upscaling procedure

We first performed a procedure of progressively upscaling the LR image instead of super resolving it with a large scale factor upfront. The test was performed on the original DIINN model from the article. The idea behind this test is that perhaps instead of upscaling the image using a large scale factor, gradually upscaling it (by using a small scale factor each time and then feeding the result back to the model) would result in a better quality image (because of the smaller change in size, more pixels to interpolate from and less "enhancements" required by the network).

Figure 4: Results of super resolved image with upfront scale 4 (SRx4) and gradually scaled image by factors of 1.25, then 1.6 and then 2 (SRx1.25x1.6x2)

These results were evaluated using PSNR and SSIM as shown in the table below:

	Upfront (x4)	Gradual (x1.25x1.6x2)
PSNR	21.0311	17.8846
SSIM	0.6638	0.5848

Table 1: Evaluation of super resolved image with upfront scale 4 (SRx4) and gradually scaled image by factors of 1.25, then 1.6 and then 2 (SRx1.25x1.6x2).

To our surprise, the upfront scaling surpassed the gradual scaling on both measures and the quality of the image is better upfront (less artifacts on the windows of the upfront building image). We can infer that the model results rely more on the network itself (which provides good results on arbitrary scale factors) and less on the interpolation operation.

5.2 Training details

Our training procedure is similar to the original article with a few changes made due to memory and computation power limitation.For each scale $factors \in \{2, 4\}$ and HR image in the minibatch, we randomly cropped a 36×36 patch and down sample it using bicubic interpolation .We randomly applied horizontal, vertical, and\or diagonal flips, each with a probability of 0.5. The training was done where each epoch is a full pass through 800 HR images in the DIV2K training set times the number of trainsets repeat defined. The initial models were trained for approximately 5 epochs (changing depends on memory limitations) with 10 trainsets repeats (8000 images each epoch), for a total of about 40,000 training images. Later we switched to training with 5 trainsets repeats (4000 images each epoch) for 5 to 20 epochs (depending on the model trained), for a total of about 20,000 to 80,000 training images (again, depending on the model). Training was done using the Adam optimizer with the default hyper parameters provided by PyTorch. The learning rate was initialized at 10 to 4. We have used the L1 loss to train our network. Each model was evaluated at the end of each epoch using an evaluation batch size of 3000.

5.3 Qualitative evaluation of results

In this section we present the image results from each trained model. We have used an input LR image of size 288x288 pixels and super-resolved it to sizes: 400x400, 800x800 and 1000x1000 (real valued arbitrary scale factors). For simplicity, we present only the SR image sized 1000x1000 for each model. The models are ordered in the following manner:

- Model A: decoder with adaptive modulation (original encoder kept).
- Model B: decoder with dilated convolution (dilation of [1, 2, 4, 8]) (original encoder kept).
- Model C: decoder with adaptive modulation, dilated convolution, changes of activations to SiLU function (in both branches of the decoder) and dropout of 0.5 (original encoder kept).
- Model D: decoder with adaptive modulation, dilated convolution, changes of activations to SiLU function (in both branches of the decoder) and dropout of 0.3 (original encoder kept).
- Model E: decoder with adaptive modulation, dilated convolution, changes of activations to SiLU function (in both branches of the decoder) and dropout of 0.1 (original encoder kept).
- Model F: decoder with adaptive modulation, dilated convolution, changes of activations to SiLU function (in both branches of the decoder) and dropout of 0 (original encoder kept).
- Model G: new encoder VGG architecture with SE blocks (original decoder kept).
- Model H: new encoder VGG architecture with SE blocks & decoder with adaptive modulation, dilated convolution, changes of activations to SiLU function (in both branches of the decoder) and dropout of 0.

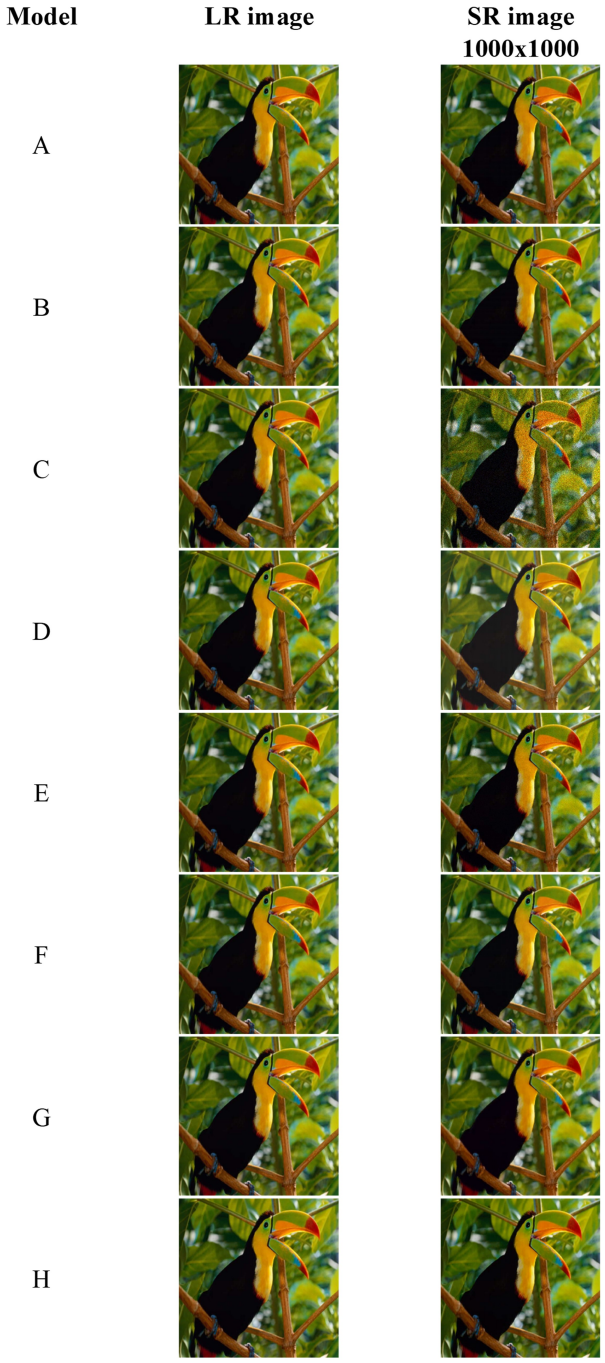

Figure 5: SR image results of the different models trained

Visually, we can see all the models did a relatively good job at super-resolving the LR image, while model C did the worst (because of the dotted artifacts on the image). The new decoder in model F performed best (across all other decoder variations) so it was chosen along with the new encoder in model G to create model H (new encoder and decoder architecture).
To demonstrate the capability of model H we have created the following example of super resolving a LR image of size 128x128 by factors of 1.5 & 2:

Figure 6: model H results, LR image on the left, SRx1.5 on the middle and SRx2 on the right

5.4 Benchmark and quantitative comparison of the results

In this section we present the benchmark results across all models trained and compare them. The benchmark used is similar to the one used in the original article, across 4 test datasets: B100, set5, set14 & URBAN100. For each dataset, the PSNR and SSIM were computed between the ground truth HR image vs the SR image. The SR scale factors tested are: 3.14, 4 & 8.

Model	SSIM 3.14	SSIM 4	SSIM 8	PSNR 3.14	PSNR 4	PSNR 8
A	0.5707	0.5028	0.3634	23.8673	23.1125	21.3797
B	0.6188	0.5643	0.4096	23.0040	22.5127	20.5930
C	0.6733	0.6078	0.4721	23.2912	22.6600	21.1770
D	0.6819	0.6031	0.4577	23.5244	22.6884	20.9663
E	0.6594	0.5960	0.4554	23.0031	22.4609	21.0288
F	0.6691	0.6044	0.4597	23.4170	22.7795	21.1399
G	0.7093	0.6466	0.4826	24.6562	23.7387	21.7813
H	0.7369	0.6712	0.5240	26.0928	25.0048	22.6406

Table 2: benchmark results on B100 dataset

Model	SSIM 3.14	SSIM 4	SSIM 8	PSNR 3.14	PSNR 4	PSNR 8
A	0.6074	0.6027	0.4254	24.2179	24.4839	21.2899
B	0.6729	0.6486	0.4542	23.3229	23.0572	19.9573
C	0.7097	0.6662	0.5123	23.0843	22.5527	20.4245
D	0.6797	0.5737	0.4081	20.9223	19.4025	17.0171
E	0.7151	0.6816	0.5161	23.1951	22.7821	20.5274
F	0.7353	0.6988	0.5321	23.2674	22.7108	20.4132
G	0.7772	0.7537	0.5936	25.6495	24.9245	21.8716
H	0.8424	0.8063	0.6193	28.2304	27.2920	22.9941

Table 3: benchmark results on set5 dataset

Model	SSIM 3.14	SSIM 4	SSIM 8	PSNR 3.14	PSNR 4	PSNR 8
A	0.5402	0.5010	0.3478	22.9226	22.6166	20.3710
B	0.6015	0.5464	0.3955	20.9313	20.4953	18.7223
C	0.6517	0.5935	0.4477	21.2536	20.7414	19.2767
D	0.6630	0.5738	0.4232	21.6052	20.4762	18.7748
E	0.6574	0.6056	0.4462	21.4753	21.1035	19.5600
F	0.6823	0.6283	0.4644	22.1275	21.5959	19.8252
G	0.6844	0.6398	0.4656	22.7454	22.0948	20.1048
H	0.7486	0.6924	0.5305	25.5856	24.5622	21.5713

Table 4: benchmark results on set14 dataset

Model	SSIM 3.14	SSIM 4	SSIM 8	PSNR 3.14	PSNR 4	PSNR 8
A	0.5030	0.4901	0.3075	19.9368	20.3152	17.9866
B	0.5719	0.5047	0.3821	16.5178	16.2864	15.2531
C	0.5534	0.5025	0.3616	16.4609	16.3464	15.4946
D	0.6499	0.5736	0.4061	20.1018	19.6327	17.8133
E	0.5701	0.5279	0.3805	17.0230	17.2865	16.1353
F	0.6599	0.5955	0.4210	19.4871	19.1938	17.5303
G	0.6403	0.6188	0.4334	18.5722	18.2792	16.8758
H	0.7439	0.6839	0.5017	23.5492	22.5949	19.7323

Table 5: benchmark results on Urban100 dataset

In general, we can see that across all datasets and models, when the scale factor increases the quality of the results decreases (as can be seen in both the PSNR and SSIM). This can be expected as the size of the output increases significantly there is less pixels to interpolate from in the original image and the network needs to perform much better enchantments to output a quality image. Set5 and Set14 are relatively small in size so a large variance in the results can be seen, so most of what we conclude about the models stems from sets B100 and URBAN100. As part of the ablation study we can see model A performs relatively good in terms of PSNR but model B outperforms it in terms of SSIM. Models C, D & E (all with dropout) performed worse than others on most cases as well as producing lower quality images with artifacts. From this we can say that the modified decoder model does not need any further regularization (excluding the already used L1 regularization) and indeed model F (without dropout) achieved better PSNR results

across most sets and better/similar SSIM across all sets. The new encoder in model G performed relatively similar to model F on its own, we then combined the models to get model H which shows the best performance of all models, across all sets and evaluations metrics.

6 Conclusion

In this work, we have tested and demonstrated the capabilities of the DIINN model for SISR. We have applied several modifications to the model, and trained several different models in an ablation study. We explored the model architecture in order to increase the model's receptive field and perform better image synthesis. We have shown the model is robust and is able to perform well across arbitrary scale factors with the new architecture and activation functions as well (even when trained little compared to the original model). We have seen that the model's performance (even when number of parameters increased while using adaptive modulation) was worse when applying dropout regularization and led to visible artifacts that reduces the SR image quality. We concluded that for this specific task and training time, regularization isn't needed. The models were tested across known benchmark sets on a variety of scale factors. Most models performed well and resulted in a visually pleasing SR image. All the results were evaluated using common metrics for this task. For further work, one can take the final model (H) as described and run it through a "full" training procedure using many more epochs and evaluation cycles (similar to the original training procedure described in the article) with sufficient computing resources.

References

[1] Nguyen, Quan H. and Beksi, William J. Single Image Super-Resolution via a Dual Interactive Implicit Neural Network.In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pages 4936-4945, 2023.

[2] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1646– 1654, 2016.

[3] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Deeplyrecursive convolutional network for image super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1637–1645, 2016.

[4] Xuecai Hu, Haoyuan Mu, Xiangyu Zhang, Zilei Wang, Tieniu Tan, and Jian Sun. Meta-sr: A magnificationarbitrary network for super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 1575–1584, 2019.

[5] Yinbo Chen, Sifei Liu, and Xiaolong Wang. Learning continuous image representation with local implicit image function. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8628–8638, 2021.

[6] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018.

[7] Ishit Mehta, Michael Gharbi, Connelly Barnes, Eli Shechtman, Ravi Ramamoorthi, and Manmohan Chandraker. Modulated periodic activations for generalizable local functional representations. In Proceedings of the IEEE/CVF International Conference on Computer Vision, pages 14214–14223, 2021.

[8] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 126–135, 2017.

[9] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In Proceedings of the IEEE/CVF International Conference on Computer Vision, volume 2, pages 416–423, 2001.

[10] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5197–5206, 2015.

[11] original source code: https://github.com/robotic-vision-lab/Dual-Interactive-Implicit-Neural-Network/tree/main

Appendix

All code files written and edited are available in the project's shared folder. Please also refer to the readme file in the folder. The files written for the project can be found in src/models/components and are the following (using pytorch and tensorflow):

- updated_new_encoder.py
- updated_new_encoder_tensorflow.py
- diinn.py
- measures.py