Single Image Super Resolution in the Wild

Isai Daniel Chacón Silva
Universidad de los Andes
Bogotá, Colombia
id.chacon@uniandes.edu.co

Juan Sebastián Urrea López
Universidad de los Andes
Bogotá, Colombia
js.urrea@uniandes.edu.co

Abstract

This paper tackles the task of Single Image Super Resolution (SISR) using the ImagePairs dataset. SISR aims to recover a high resolution image from a single low resolution image. In this work we adapt a novel approach in this task: visual transformers, in particular SwinIR. Using this architecture, we obtained comparable state-of-the-art results for this dataset: $23,470$ *in the PSNR and* $0,784$ *for the SSIM metric by fine tuning with the DIV2K dataset for 30000 iterations. We also obtained really good results with less iterations (10000) by removing 2 attention heads with respect to the default implementation of SwinIR. With this method we got* $23,270$ *in the PSNR and* $0,764$ *for the SSIM metric. With this last method we use 15 times less training iterations than state-of-the-art methods, which implies less training time, lower computational cost and a faster and more efficient convergence of the model in the SISR context.*

1. Introduction

Super Resolution (SR) is the process of recovering a high resolution (HR) image from a single or multiple low resultion (LR) images of the same scene [17]. In particular, the Single Image Super Resolution (SISR) is the problem of recovering the HR image from only one LR observation. According to this definition, given a HR image, we can compute the LR so that

$$\mathbf{LR} = D(\mathbf{HR}, \theta_D) \tag{1}$$

where $D(\cdot)$ is a degradation process defined by the parameters θ_D. In a real scenario, those parameters are unknown. In this sense, SISR aims to recover a possible estimate of the HR image by reversing the degradation process shown in equation 1. This can be stated as

$$\widehat{\mathbf{HR}} = R(\mathbf{LR}, \theta_R) \tag{2}$$

where $R(\cdot)$ represents the resolution function with its corresponding parameters θ_R [9]. However, the HR space that is intended to be mapped through R is usually intractable [49, 47], which makes it a challenging task.

HR images are valuable in various areas. In applications such as traffic surveillance and security monitoring results impossible and expensive to equip large-scale HR hardware. In terms of medical imaging, LR limitations degrade the valuable anatomical information of the human body and its functionality, which makes a proper diagnosis difficult [51]. Also, better quality images can help in the astronomical area to study space in a more precise way that helps with the progress of science. There also exist many industrial applications for the verification of production standards automatically. At last, people are interested in obtaining HR video and images in their everyday lives, which makes regular images and videos information enhancement an area of great interest.

A logical approach to obtain HR images could be to upgrade the hardware used. However, the two main limitations to this approach are that new devices may capture new HR images, but they will not improve the resolution of existing LR images. Moreover, the demand in practical applications are constantly changing, which makes it inflexible and costly [9].

Probabilistic models and interpolation methods (e.g. bilineal and bicubic) have been developed in the last decades with considerable restrictions on its performance for images in the wild. For this reason, deep learning-based methods have been recently used, such as Convolutional Neural Networks (CNNs), Generative Adversarial Networks (GANs) and visual transformers [40].

In this paper, we implement the state-of-the-art architecture in the SR task, SwinIR [25] to the ImagePairs dataset and propose a new distribution in the number of multi-head self-attention layers. We also perform several ablation studies over different layers of the network, its loss function and its various hyperparameters.

2. Related work

SISR methods are mainly divided into three categories: interpolation-based, reconstruction-based and learning-

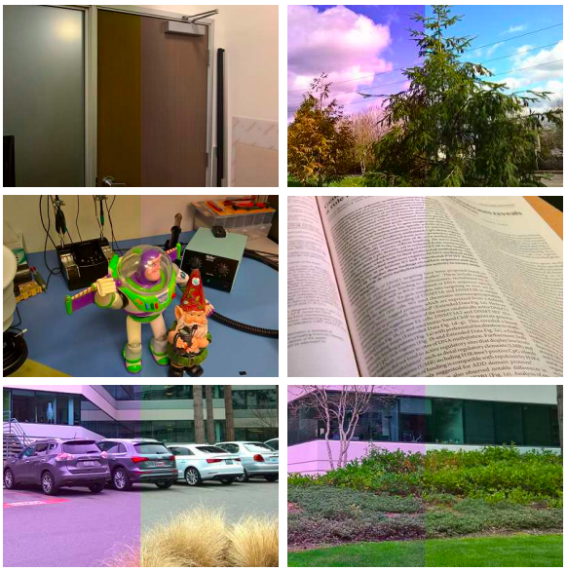


Figure 1. Sample images from ImagePairs dataset [17]. Each image divided by half horizontally to show LR (left) and HR (right).

based [49]. The learning-based methods are characterized by their fast computation and outstanding performance [49]. Among the various deep learning techniques used in these methods, neural networks (NN) stand out, often achieving state-of-the-art at several SR tasks [44, 49]. Different NN architectures are used for SR, such as those based on CNNs, those based on GANs and those based on Transformers.

2.1. Convolutional Neural Networks

Deep Learning was introduced for the task of SR with SRCNN, which learned an end-to-end mapping between an upsampled LR image and the HR image [12]. Following this, FSRCNN implemented a deconvolution operation to learn the mapping directly from the LR image [13]. This was then optimized by the introduction of ESPCN, an efficient subpixel convolution layer [34]. After that, VDSR introduced the first very deep model in SR, with a 20-layer VGG network that implemented residual learning and could be used at different scales [21]. More recent work includes multisupervised training [21, 22, 27], residual units [22, 24, 35], dropping batch normalization [27], dense connections [15, 37, 36, 52], ensembles of Neural Networks [28], progressive methodologies [48, 23, 10, 39, 38] and backprojection [14].

GuidanceNet [26] proposed a deep NN architecture to map images between the RAW and RGB domains. This NN consists mainly of global and local sub-networks. The first sub-network deals with color mapping and illumination, whereas the second focuses in recovering image details. In the ImagePairs [17] dataset, they did not use the LR/HR pair, but the LR and its RAW data to train an Image Signal Processing (ISP) network. With this framework, GuidanceNet is the current state of the art in the ImagePairs dataset, obtaining results of 29.22 (db) in PSNR and 0.96 in SSIM.

2.2. Generative Adversarial Networks

GANs have the ability to synthesize high-quality images [41]. A major breakthrough of this architecture in the field of SR is SRGAN [24]. This network uses an adversarial loss to push the solution towards the natural image manifold by comparing the generated image and the original HR image in the discriminator [24]. Following this work, ESRGAN [43] proposes several improvements such as the Residual-in-Residual Dense Block to achieve better visual quality. In the same way, to yield more realistic textures ESRGAN+ [32] proposes changes in the basic block, while [42] has proposed the use of a Spatial Feature Transform layer.

There are more architectures and training strategies that have been implemented with success for the task of SR. For example, ProGAN grows both the generator and discriminator progressively, layer by layer, to increase fine details as training progresses [18]. StyleGAN uses a style-based design that allows to differentiate between high-level and low-level attributes of the generated images [19]. BigGANs are able to generate diverse samples from complex datasets such as ImageNet [4]. Progressive face SR proposes a novel facial attention loss that achieves great performance on SR for face images [20]. The method by Ren et al. proposed a method that synthesizes the HR image from an ensemble of different GANs trained with different adversarial objectives [33]. Finally, Robust Super-Resolution (RSR) [7] employs adversarial attacks to create difficult examples that target the model's weaknesses.

2.3. Transformers

Visual transformers [40] are state of the art architectures in deep learning that have surpassed traditional CNN methods in a variety of tasks in computer vision, such as detection [6, 54] and segmentation [6, 46]. It is worth mentioning that this novel approach has also obtained state of the art results in the SR task, with architectures such as SwinIr [25].

SwinIR [25] is the current state of the art in real-world SR. This architecture consists of mainly 3 parts: shallow feature extraction, deep feature extraction and high quality image reconstruction.

Firstly, the shallow features are obtained via a 3×3 convolutional stem layer, which allows an stable optimization and better performance [45] on the Vision transformers (ViT) of the subsequent layers. To obtain the deep features, a series of residual Swin Transformer blocks (RSTB) and a 3×3 convolution is used. Each RSTB is formed from a series of Swin Transformer Layers (STL) [29] with a residual connection. The STL architecture replaced the multi-head self-attention proposed by [40] for a module based on

shifted windows that computes self-attention within non-overlapping local windows. Lastly, for the reconstruction module, a sub-pixel convolution [34] layer that learns an array of upscaling filters maps the LR features into the HR output.

3. Approach

3.1. Baseline

The proposed baseline without using deep learning methods is bicubic interpolation. We upsampled the LR images to the HR images resolution using this technique. With this baseline we want to highlight the importance of using neural networks to obtain higher quality results, particularly associated with computer vision tasks. We also propose a baseline with deep learning methods. This baseline consists of pretrained models in the SwinIR [25] architecture to upscale images by a factor of 2. These models were trained in the DIVerse 2K (DIV2K) [1] and Flickr2K [1] datasets, which are recognized datasets in the context of SISR. Table 1 shows the evaluation metrics on the validation set for these methods.

From Table 1 we could observe that the pretrained SwinIR [25] networks obtained higher metrics in the validation set. However, these differ only by a few decimal places.

Since the HR and LR images are taken using two different devices, the reconstructed HR image is not consistent with the colors of the annotation. Besides this, Figure 2 shows that the reconstructed image lacks clear definition of small objects.

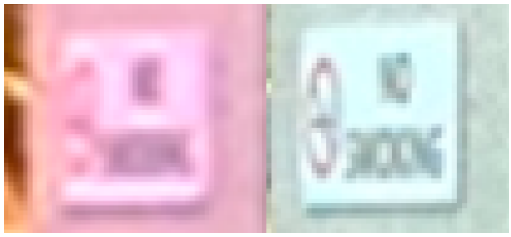


Figure 2. Comparison of small details between reconstructed (left) and ground-truth (right) images for the bicubic interpolation baseline. Notice that in the ground-truth image it is possible to distinguish a "No smoking" sign.

3.2. Proposed Method

Our proposed method is based on SwinIR [25] (Figure 3), which is the state of the art method in SISR.

The network consists of a shallow feature extraction encoder, deep feature extractor and a High Quality decoder to reconstruct the HR image. We maintain the shallow feature extractor, which uses a 3×3 convolutional layer. Therefore, we extract the deep features with a set of Residual Swin Transformer Blocks (RSTB) and a 3×3 convolutional layer. Each RSTB contains a series of Swin Transformer Layers (STL), which is where the attention mechanism takes place.

In practice, we calculate for a window local feature X the attention matrix as,

$$\text{Attention} = \text{SoftMax}\left(\frac{QK^T}{\sqrt{d}} + B\right)V$$

where B is the learnable positional encoding, d is the corresponding dimension of the keys and queries, and Q,K,V are:

$$Q = X \cdot P_Q, K = X \cdot P_K, V = X \cdot P_V$$

where P_Q, P_K, P_V are projection matrices shared across different windows. Finally, to complete the multi-head self-attention (MSA) we concatenate all of these results. Finally, in the STL we employ a multi-layer perceptron (MLP) and intermediate LayerNorm [2] operations for further feature transformation with GELU non-linearities.

Finally, the reconstruction module uses a sub-pixel convolutional network [34]. SwinIR [25] is constructed in a way such that shallow features (low frequencies) are directly transmitted to the reconstruction module with a skip connection. This allows the deep feature extractors to focus on high frequencies mainly.

We propose to vary the standard values described in the SwinIR [25] for the multi-head self-attention (MSA) mechanism, from 6 to 4 in each STL. We found that less attention heads adjusted better for the ImagePairs [17] dataset. We maintain the embedding dimension in 180 and the RSTB depth with 6 layers.

We use the *l1* norm cost function since it allows to make comparisons pixel by pixel and include the absolute value of the penalty term, be robust to outliers and have sparse solutions, such that,

$$L = \left\| \widehat{HR} - HR \right\|_1$$

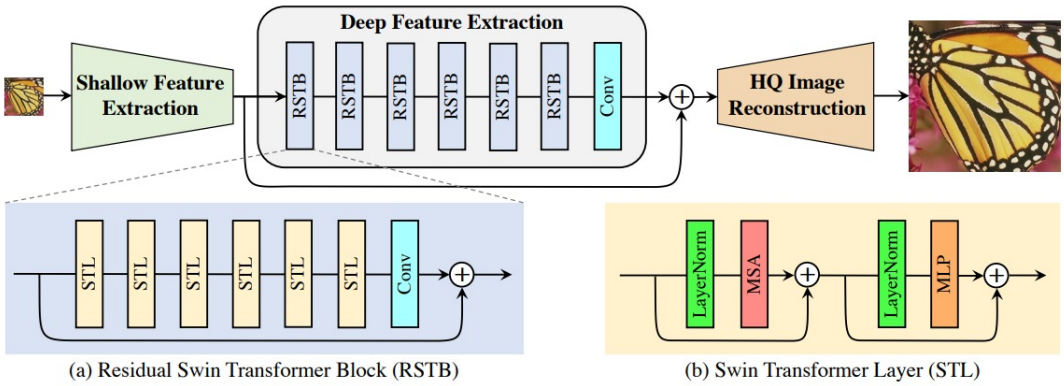
4. Experiments

4.1. Dataset

In the SR context, there exist mainly 2 types of datasets: simulated and real-world. The simulated datasets generate LR images from the HR ones via a degradation procedure. In contrast with simulated datasets, real-world datasets capture the same scene with different cameras or configurations [5]. Simulated datasets are an issue because the degradation process introduces noise, which might not follow the same degradation parameters (i.e. the θ_D in equation 1 may vary drastically) obtained for the real world cases that do not incorporate all of these artefacts. This difference in the acquisition process encourages models to learn how to reverse more complicated degradation functions.

Table 1. Baseline results on the validation set.

Experimentation	PSNR	SSIM	PSNR_Y	SSIM_Y	Pretrained
Bicubic Interpolation	21,130	0,735	-	-	-
SWINIR2X_classicalSR	22,230	**0,812**	24,910	**0,878**	DIV2K
SWINIR2X_classicalSR	**22,240**	**0,812**	**24,920**	**0,878**	DIV2K+Flickr2K



(a) Residual Swin Transformer Block (RSTB) (b) Swin Transformer Layer (STL)

Figure 3. SwinIR [25] architecture for SISR.

The ImagePairs dataset [17] consists of 11,421 pairs of LR-HR images captured using a beam-splitter, which categorizes it into the real-world datasets. The beam-splitter allows both images to be captured using different camera specifications, as shown in Table 7. Figure 1 exhibits the input (LR) and expected output (HR) of several training images in the dataset. It is evident that the images differ not only in the definition of small objects, but also in other characteristics like the luminescence, color, contrast, etc.

Table 2 shows the proposed distribution of pairs in the dataset for which train, valid and test sets represent approximately 65%, 10% and 25% of the whole dataset, respectively.

Table 2. Distribution of image pairs in the dataset

Number of LR-HR pairs	
Train	7,449
Valid	1,142
Test	2,830
Total	**11,421**

There are more real-world datasets developed using different camera parameters such as RealSR [5] for SR and See-In-the-Dark [8] for the related task of Image Signal Processing (ISP). There are also a few standard datasets for SR such as Set5 [3], Set14 [53], Urban100 [16] and DIV2K [1]. Finally, there are many datasets designed for other tasks that are adapted to be used for SR via artificial generation of LR images by a down-sampling algorithm. Examples of these datasets include ImageNet [11] and The Berkeley segmentation dataset [30].

4.2. Evaluation metrics

We use perceptual and computational metrics to quantitatively evaluate our method. The perceptual metrics are based on human's perception of the image quality; for example, the Mean Opinion Score asks a group of human raters to give a score between 1 and 5 to each of the images. Conversely, computational metrics, like the Peak Noise-to-Signal Ratio (PSNR), the Structural Similarity Index (SSIM), and the Normalized Mean Square Error (NMSE) compare the prediction with the expected output through a mathematical formula [44]. These approaches are not necessarily consistent with each other. In this sense, the computational metrics are usually chosen because they are less expensive and yield more reproducible results; however, they are unable to fully capture the human visual perception [44].

We use the PSNR since it is one of the most important reconstruction metrics. We define it as,

$$PSNR = 20 \cdot \log_{10} \left(\frac{L^2}{\frac{1}{N} \cdot \sum_{i=1}^{N} \left(\mathbf{HR}(i) - \widehat{\mathbf{HR}}(i) \right)^2} \right) \tag{3}$$

where L is the maximum pixel value, $\mathbf{HR}$ is the truth image, $\widehat{\mathbf{HR}}$ is the reconstruction and N is the number of pixels. Moreover, we use the SSIM because it measures the structural similarity in terms of comparisons of luminance (C_l), contrast (C_c) and structures (C_s) [44], each defined as follows:

$$C_l(\mathbf{HR}, \widehat{\mathbf{HR}}) = \frac{2 \cdot \mu_{HR} \cdot \mu_{\widehat{HR}} + C_1}{\mu_{HR}^2 + \mu_{\widehat{HR}}^2 + C_1}$$

$$C_c(\mathbf{HR}, \widehat{\mathbf{HR}}) = \frac{2 \cdot \sigma_{HR} \cdot \sigma_{\widehat{HR}} + C_2}{\sigma_{HR}^2 + \sigma_{\widehat{HR}}^2 + C_2}$$

$$C_s(\mathbf{HR}, \widehat{\mathbf{HR}}) = \frac{\sigma_{HR,\widehat{HR}} + C_3}{\sigma_{HR} \cdot \sigma_{\widehat{HR}} + C_3}$$

where μ and σ represent the mean and standard deviation of intensities in each image, $\sigma_{HR,\widehat{HR}}$ represents the correlation coefficient between both images and C_1, C_2, C_3 are constants for stability. Finally, the SSIM is the product of these comparisons weighted by control parameters α, β, γ :

$$SSIM(\mathbf{HR}, \widehat{\mathbf{HR}}) = C_l^{\alpha} \cdot C_c^{\beta} \cdot C_s^{\gamma} \tag{4}$$

4.3. Validation experiments

We performed a series of experiments by varying certain hyperparameters (one by one), the loss function and the network architecture. We show this validation results in Table 3.

We obtained the lowest validation metrics when varying the loss function from *l1* to *ssim* and by increasing the learning rate by 2 orders of magnitude. In the Table 3 the only experiment that has 5000 iterations was the one when we varied the learning rate. The reason is that training diverged after these iterations, which could be inferred due to the tendency that led to very low metrics. We could also notice that when we modified the self-attention heads from 6 to 4, we obtained the highest metric in the validation set for 10000 iterations. For this model we had a PSNR of $23,420$ and a SSIM of $0,754$. We accomplished the highest SSIM metric $(0,812)$ with the pretrained models in DIV2K and DIV2K+Flickr2K.

We also performed a series of fine tuning experiments taking as a starting point the DIV2K [1] dataset. We present these results in table 4.

In this case, we found that by training with fine tuning during more epochs, both the PSNR and SSIM increased. This indicates that what the model learned in the DIV2K [1] was useful for the ImagePairs [17] dataset as well. By fine tuning for 30000 iterations, we obtained the highest metric among all of the experiments in the validation set.

4.4. Evaluation Experiments

By testing over the test set we found that our models were in fact learning to map the LR images to HR in the context of the ImagePairs [17] dataset. The metrics were very similar among the test and valid set, which demonstrates the model is not doing overfitting.

We obtained metrics for the best model trained for 10000 iterations and the best model obtained via fine-tuning (the one with 30000) iterations in the valid test. This model had the highest metric in the test set as well, with a PSNR of 23,470 and a SSIM of 0,784 (Table 5). For comparison purposes, we also tested the fine tuning method with 10000 iterations.

It is important to highlight that the model trained from scratch with the ImagePairs [17] dataset got higher metrics that the model pretrained in DIV2K [1] and fine-tuned for 10000 iterations. This shows this architecture´s high power of generalization. We also show some qualitative results over the test set in Figures 4, 5, 6.

The table 6 shows the state of the art methods in the ImagePairs [17] dataset. This positions us within the current best methods for SISR for the given dataset. Also, we get the highest metric according to the SSIM metric. According to Vaezi et al. [17], all of the methods in Table 6 were trained for 150000 iterations, wheres we trained our fine-tuned model for 30000 iteration and the one modified with less attention heads for only 10000. This implies a lower computational cost and a faster convergence of the model.

5. Discussion

First of all, it is important to mention how impressive is that the metric decreases so much only by increasing the learning rate from $2e-4$ to $2e-2$. This indicates that the associated surface to the *l1* norm loss is very irregular and we should approach the gradient carefully because training can diverge for a large number of epochs. Thus, we had to minimize the error with small steps, given by a very small learning rate.

Also, the metrics decreased a lot when trying to minimize the loss associated to *ssim* instead of *l1*. This shows how important it is to define a proper loss function. By minimizing the loss to *l1*, both the PSNR and SSIM increased. However, this did not happen when defining the loss function to SSIM.

We also tried to implement a weight decay of 0.01, which did not improve the metrics so much. This might be due to the fact that the loss function is in term of the *l1* norm, whereas weight decay regularizes the cost function using the *l2* norm, which might not be very useful for SISR.

We varied the reconstruction module of the network as well. The pixelshuffle upsampler employs a convolutional layer before upsampling that takes in a set of features with the embedding dimension as in channels and a dimension of out channels predefined in 64. Then, it upsamples the features with a set of convolutions and sub-pixel convolutional layers [34]. The main difference between these reconstruction and pixelshuffledirect module is that the last one only has 1 convolution and 1 sub-pixel [34] layer. Even though this experiment allowed the model to save parameters, the improvements were not significantly.

We achieved the main improvements by reducing the embedding dimension, the depth of the each RSTB module, i.e. the number of STL blocks, from 6 to 4, and by reducing the number of self-attention heads from 6 to 4. We performed each of these experiments separately. The model that got better results in the valid test was the one

Table 3. Experimentation with multiple hyperparameters and SwinIR architecture in the validation set.

Experimentation	PSNR	SSIM	PSNR_Y	SSIM_Y	Pretrained	wdecay	Loss	Learning rate	Upsampler	Embed_dim	Depth	Num_heads	Iterations
Interpolation	21,130	0,735	-	-	-	-	l1	2,00E-04	pixelshuffle	180	6	6	-
SWINIR2X_classicalSR	22,230	**0,812**	24,910	**0,878**	DIV2K	-	l1	2,00E-04	pixelshuffle	180	6	6	-
SWINIR2X_classicalSR	22,240	**0,812**	24,920	**0,878**	DIV2K+Flickr2K	-	l1	2,00E-04	pixelshuffle	180	6	6	-
SWINIR2X_classicalSR	22,650	0,743	25,000	0,795	-	0.01	l1	2,00E-04	pixelshuffle	180	6	6	10000
SWINIR2X_classicalSR	4,930	0,003	8,920	0,010	-	-	ssim	2,00E-04	pixelshuffle	180	6	6	10000
SWINIR2X_classicalSR	6,820	0,015	8,000	0,255	-	-	l1	2,00E-02	pixelshuffle	180	6	6	5000
SWINIR2X_classicalSR	22,850	0,745	25,110	0,799	-	-	l1	2,00E-04	pixelshuffledirect	180	6	6	10000
SWINIR2X_classicalSR	23,220	0,758	25,370	0,812	-	-	l1	2,00E-04	pixelshuffle	90	6	6	10000
SWINIR2X_classicalSR	23,140	0,737	25,520	0,805	-	-	l1	2,00E-04	pixelshuffle	180	4	6	10000
SWINIR2X_classicalSR	**23,420**	0,754	**25,710**	0,809	-	-	l1	2,00E-04	pixelshuffle	180	6	4	10000
SWINIR2X_classicalSR	23,340	0,760	25,590	0,813	-	-	l1	2,00E-04	pixelshuffle	180	6	4	20000
SWINIR2X_classicalSR	22,880	0,745	25,120	0,797	-	-	l1	2,00E-04	pixelshuffle	120	6	4	10000
SWINIR2X_classicalSR_FineTune	23,300	0,753	25,620	0,814	DIV2K	-	l1	2,00E-04	pixelshuffle	180	6	6	10000

Table 4. Fine tuning experimentation with SwinIR architecture in the validation set.

Experimentation	PSNR	SSIM	PSNR_Y	SSIM_Y	Pretrained	Iterations
SWINIR2X_classicalSR_FineTune	23,300	0,753	25,620	0,814	DIV2K	10000
SWINIR2X_classicalSR_FineTune	23,500	0,765	25,660	0,821	DIV2K	20000
SWINIR2X_classicalSR_FineTune	**23,620**	**0,778**	**25,780**	**0,826**	DIV2K	30000

Table 5. Results of the final methods over the test set.

Experimentation	PSNR	SSIM	PSNR_Y	SSIM_Y	Pretrained	Iterations
SWINIR2X_classicalSR_FineTune	23,120	0,764	25,650	0,824	DIV2K	10000
SWINIR2X_classicalSR_FineTune	**23,470**	**0,784**	**25,890**	**0,834**	DIV2K	30000
SWINIR2X_classicalSR	23,270	0,759	25,760	0,817	-	10000

Table 6. Comparison of state-of-the-art SISR algorithms on ImagePairs dataset

Model	Train data	PSNR	SSIM
SRGAN [24]	DIV2K	21,906	0,699
WDSR [50]	DIV2K	21,299	0,697
EDSR [27]	DIV2K	21,298	0,697
SRGAN [24]	ImagePairs	22,161	0,673
Ours-Attention heads	ImagePairs	23,270	0,764
Ours-FineTune	DIV2K & ImagePairs	23,470	**0,784**
WDSR [50]	ImagePairs	23,805	0,767
EDSR [27]	ImagePairs	**23,845**	0,764

with less attention heads. All of these results might be due to the complexity of the network and the high amount of parameters that need we have to calculate and optimize. Also, it is possible that many of the attention heads were not learning important information about the image resolution to enhance it. Not always too many attention heads are better than a few ones. In fact, Michel et al. [31] found that pruning attention heads during test time did not affect the network´s performance significantly.

Even though eliminating some attention heads during training allowed to find the best model during 10000 epochs, training this model for 20000 epochs (Table 3) did not improve its performance, which indicates a local minimum has been reached and it is hard to get further enhancement.

One possible improvement to our method is to combine those combinations in which we reduced the dimensional complexity of the network (number of heads, depth and embedding dimension). We only performed one of these experiments and found no further improvement, but many more combinations can be tried. Another possibility could be to modify the shallow feature extractor with convolutions that extract features at different scales or in a more sophisticated way. The most recent implementation only performs one 3×3 convolution at the input scale.

We would also like to indicate that the metrics of the state-of-the-art models are relatively low (Table 6) in the ImagePairs dataset. There are 2 possibilities: the dataset is very challenging or they did not acquire the images in the best way possible.

We know that real LR generation is a challenging task. However, the investigators must be very careful when using the opto-mechanical layout of dual camera combiner with the beam-splitter they propose. We found some cases in which was clear that the images had a small difference in the time lapse of data acquisition (Figure 7 and Figure 8). Moreover, there were drastic color changes between a LR and its corresponding HR image. All of these differences between images make it almost impossible for the neural network to converge to a satisfactory result since we did not train it for deformation scenes. It is important to work with well-defined benchmarks that enhance machine learning technologies. We show some of these images in the Annexes section.

Figure 4. Visual comparison of the finetuned model with 10000 iterations and the model with 4 attetion-heads. Best viewed by zooming.

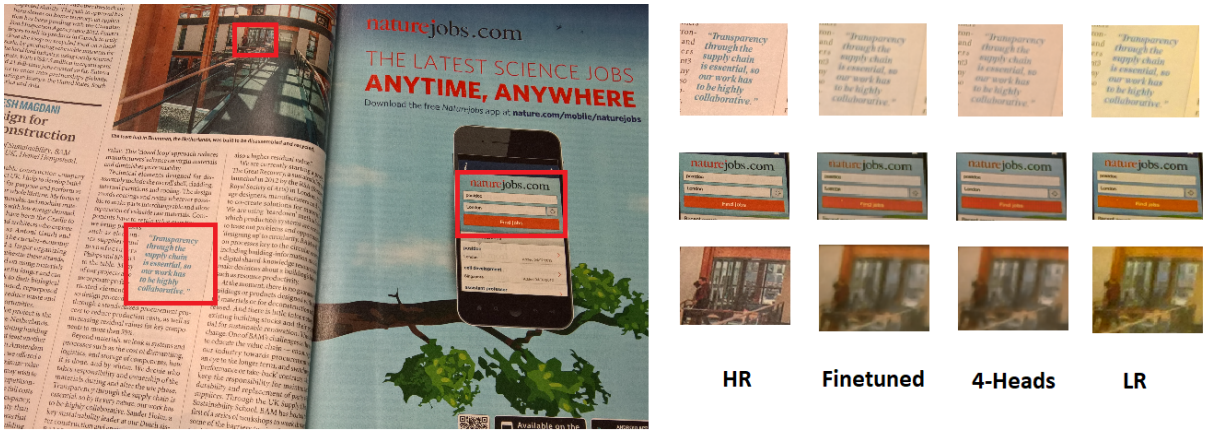


Figure 5. Visual comparison of the finetuned model with 10000 iterations and the model with 4 attetion-heads. Best viewed by zooming.

6. Ethical Considerations

It is important to take into account ethical considerations in the SISR task. This is mainly because different approaches intend to create pixel information values by increasing the image resolution. This could create digital artefacts that are not really there. Thus, we have to critically evaluate the purpose for which we will use this type of technology.

For example, these kind of technologies could help finding criminals recorded on video footage. However, we must rely in really precise algorithms in order not to prosecute an innocent person, who maybe was not even there. Another example is the application of SISR in medical imaging. Although we could recover valuable anatomical details to make a more proper diagnosis, we could also impose small intensity variations that would change the medical professional´s opinion of the diagnosis based on the HR output.

Also, it is important to mention that Reza et al. [17] collected the ImagePairs dataset with certain cameras specifications (Table 7). Therefore, the SR algorithm might be biased towards those specifications. If we try to implement our network on different cameras, color aberrations or focal lengths, it might fail. Consequently, we must provide users with robust models so that they feel safe using these technologies.

Consider as well the case in which a photo was changed subtly by SR algorithms but enough so that another neural network does not recognize its objects no more or classifies them in the wrong class, such as in adversarial attacks. Hence, we must do diverse training to avoid all these scenarios.

Therefore, if we want to use this kind of technology we will have to establish some frameworks in order to obtain quantitative and qualitative results for specific tasks such as traffic surveillance, security monitoring and medical imaging. In this sense, the government of each country should regularize these new technologies to offer users, scientists, surveillance entities, companies and health professionals a set of minimum standards.

References

[1] Eirikur Agustsson and Radu Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *Pro-*

Figure 6. Visual comparison of the finetuned model with 10000 iterations and the model with 4 attetion-heads. Best viewed by zooming.

ceedings of the IEEE conference on computer vision and pattern recognition workshops, pages 126–135, 2017.

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

[3] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012.

[4] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. *arXiv preprint arXiv:1809.11096*, 2018.

[5] Jianrui Cai, Hui Zeng, Hongwei Yong, Zisheng Cao, and Lei Zhang. Toward real-world single image super-resolution: A new benchmark and a new model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3086–3095, 2019.

[6] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[7] Angela Castillo, María Escobar, Juan C Pérez, Andrés Romero, Radu Timofte, Luc Van Gool, and Pablo Arbelaez. Generalized real-world super-resolution through adversarial robustness. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1855–1865, 2021.

[8] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun. Learning to see in the dark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3291–3300, 2018.

[9] Honggang Chen, Xiaohai He, Linbo Qing, Yuanyuan Wu, Chao Ren, Ray E Sheriff, and Ce Zhu. Real-world single image super-resolution: A brief review. *Information Fusion*, 79:124–145, 2022.

[10] Ryan Dahl, Mohammad Norouzi, and Jonathon Shlens. Pixel recursive super resolution. In *Proceedings of the IEEE international conference on computer vision*, pages 5439–5448, 2017.

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[12] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Learning a deep convolutional network for image super-resolution. In *European conference on computer vision*, pages 184–199. Springer, 2014.

[13] Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *European conference on computer vision*, pages 391–407. Springer, 2016.

[14] Muhammad Haris, Gregory Shakhnarovich, and Norimichi Ukita. Deep back-projection networks for super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1664–1673, 2018.

[15] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[16] Jia-Bin Huang, Abhishek Singh, and Narendra 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.

[17] Hamid Reza Vaezi Joze, Ilya Zharkov, Karlton Powell, Carl Ringler, Luming Liang, Andy Roulston, Moshe Lutz, and Vivek Pradeep. Imagepairs: Realistic super resolution dataset via beam splitter camera rig. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 518–519, 2020.

[18] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[20] Deokyun Kim, Minseon Kim, Gihyun Kwon, and Dae-Shik Kim. Progressive face super-resolution via attention to facial landmark. *arXiv preprint arXiv:1908.08239*, 2019.

[21] Jiwon Kim, Jung Kwon Lee, and Kyoung Mu Lee. Accurate image super-resolution using very deep convolutional net-

works. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654, 2016.

[22] Jiwon Kim, Jung Kwon Lee, and Kyoung 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.

[23] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan 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.

[24] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan 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.

[25] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1833–1844, 2021.

[26] Luming Liang, Ilya Zharkov, Faezeh Amjadi, Hamid Reza Vaezi Joze, and Vivek Pradeep. Guidance network with staged learning for image enhancement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 836–845, 2021.

[27] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung 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.

[28] Ding Liu, Zhaowen Wang, Nasser Nasrabadi, and Thomas Huang. Learning a mixture of deep networks for single image super-resolution. In *Asian Conference on Computer Vision*, pages 145–156. Springer, 2016.

[29] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[30] 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 Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, pages 416–423. IEEE, 2001.

[31] Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? *Advances in neural information processing systems*, 32, 2019.

[32] Nathanaël Carraz Rakotonirina and Andry Rasoanaivo. Esrgan+: Further improving enhanced super-resolution generative adversarial network. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3637–3641. IEEE, 2020.

[33] Haoyu Ren, Amin Kheradmand, Mostafa El-Khamy, Shuangquan Wang, Dongwoon Bai, and Jungwon Lee. Real-world super-resolution using generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 436–437, 2020.

[34] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan 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.

[35] Ying Tai, Jian Yang, and Xiaoming 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.

[36] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE international conference on computer vision*, pages 4539–4547, 2017.

[37] Tong Tong, Gen Li, Xiejie Liu, and Qinquan Gao. Image super-resolution using dense skip connections. In *Proceedings of the IEEE international conference on computer vision*, pages 4799–4807, 2017.

[38] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *Advances in neural information processing systems*, 29, 2016.

[39] Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR, 2016.

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[41] Lei Wang, Wei Chen, Wenjia Yang, Fangming Bi, and Fei Richard Yu. A state-of-the-art review on image synthesis with generative adversarial networks. *IEEE Access*, 8:63514–63537, 2020.

[42] Xintao Wang, Ke Yu, Chao Dong, and Chen Change Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 606–615, 2018.

[43] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018.

[44] Zhihao Wang, Jian Chen, and Steven C. H. Hoi. Deep learning for image super-resolution: A survey. *CoRR*, abs/1902.06068, 2019.

[45] Tete Xiao, Piotr Dollar, Mannat Singh, Eric Mintun, Trevor Darrell, and Ross Girshick. Early convolutions help transformers see better. *Advances in Neural Information Processing Systems*, 34, 2021.

[46] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and

efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34, 2021.

[47] Chih-Yuan Yang, Chao Ma, and Ming-Hsuan Yang. Single-image super-resolution: A benchmark. In *European conference on computer vision*, pages 372–386. Springer, 2014.

[48] Wenhan Yang, Jiashi Feng, Jianchao Yang, Fang Zhao, Jiaying Liu, Zongming Guo, and Shuicheng Yan. Deep edge guided recurrent residual learning for image super-resolution. *IEEE Transactions on Image Processing*, 26(12):5895–5907, 2017.

[49] Wenming Yang, Xuechen Zhang, Yapeng Tian, Wei Wang, Jing-Hao Xue, and Qingmin Liao. Deep learning for single image super-resolution: A brief review. *IEEE Transactions on Multimedia*, 21(12):3106–3121, 2019.

[50] Jiahui Yu, Yuchen Fan, Jianchao Yang, Ning Xu, Zhaowen Wang, Xinchao Wang, and Thomas Huang. Wide activation for efficient and accurate image super-resolution. *arXiv preprint arXiv:1808.08718*, 2018.

[51] Linwei Yue, Huanfeng Shen, Jie Li, Qiangqiang Yuan, Hongyan Zhang, and Liangpei Zhang. Image super-resolution: The techniques, applications, and future. *Signal Processing*, 128:389–408, 2016.

[52] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2472–2481, 2018.

[53] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for neural networks for image processing. *arXiv preprint arXiv:1511.08861*, 2015.

[54] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

7. Credits

The data acquisition, distribution and cubic interpolation baseline was done by Sebastián Urrea. The experimentation (training and evaluation over valid and test) was done by Daniel Chacón. Both of the authors contributed in the paper. Sebastián Urrea did Related work, Evaluation metrics and the main.py (test and demo modes). Daniel Chacón did Abstract, Introduction, Approach, Experiments, Discussion and Ethical Considerations.

We would like to thank Cristhian Forigua for his advices and discussions to make this work possible.

8. Annexes

Table 7. Specifications for the LR and HR cameras

Camera	Low-resolution	High-resolution
Image sensor format	1/4	1/2.4
Pixel size	$1.4\mu m$	$1.12\mu m$
Resolution	5MP	20.1MP
Dimensions	**1752 × 1166**	**3504 × 2332**
FOV (H, V)	64°, 50.3°	68.2°, 50.9°
Lens focal length	$2.9mm$	$4.418mm$
Focus	fixed-focus	auto-focus

Figure 7. Low resolution image (left) and its high resolution image (right). We can notice that the car has moved considerably between one image and the other one.

Figure 8. Low resolution image (left) and its high resolution image (right). We can notice that the car has moved a bit between one image and the other one. Also, the sky in the LR image is blue, whereas the HR takes this as white.