Optimization-based Image Filter Design for Self-supervised Super-resolution Training

Klemen Grm, Vitomir Štruc

Univerza v Ljubljani, Fakulteta za Elektrotehniko
klemen.grm@fe.uni-lj.si

Abstract

Single-image super-resolution can be posed as a self - supervised machine learning task where the training inputs and targets are derived from an unlabelled dataset of high-resolution images. For super-resolution training, the derivation takes the form of a degradation function that yields low-resolution images given high-resolution ones. Typically, the degradation function is selected manually based on heuristics such as the desired magnification ratio of the super-resolution method being trained. In this paper, we instead propose principled, optimization-based methods for picking the image filter of the degradation function based on its desired properties in the frequency domain. We develop implicit and explicit methods for filter optimization and demonstrate the resulting filters are better at rejecting aliasing and matching the frequency domain characteristics of real-life low-resolution images than commonly used heuristic picks.

1 Introduction

Single-image super-resolution is the task of recovering high-resolution details from low-resolution observations. Existing work on super-resolution commonly relies on machine learning techniques (e.g., [3, 6, 4, 5, 7]) that learn from artifically generated pairs of aligned low-resolution and high-resolution images, $\mathbf{x}$ and $\mathbf{y}$, respectively. The training data required for the training procedure is typically generated by starting with a high-resolution image dataset and downsampling the images through a process of the following form:

$$\mathbf{x} = H\mathbf{y} \downarrow_d + N, \tag{1}$$

where H is an image filtering operator, $\downarrow$ is the sub-sampling operation, d is the sub-sampling factor, and N is a noise component. In existing work on machine-learning based super-resolution, the degradation process is typically picked using basic heuristics, e.g., a separable Gaussian or Lanczos filter with appropriate width given the desired magnification factor d. Given the generated dataset of $(\mathbf{x}, \mathbf{y})$ pairs, a differentiable model m_θ with free parameters θ is then trained to approximate the inverse process, i.e., to predict an approximation of the high resolution image $\hat{\mathbf{y}}$ given a low-resolution image $\mathbf{x}$ as an input, i.e.:

$$\hat{\mathbf{y}} = m_\theta(\mathbf{x}), \tag{2}$$

by setting the model parameters θ through gradient descent on some appropriate loss function. Here, the loss is typically a distortion measure between the model predictions and the ground-truth high-resolution images, e.g.,

$$\mathfrak{L}(\mathbf{x}, \mathbf{y}) = \|m_\theta(\mathbf{x}) - \mathbf{y}\|_p^p, \tag{3}$$

where p is the order of the norm used to measure the distortion between the model approximations and expected (i.e., ground truth) high-resolution images.

In this paper, we develop a principled methodology to determine the optimal degradation function from (1), i.e., by picking an appropriate discrete image filtering operator for the given super-resolution training problem. Unlike previous approaches, where the filter is heuristically picked based on known classes of continuous filters with desired characteristics in the frequency domain (e.g., the Gaussian and Lanczos family of filters) and extended for image filtering as a separable filter, we develop implicit and explicit methods to design optimal image filters including sampling constraints. Also unlike previous approaches, we separate the filter design into two discrete stages, namely: *(i)* determining constraints on the filter structure and its desired frequency characteristics, and *(ii)*, finding a filter that satisfies the constraints and requirements, using either implicit or explicit optimization. The derived image filters more closely follow the sampling laws than typical heuristic choices used in existing super-resolution work, and are guaranteed to avoid sampling artefacts such as aliasing, ringing and oversharpening. Our work leads to the following key contribution that are presented in this paper:

1. We develop a set of soft- and hard-constraint based filter specifications that allow us to enforce the desired filter structure in the learned filters, i.e., symmetric, separable, or neither;
2. We develop an optimization strategy to find filters that satisfy the above specified constraints with filters that approximate the desired characteristics in the frequency domain, using explicit or implicit filter modeling;
3. We evaluate the resulting filters by comparing the frequency characteristics of filtered images with a real-life dataset of low-resolution images.

Supported by the ARRS Research Program P2–0250 (B).

2 Methodology

In self-supervised super-resolution training, we begin with a set of high-resolution images, which we downsample into low-resolution training inputs using the process described in the Eq (1). We would like the artificially downsampled images to have similar spatial and frequency domain characteristics to real-world low-resolution images in order to ensure that models trained on the artificially derived training pairs can then be applied to real-world super-resolution problems.

Explicit design. We first develop an optimization strategy where the ideal image filter for use in (1) is optimized explicitly. For the purposes of this paper, we limit ourselves to the single task of super-resolution with an $8\times$ magnification factor. Let $\mathbf{y}$ be an RGB image sampled with the sampling frequency f_s. Given the chosen magnification factor, we would ideally like an image filtering operator H, implementing a spatial filter $\mathbf{w}$, such that $\mathbf{y} * \mathbf{w}$ retains all the frequency components of $\mathbf{y}$ from 0 to $\frac{f_s}{8}$, and suppresses all the higher frequency components. In classical filter design theory, this is represented by a desired gain of $-20dB$ at the threshold frequency, and a limitingly narrow transition band. For the continuous case, there are known analytical solutions such as Gaussian and Lanczos ($sinc$) filters which, given a filter size budget, offer an optimal tradeoff between the gain in the passband, the width of the transition band, and the suppression of the stopband, as shown in the Figure 1.

However, the continuous case for 1-dimensional signals is not trivially extensible to sampled and re-sampled two dimensional signals, such as the images we are considering using for the self-supervised training of super-resolution models. Firstly, an ideal filter in the continuous case has infinite spatial support, whereas the support of useful filters in our case is limited by the resolution of the images. This is due to the fact that the window size of the discrete filters used has to be considerably smaller than the resolution of our groundtruth (high resolution) images, otherwise the boundary conditions of convolution (e.g., zero-padding or reflection-padding) dominate the resulting filtered image. We show the discrepancy between the mean axial spectrum gain of ideal continuous Gaussian and $sinc$ filters, and the real-life results on an extensive image dataset, in the Figure 2.

We notice that when comparing the analytical gain to the observed image spectra, the ideal Gaussian filter has a stronger gain in the passband and far more suppression of the stopband. In turn, the gain characteristic computed from the filtered and re-sampled images has a significantly lower threshold frequency, and its suppression in the stopband is limited by the 8-bit image quantization.

The experimental results of the $sinc$ filter match its analytically determined gain more closely, but the threshold frequency is still lowered significantly, and the passband overamplification (i.e., ringing) effect is even larger than in the analytically determined gain characteristic.

Given the discrepancy between analytical and experimental results, we would like to learn an optimal discrete, resampling filter from the data itself. The learning objective is the contents of a discrete filter window, $\mathbf{w} \in \mathbb{R}^{s\times s}$, where s is the size of the window (measured in samples). In the explicit case, we directly optimize the contents of $\mathbf{w}$ by minimizing the loss function:

$$\mathfrak{L}(\mathbf{y}, \mathbf{w}) = |\mathbf{y} - ((\mathbf{y} * \mathbf{w}) \downarrow) \uparrow|, \tag{4}$$

where $\mathbf{y}$ is an image from the dataset, and $\downarrow$ and $\uparrow$ are the $8\times$ downsampling and upsampling operators, respectively. The explicit formulation of the optimization problem allows us to directly specify a number of hard constraints on the optimization problem, namely,

1. separability, by setting $\mathbf{k} = [w_1, w_2, ..., w_s]^\top; \mathbf{w} = \mathbf{k}\mathbf{k}^\top$,
2. symmetry, by setting $\mathbf{k} = \left[w_1, w_2, ..., w_{\lfloor \frac{s}{2} \rfloor}, w_{\lfloor \frac{s}{2} \rfloor - 1}, ..., w_1\right]^\top$; $\mathbf{w} = \mathbf{k}\mathbf{k}^\top$, and
3. normalization, by setting $\mathbf{w} = \left(\sum_i k_i\right)^{-2} \mathbf{k}\mathbf{k}^\top$,

where either $\mathbf{k} \in \mathbb{R}^s$ (cases 1. and 3.), or otherwise $\mathbf{k} \in \mathbb{R}^{\lfloor \frac{s}{2} \rfloor}$ (case 2.). The hard constraints reduce the dimensionality of the optimization problem, making it more tractable when optimizing (4) over a large dataset of diverse images using stochastic gradient descent with regards to the explicit kernel. When explicitly optimizing the filter without constraints, the optimization diverges towards the suboptimal solution shown in Figure 3.

Implicit design. For the implicit approach, we use the deep linear generator model from [1]. Here, the network used to design the image filter consists of a sequence of convolutional neural network layers without biases and non-linear activations. This means the layer sequence has the same expressive power as a single larger filter, however, as an optimization task, learning the parameters of the deep linear generator is much better conditioned than learning the same large filter explicitly.

The filter implicitly learned by the deep linear generator can also be recovered, simply by taking the discrete finite impulse response of the network - i.e., its output given a Dirac δ-signal as an input. Recovering the learned filter in this manner allows us to use the implicitly learned filter directly, without needing to perform inference over the deep linear generator network. It also allows us to regularize the learning process by forcing the learned filter's centre of mass to lie in the center of the kernel window (to eliminate kernel shift), and by forcing the sum of the learned filter towards 1, to enforce unit gain at DC. Specifically, if $k_{i,j}; i, j \in \omega$ is the recovered implicitly learned kernel, we enforce the zero-phase centre of mass for an odd filter using the regularization term

$$\mathfrak{L}_c = \left\| (x_0, y_0) - \frac{\sum_{i,j} k_{i,j} \cdot (i, j)}{\sum_{i,j} k_{i,j}} \right\|_2, \tag{5}$$

where (x_0, y_0) are the coordinates of the window centre. The regularization term directly penalizes the Euclidean distance between the actual centre of mass of the kernel

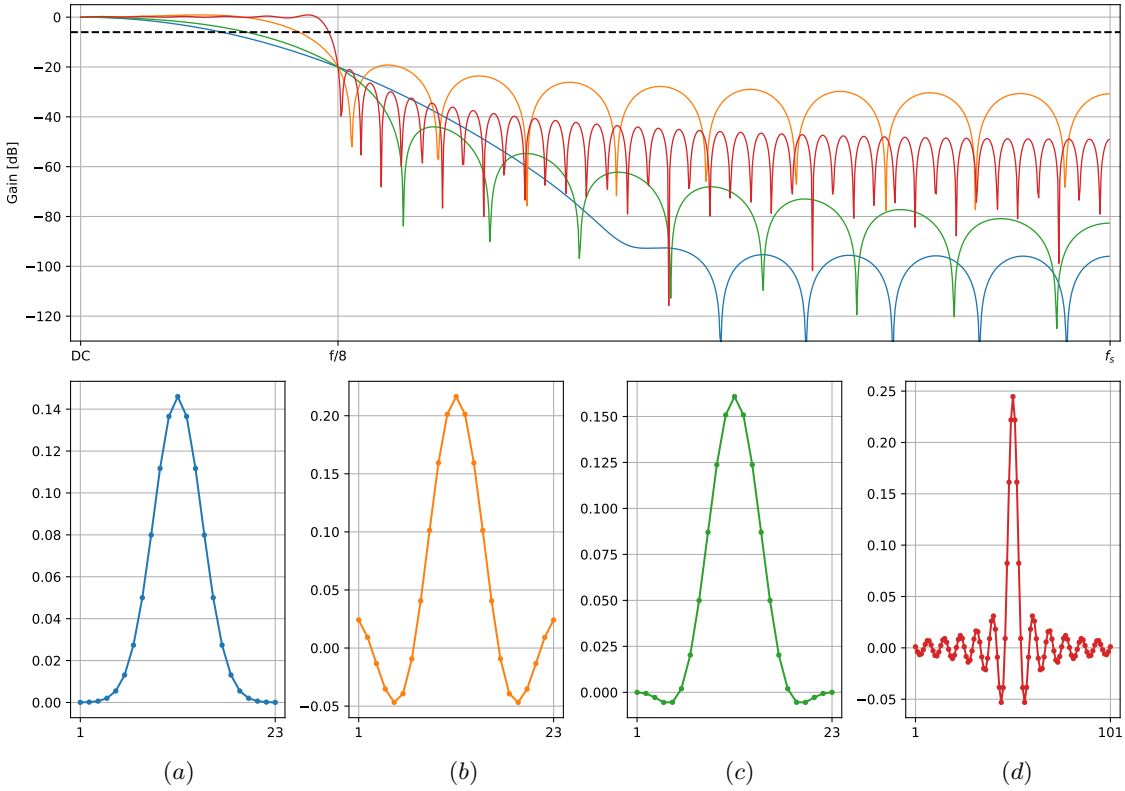

Figure 1: Impulse responses of ideal odd (zero-phase) downsampling filters for $-20dB$ gain at $\frac{f_s}{8}$ in the continuous case. (a), Gaussian, 23 taps; (b), $sinc$, 23 taps; (c), Lanczos, 23 taps; (d), $sinc$, 101 taps; and their gain characteristics in the frequency domain *(top)*. Impractically large filters are required for the desired frequency characteristic.

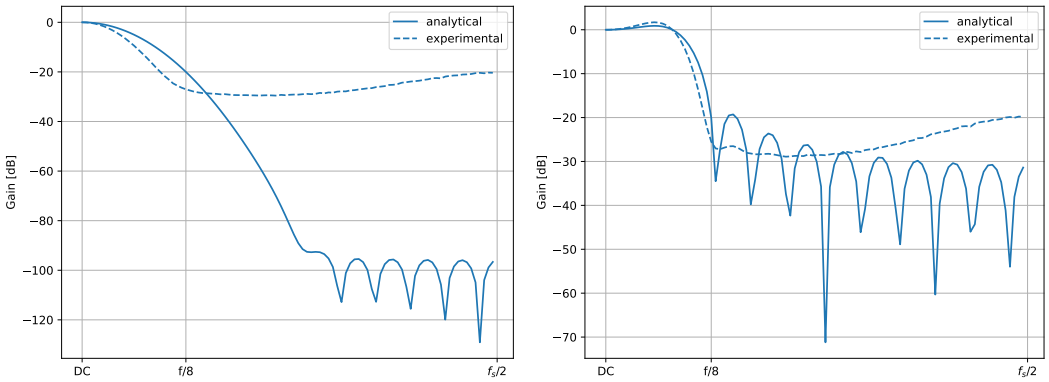

Figure 2: A comparison of analytical gain characteristics in the continuous case and the observed gains averaged over a large image dataset, for the Gaussian (left) and $sinc$ filters (right). Note that the experimentally determined gain characteristics bottom out faster due to the 8-bit image quantization.

and the desired zero-phase centre of mass. Furthermore, we enforce kernel normalization (i.e., unit gain at DC) using the following regularization term

$$\mathfrak{L}_n = \left| 1 - \sum_{i,j} k_{i,j} \right|. \tag{6}$$

Let $\mathbf{y}$ be the input image, and $m_k(\cdot)$ be the implicit kernel model, which models the kernel $\mathbf{k}$, the final loss used to train the implicit kernel model is then

$$\begin{aligned} \mathfrak{L}(\mathbf{y}, m_k) = |\mathbf{y} - m_k(\mathbf{y}) \downarrow\uparrow| + \\ + \lambda_c \mathfrak{L}_c(m_k(\delta)) + \lambda_n \mathfrak{L}_n(m_k(\delta)), \end{aligned} \tag{7}$$

where λ_c and λ_n are weights for the centre-of-mass and normalization regularization terms. We set them as $\lambda_c = 10^{-2}, \lambda_n = 10^{-3}$ using a logarithmic grid search.

3 Results

We train the implicit and explicit models on the VGGFace2 image dataset [2] to convergence using the above described loss functions and constraints.

We show the filters learned by the explicit model in the Figure 3. Note that without constraints, the learned filter does not have the desired structure, and fails to suppress the frequency components in the stopband. The fil-

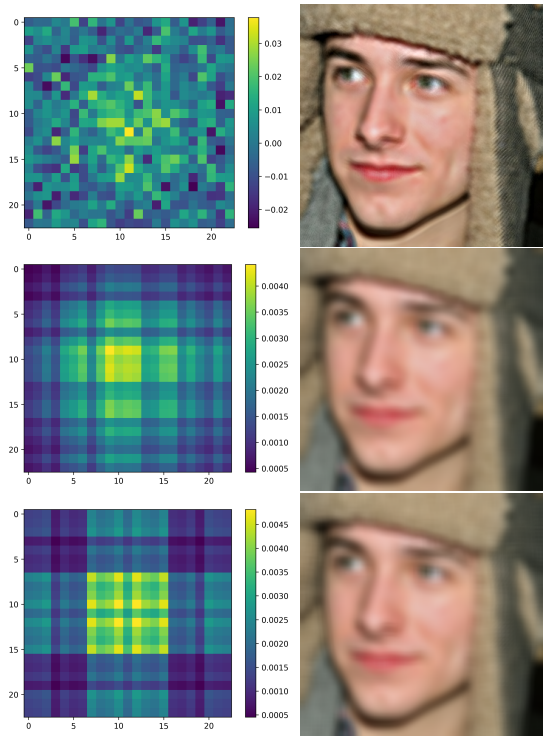

Figure 3: Filters learned by the explicit model without constraints (top), using the separability constraint (centre), and using the symmetry constraint (bottom). The figure shows the filters (left) and the result of applying the filters to an image from the test set (right).

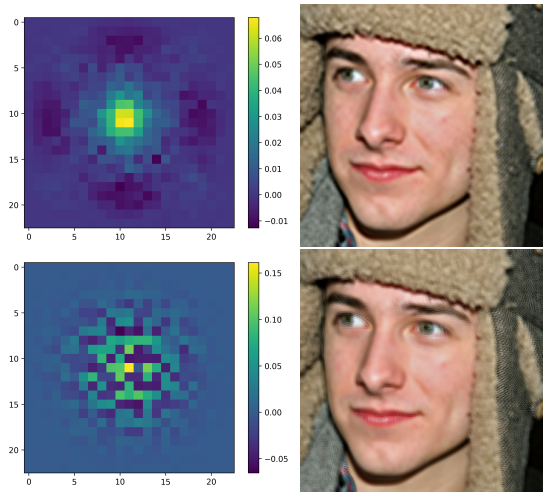

Figure 4: Filters learned by the implicit model without kernel regularization (top) and with the described regularization terms (bottom).

ters learned by the implicit model are shown in Figure 4. Without the regularization terms described in the previous section, the implicit model learns a $sinc$-like filter, whereas if we use the regularization terms, the structure of the filter is less regular. In this case, the regularized model has better frequency characteristics, and is considered the main result of this paper.

Next, we evaluate the filters learned by the proposed method by using them to filter and downsample images from the test set of the VGGFace2 dataset, using a downsampling factor of 8. We compare the frequency characteristics of the downsampled images and small patches

Table 1: Results of the spectrum matching experiment.

Filter	**Spectral deviation (↓)**
Gaussian (analytical)	$7.1dB$
$sinc$ (analytical)	$6.9dB$
Lanczos (analytical)	$6.9dB$
Explicit model, no constraints	$9.1dB$
Explicit model, constraints	$7.3dB$
Implicit model, no regularization	$6.8dB$
Implicit model, regularization	**6.6dB**

of the high-resolution images with the same resolution. We take the average axial spectrum of both over the entire dataset and compare the mean absolute deviation in dB. The results are shown in the Table 1. Using the described loss function and regularization terms, the implicit model is able to learn better image filters than the analytically derived separable filters that don't account for sampling artefacts. The explicit kernel model requires hard constraints to tractably learn filters with the desired frequency characteristics, which prevents it from surpassing the performance of analytically derived filter kernels.

4 Conclusion

In this paper, we have presented a data-based filter design technique to prepare pairs of low- and high-resolution images for super-resolution training. The results show that our learned filters are better suited for image downsampling for this task than the heuristically picked, analytically designed filters typically used for this purpose. As part of our future work we plan to apply the learned filters for generating aligned pairs of high- and low-resolution face images and evaluate their influence on the performance of the learned super-resolution models.

References

[1] S. Bell-Kligler, A. Shocher, and M. Irani. Blind super-resolution kernel estimation using an internal-gan. *Advances in Neural Information Processing Systems*, 32, 2019.

[2] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE international conference on automatic face & gesture recognition (FG 2018)*, pages 67–74. IEEE, 2018.

[3] J. Kim, J. K. Lee, and K. M. 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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) workshops*, 2018.