

# Multiobjective optimization of deep neural networks with combinations of Lp-norm cost functions for 3D medical image super-resolution

Karl Thurnhofer-Hemsi<sup>a</sup>, Ezequiel López-Rubio<sup>a</sup>, Núria Roé-Vellvé<sup>b</sup> and Miguel A. Molina-Cabello<sup>a</sup>,

<sup>a</sup> *Department of Computer Languages and Computer Science. University of Málaga.*

*Bulevar Louis Pasteur, 35. 29071 Málaga, Spain.*

*E-mail: {karlkhader,ezeqlr,miguelangel}@lcc.uma.es*

*Institute of Biomedical Research of Málaga (IBIMA)*

*C/ Doctor Miguel Díaz Recio, 28, 29010, Málaga, Spain*

<sup>b</sup> *Biomedical Research Networking Center in Bioengineering*

*Biomaterials and Nanomedicine (CIBER-BBN), Barcelona, Spain*

*E-mail: nuriaroe@gmail.com*

**Abstract.** In medical imaging, the lack of high-quality images is present in many areas such as magnetic resonance (MR). Due to many acquisition impediments, the generated images have not enough resolution to carry out an adequate diagnosis. Image super-resolution (SR) is an ill-posed problem that tries to infer information from the image to enhance its resolution. Nowadays, deep learning techniques have become a powerful tool to extract features from images and infer new information. In MR, most of the recent works are based on the minimization of the errors between the input and the output images based on the Euclidean norm. This work presents a new methodology to perform three-dimensional SR based on the combination of Lp-norms in the loss layer. Two multiobjective optimization techniques are used to combine two cost functions. The proposed loss layers were trained with the SRCNN3D and DCSRN networks and tested with two MR structural T1-weighted datasets, and then compared with the traditional euclidean loss. Experimental results show significant differences in terms of Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM) and Bhattacharyya Coefficient (BC), while the residual images show refined details.

Keywords: Super-resolution, magnetic resonance images, convolutional neural networks, multiobjective optimization, Lp-norm

## 1. Introduction

The improvement of image quality and resolution is a constant aim in medical imaging, due to the critical importance of these images to find the correct diagnosis and treatment for patients. This is not only reflected in the optimization of acquisition techniques, for instance in the case of magnetic resonance imaging (MRI), but also at the post-processing stage, with an ever-increasing interest in new, improved algorithms. Enhancing resolution is particularly relevant in this

area, given the need to inspect the details of anatomical structures and to locate functional information in a more precise way. Many machine-learning approaches are being proposed for medical imaging applications, and deep learning is becoming increasingly popular among them [1]. The interest of these algorithms for medical imaging is developing as they evolve towards greater efficiency and reliability. The improvement of image resolution is a fundamental step towards attaining an adequate performance in subsequent phases of the medical image processing pipeline. For example,

segmentation of the regions of the human brain by clustering is a key task [2], which can benefit from the enhancement of the quality of the input image. Deep learning-based MRI super-resolution technology has the potential to become a commonplace procedure in all MRI medical protocols [3].

These methods have largely overcome the traditional interpolation and spline-based approaches to increase image resolution, which typically causes blurring. Among them, example-based methods have become popular as super-resolution techniques [4]. Some exploit the internal similarities of the image [5], while others use external datasets to learn mapping patterns between LR and HR images [6]. Recently, an example-based super-resolution algorithm, the SRCNN convolutional neural network [7], and its 3D version SRCNN3D [8] have obtained great attention because of their ability to learn an end-to-end mapping between LR and HR images, thus avoiding to learn from dictionaries or manifolds to model the high-resolution space.

More generally, convolutional neuronal networks (CNNs) have demonstrated excellent performance in image and video processing. These methods are under constant development [9,10,11], and they have been successfully applied in detection and recognition of objects, classification of images or within recommender systems [12,13]. This has also been facilitated by the power of the new graphical acceleration devices (GPU), and more specific hardware developments [14]. Hundreds of articles based on the development of CNNs have been published in several areas [15] including medical image analysis [4,5,16,17], where the use of CNNs is progressively expanding.

Deep learning neural networks use loss functions that are commonly based on the squared Euclidean norm. However, the use of alternatives like  $L_p$ -norms has attracted attention for a variety of tasks. The  $L_p$ -norm is analogous to the Euclidean norm where the exponent 2 is substituted by an alternative value  $p$ . The robustness properties of the  $L_p$ -norm have been extensively studied, as seen in [18,19] and references therein. Standard minimization of the Euclidean norm is known to be optimal for Gaussian noise, while  $L_p$ -norms can perform better for datasets containing outliers and non-Gaussian noise [20]. Typical machine learning applications such as binary classification have already taken advantage of  $L_p$ -norm techniques [21,22]. Optimal control has also employed this kind of norms [23]. The favorable properties of the  $p$ -norm to manage sparsity in matrices and vectors have

been widely recognized [24,25,26,27], which has led to applications in the feature selection field [28].

Using an  $L_p$ -norm approach can be useful to reduce the effect of outliers in minimization problems when  $p < 2$  values are selected. If  $p$  lies within the  $[1, 2]$  interval it will also fulfill the formal definition of a geometrical norm. In MRI applications, using  $p$  within this range can allow compensating for noise and artifacts that are often present in the images, thereby increasing the robustness of an SR algorithm. We start out, then, proposing to improve a deep learning neural network by using  $p$  values in the  $[1, 2]$  interval.

However, it is still interesting to be able to count on the robustness of the traditional squared Euclidean norm. Multiobjective optimization methods have been developed for the case where more than one goal is aimed at in the optimization process. For example, image enhancement aims to reduce the noise while preserving the small details. Usually, these two goals clash, since noise reduction is often achieved by smoothing out the image and thereby removing the details. Multiobjective optimization makes it possible to combine two or more loss functions and, depending on the methods and parameters used, to give more relevance to the most relevant goal in each case. A fundamental concept related to multiobjective optimization is that of the Pareto front [29], which is a surface in the space of possible solutions which comprises all solutions that are not dominated by any other solution, i.e. there is no solution which is equal or better than them for all the goals which are optimized.

Multiobjective optimization is based on the premise that the improvement of one of the objectives may lead to the deterioration of another objective. Therefore, a globally optimal solution is not possible, so that a search over the Pareto front is required. Evolutionary algorithms can be employed to approximate the Pareto front of a problem within a population of possible solutions, which has led to their extensive application [30]. Popular current proposals for multiobjective optimization include heuristic methods based on evolutionary algorithms [31]. Among these, genetic algorithms, decomposition-based proposals, particle swarm optimization, bat algorithms, harmony search, ant colony optimization, and non-dominated sorting genetic algorithms are designed to cope with various challenges of multiobjective optimization problems. All of them use operators inspired in biological evolution in order to improve a population of possible solutions to the optimization problem. Also, it is very common that several methodologies are hybridized, by combining

search and updating methods, or alternating methods in different phases [30]. In our case, the number of objectives to be optimized is two, although there are methods to optimize more than four objectives, which is known as many-objective optimization [32].

Scalarization is another approach to multiobjective optimization, which is based on the optimization of a scalar function which combines the multiple objective functions of the original problem. There are many scalarization methods in multiobjective optimization with different characteristics such as convexity, boundedness, the ability to generate proper efficient solutions, the number of additional constraints, etc. For example, the elastic constraint method [33] gives conditions on the characterization for properly efficient solutions, and the augmented weighted Chebyshev scalar problem [34] generates properly efficient solutions for certain selected values of weights and augmentation parameter. This work focuses on a particularization of the Pascoletti-Serafini scalarization [35], which guarantees to generate proper solutions just by selecting the proper weights.

In light of the above, our proposal involves combining two different cost functions with different values of  $p$ , which should lead to benefiting from the advantages of each of the two configurations. Therefore, the aim of this work is to obtain an improved deep learning network by using a multiobjective optimization approach with two  $L_p$ -norm-based loss functions and to find the parameters of the new cost function that yield the best results for their application to three-dimensional MR images. The combination of two alternative cost functions is a novel approach to the training of deep convolutional neural networks since the standard approach to deep learning relies on the minimization by stochastic gradient descent of a single loss function. While the standard approach has obtained remarkable results, our aim is to further improve the performance of the deep networks by making them pursue complementary goals which may help in solving the problem at hand, in our case the super-resolution of a three-dimensional MRI.

The rest of this paper is organized as follows: The theoretical background of this work is detailed in Section 2. Section 3 provides a description of the SR network to be improved, the datasets for testing and the optimization experiments, followed by the obtained results. A discussion of the obtained results is carried out in Section 4. Finally, the conclusions and proposals for future work are presented in Section 5.

## 2. Methodology

In this section, we propose a new optimization framework to train deep neural networks. Our proposal considers cost functions based on the  $L_p$ -norm [36,37] of the error vector. Depending on the specific norm, the optimization goal varies. Therefore, we employ multiobjective optimization techniques [29,38] in order to combine two cost functions. To do this, scalarization techniques are advocated, since the training process requires a single scalar quantity to be minimized. Subsection 2.1 deals with the  $L_p$ -norm cost functions, while Subsection 2.2 introduces multiobjective optimization of two cost functions by scalarization.

### 2.1. $L_p$ -norm loss functions

Next, the usage of  $L_p$ -norm loss functions for deep neural networks is investigated. The  $L_p$ -norm has previously been considered as a cost function for signal processing [39,40,41,42], image processing [43,44] and machine learning [45,46] tasks. The  $L_p$ -norm of a  $D$ -dimensional vector  $\mathbf{z} \in \mathbb{R}^D$  is given by:

$$\|\mathbf{z}\|_p = \sum_{j=1}^D |z_j|^p \quad (1)$$

The starting point for our strategy is the realization that the squared Euclidean norm loss function, i.e.  $p = 2$ , is not the only choice, and it might yield worse results than other loss functions associated to the  $L_p$ -norm of the error vector at a certain neural layer. Training samples with unusually high values of the components of the error vector will have a less dominating effect on the optimization if  $p < 2$ , which means that these values of  $p$  are likely to enhance the behavior of the learning rule whenever those extreme values of the error are present. This happens because the higher the value of the exponent  $p$ , the faster the  $L_p$ -norm grows for increasing values of its argument. For example, if  $p = 1$ , doubling the argument of the norm doubles the value of the  $L_p$ -norm, while if  $p = 2$ , doubling the argument of the norm quadruples the value of the  $L_p$ -norm. Consequently, the effect of outliers with unusually high values of the error, i.e. high values of the argument of the  $L_p$ -norm, is more exacerbated for high values of  $p$ .

The standard configuration of a deep learning neural network includes a loss function given by the average for all training data of the squared Euclidean norm

( $p = 2$ ) of the error vector computed from the desired output vector and the actual output vector yielded by the neural layer of interest. The introduction of the  $L_p$ -norm means that there is an extra tunable parameter of the learning algorithm, namely the exponent  $p$  in (1), which controls how much the learning is focused on reducing the error for the training data associated error vectors have components with large absolute values  $|z_j|$ . Lower values of  $p$  mean that less relevance is provided to the training data with the highest values of  $|z_j|$ . This is advantageous in those situations where the largest errors are associated with measurement errors, impulse noise, missing data, and similar factors. In this context, cost functions other than the squared Euclidean norm might provide better results, given their resilience to the kind of inconveniences mentioned before. In this investigation, the search for good performing exponents  $p$  is restricted to the interval of real numbers which yield a  $L_p$ -norm which agrees with the formal definition of a geometrical norm, i.e.  $p \in [1, 2]$ .

In light of the above, we advocate the use of  $L_p$ -norm loss functions for the neural layers of a deep neural network. The general definition of a  $L_p$ -norm loss function is given by:

$$E_p = \sum_{i=1}^N \sum_{j=1}^D |y_{ij} - v_{ij}|^p \quad (2)$$

where  $D$  is the dimension of the samples,  $N$  is the number of samples,  $y_{ij}$  is the  $j$ -th component of the  $i$ -th actual output vector and  $v_{ij}$  is the desired output vector.

The gradient of the  $L_p$ -norm loss (2) with respect to a neural weight  $w$  is:

$$\frac{\partial E_p}{\partial w} = \sum_{i=1}^N \sum_{j=1}^D p |y_{ij} - v_{ij}|^{p-1} \text{sign}(y_{ij} - v_{ij}) \frac{\partial y_{ij}}{\partial w} \quad (3)$$

where

$$\text{sign}(x) = \begin{cases} -1 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases} \quad (4)$$

## 2.2. Multiobjective optimization by scalarization

The optimization of the  $L_p$ -norm loss function (2) can lead to different configurations of a network de-

pending on the value of  $p$ . This suggests that a combination of two different loss functions  $E_p$  and  $E_q$ , with  $p \neq q$ , might merge the advantages of the optimal configurations associated with  $E_p$  and  $E_q$ . In other words, multiobjective optimization is proposed for the training of neural layers within a deep neural network. More than two loss functions might be combined, but we have not done it in this work.

As done in standard learning procedures, in our proposal the adjustment of the weights of a neural network is carried out by gradient descent. Since gradient descent minimizes a single loss function, this implies that a combined loss function  $S$  must be defined for each neural layer, which integrates the loss functions  $E_p$  and  $E_q$ . Our choice is driven by the excellent performance of the stochastic gradient descent optimization methods for deep neural networks which are readily available. In multiobjective optimization terms, this means that a scalarization must be performed [47], although metaheuristics methods could also be considered in future developments of our proposal [48]. Two scalarization strategies have been considered in this work:

- Weighted sum scalarization (WSS, [49]). In this simple approach, the combined loss function is defined as the weighted average of the two loss functions:

$$S_{WSS} = \lambda_1 E_p + \lambda_2 E_q \quad (5)$$

where  $\lambda_1, \lambda_2 \geq 0$ .

- Weighted Chebyshev scalarization (WCS, [50, 51]). This strategy is based on the previous calculation of an ideal point:

$$\mathbf{u} = (\min E_p, \min E_q) \quad (6)$$

where the minima are computed over the entire domain of the loss functions  $E_p$  and  $E_q$ . Then the combined loss function is defined as the weighted Chebyshev distance to the ideal point  $\mathbf{u}$ :

$$S_{WCS} = \max \{ \lambda_1 (E_p - u_1), \lambda_2 (E_q - u_2) \} \quad (7)$$

where again  $\lambda_1, \lambda_2 \geq 0$ . In our case, the ideal point is taken to be the null vector  $\mathbf{u} = (0, 0)$ , since the loss functions  $E_p$  and  $E_q$  are non negative.

Once the selected combined loss function  $S$  has been calculated, the gradient of the combined loss function with respect to a neural weight must be computed,  $\frac{\partial S}{\partial w}$ , in order to apply stochastic gradient descent for the update of the neural weight. Stochastic gradient descent variants and optimizations can be applied at this point.

### 3. Experiments

The content of this section reports the experiments that were carried out. The proposed methodology is applicable to any kind of regression network whose optimization layer is based on the minimization of a cost function that compares the input and the output of the network. Thus, two sets of experiments with two different convolutional neural networks were carried out, whose description is made in Subsections 3.1 and 3.2, including the description of the used datasets as well as the low-resolution image generation procedure. In addition to this, in Subsection 3.3 the third experiment for anisotropic generated data is presented in order to study the performance of the proposed methodology. The metrics employed to evaluate the performance of the proposal are detailed in Subsections 3.4 and the parameter tuning of the proposals is performed in Subsection 3.5. Subsection 3.6 details the statistic analysis we carried out. Finally, Subsection 3.7 sums up the outcomes of the experimental analysis.

Four different optimization models have been tested in each experiment: the standard squared Euclidean norm ( $p = 2$ ), which is the control algorithm, the best  $p$ -norm found in the parameter selection (Subsection 3.5), as well as the two proposed multiobjective scalarization methods, WSS and WCS. All the experiments were carried out on a 64-bit Personal Computer with a six-core Intel i7 3.50GHz CPU, 64 GB RAM, with a GPU Nvidia GTX Titan, with 12GB of dedicated memory. With the exception of the neural network execution, the low-resolution image generation and performance analysis were run on Matlab R2019a, using default parameters.

#### 3.1. Experiment 1: SRCNN3D

Firstly, we make use of the SRCNN3D deep neural network [8], which is a convolutional neural network that carries out the super-resolution of three-dimensional MR images.

SRCNN3D is based on the application of three blocks of convolutional layers successively, compris-

ing Rectified Linear Unit (ReLU) layers. The method first creates a pre-interpolated image  $I(\mathbf{X})$ , where  $\mathbf{X}$  is the input LR image. Then, the net computes a super-resolved HR image  $\hat{\mathbf{Y}}$  by the minimization of the squared Euclidean loss between the output of the CNN and the original HR image  $\mathbf{Y}$ .

$$f_{SqrEuc} = \arg \min_F \sum \|\mathbf{Y} - \hat{\mathbf{Y}}\|^2 \quad (8)$$

where  $F(I(\mathbf{X})) = \hat{\mathbf{Y}}$ .

This network is trained using overlapping patches extracted from a set of HR reference images. A down-sampling and up-sampling are applied to each patch and a set of pairs input-target is created to learn an end-to-end function between low and high-resolution images. Specific details of the implementation of this network can be found in the literature [8].

A scheme of the operation of the network is shown in Fig. 1. Our proposal<sup>1</sup> consists of the substitution of the squared Euclidean cost function  $f_{SqrEuc}$  by a cost function based on the  $L_p$ -norm as described in Section 2.

##### 3.1.1. OASIS dataset

In order to carry out an adequate analysis of the performance of each model, it was necessary to provide a large dataset of MR images. Nowadays, the number of public datasets has increased, although it is still hard to find the ideal images to be processed since they usually have images of both control and pathological subjects. For this experiment, we considered the OASIS-1 dataset, consisting of a cross-sectional MRI Data of 416 subjects aged 18 to 96. Data were acquired on a 1.5-T Vision scanner with a  $1.0 \times 1.0 \times 1.25 \text{ mm}^3$  voxel resolution over a FOV of  $256 \times 256 \text{ mm}$ .

A total of 220 T1-weighted MR images of the dataset were considered for the evaluation of the proposed models, which correspond to indices from 0001 until 0240 of type MR1 (patient's first visit), except the image 0080.

Since the whole dataset contains high-resolution images only, low-resolution images were created from the high-resolution ones and fed into the networks. As is stated in [52], the observation model is usually decomposed into a linear downsampling operator after a space-invariant blurring model as a Gaussian kernel with the full-width-at-half-maximum (FWHM) equal to slice thickness. SRCNN3D is based on this

<sup>1</sup>The source code and demo of the proposed approach will be published in case of acceptance.

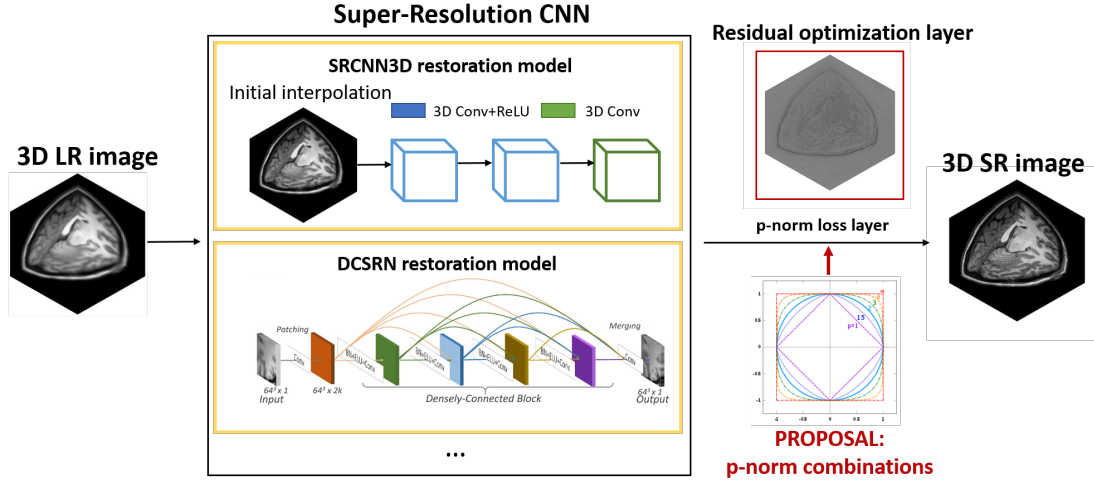


Fig. 1. Scheme of the proposed model: the LR image is fed into a convolutional neural network with a modified loss layer, producing an optimized SR image. These networks are based on the minimization of the residue between the original HR image and the output of the network.

model. For that purpose, the following procedure is applied. Firstly the HR images were adequately cropped to make the image dimensions divisible by the zoom factor. Then, a 3D Gaussian filter with a standard deviation equal to 1 is applied. Finally, *imresize3* Matlab function was used to perform a 3D cubic interpolation to obtain the LR image. This is a standard procedure to generate the LR versions of HR images for the evaluation of MRI super-resolution algorithms, as seen in [53,54,55].

### 3.1.2. Training procedure

The SRCNN3D has been developed using the Caffe package [56] on a Python framework. In this work, a training over 50000 iterations was carried out for each model as well as for the parameter selection. We considered this value to cover a large enough number of epochs that allows the network to converge properly and without taking too much time because each training takes around 12-14 hours to complete. The rest of the network hyper-parameters were set to default: momentum of 0.9, learning rate of 0.0001 and batch size of 256, using Stochastic Gradient Descent (SGD) for model optimization. In order to make the experiments replicable, we set the pseudorandom seed in the Caffe engine to the value 1701.

As described in Subsection 3.1.1, from a total of 220 images, the first 120 were used for training. Specifically, the first 100 images were used to train the network and the next 20 ones were used as a validation set to monitor the error curves. The remaining 100 images were used for testing. We divided each training, valida-

tion and testing sets into 10 folds with an equal number of images into them (10, 2 and 10 images, respectively) in order to carry out the statistical analysis described in Section 3.6. Although it may seem that the proportion training/testing is inadequate, the SRCNN3D model is patch-wise based and extracts around 15000 samples from each training image in order to have a sufficient number of inputs, while the testing is carried on the whole image without patch extraction.

Furthermore, taking advantage of that the SRCNN3D network can be trained for multiple scale factors at the same time, zoom factors 2, 3 and 4 were employed in our analysis. For it, the triple amount of patches was extracted from the training dataset and they were given to the network.

### 3.2. Experiment 2: DCSRN

In the second set of experiments, we make use of the 3D Densely Connected Super-Resolution Network (DCSRN) [57], which is focused on the super-resolution of three-dimensional MR images.

DCSRN is based on a densely-connected block. The network starts with a convolutional layer applied to the input image, and the output is fed to densely-connected block with 4 units, composed by a batch normalization layer and an exponential linear unit activation followed by a convolutional layer. In the end, a convolution is applied before providing the final SR image.

This network is patch-based so it is faster in training, back-propagation is more efficient, and the model is smaller. The patch size provided to the network is

$64 \times 64 \times 64$ . Another advantage is that there is less over-fitting during training. Specific details of the implementation of this network can be found in the literature. A scheme of the operation of the network is shown in Fig. 1.

### 3.2.1. HCP dataset

In this set of experiments, the Human Connectome Project (HCP) [58,59] was employed, consisting of a great amount of neuroimaging data ranging from structural MRI, functional MRI and diffusion tensor imaging (DTI), from multiple sites. Concretely, we used the HCP Young Adult 1200 Subjects Data Release, which includes 1113 structural MR scans acquired on a 3-T Siemens scanner. The image size is  $256 \times 320 \times 320$  with  $0.7 \times 0.7 \times 0.7 \text{ mm}^3$  voxel resolution. The first 600 T1-weighted MR raw images (no preprocessed) of the dataset were considered for the evaluation of the proposed models.

The low-resolution images were created in a different manner from the one used for SRCNN3D. A Fourier-based procedure is applied [60]: first, the Fast Fourier Transform (FFT) is computed on the HR images, then the resolution is degraded by zeroing the outer part of the 3D  $K$ -space, and finally, the inverse FFT is applied. The resulting LR image has the same size as the HR one but avoiding the generation of artifacts and it follows the real MR acquisition process. The raw images from HCP were treated as the ground-truth and we lowered the spatial resolution by a factor of 2 in each phase encoding direction to obtain the LR images, that are degraded like using  $zoom = 4$ .

### 3.2.2. Training procedure

The DCSRN has been developed using Tensorflow 1.8 on a Python 3.6 framework. The training was carried out with the default parameters set by their authors, over a total of 49000 iterations and a batch size of 2. This number of evaluations was empirically deduced monitoring the loss curves until they do not change. Adam optimizer with a learning rate of  $10^{-5}$  was used for model optimization.

From the 600 images of HCP1113 dataset, the first 500 were used for training and the remaining 100 images were used for testing. As well as described in Subsection 3.1.2, data were divided into training and testing sets into 10 folds with 50 and 10 images, respectively to carry out the statistical analysis described in 3.6. The DCSRN model is patch-wise based and 200 patches of  $64 \times 64 \times 64$  voxels are randomly extracted from each training image in order to have a sufficient number of inputs. For testing, the image was

first padded with zeros and then split into cubes of the mentioned size in a 3D sliding window manner using a stride of size 32. Therefore, the reconstructed patches obtained by the networks are merged by averaging the overlapped cubes.

### 3.3. Handling anisotropic data

There are many cases where isotropic low resolution structural brain MRI is uncommon. When such MR images are acquired, for example, a T2 or FLAIR modality, they typically retain a high in-plane resolution to provide sufficient quality data for radiologists to interpret across slices. Thus, the resolution is sacrificed in the through-plane direction and the voxel sizes become anisotropic.

In order to handle this type of image, the SRCNN3D network was used because it has the ability to restore an image in one plane without carrying out an extra training. A set of images of different databases were selected and low-resolution images were generated artificially by extending the voxel size in the last plane by a factor of 2, 3 and 4. Thus, if the image has voxel resolution  $1 \times 1 \times 1 \text{ mm}^3$ , for each image three LR versions were created of voxel size  $1 \times 1 \times 2 \text{ mm}^3$ ,  $1 \times 1 \times 3 \text{ mm}^3$  and  $1 \times 1 \times 4 \text{ mm}^3$ .

The images used in this experiment are four:

- Images 10 and 11 (named as MPRAGE10 and MPRAGE11 resp.) of the Kirby 21 dataset [61]. These data were acquired using a 3-T MR scanner with a  $1.0 \times 1.0 \times 1.2 \text{ mm}^3$  voxel resolution and size  $170 \times 256 \times 256$ .
- An image of the Medical Research Center of the University of Málaga (CIMES) acquired with size  $256 \times 256 \times 190$  and  $0.93 \times 0.93 \times 1.0 \text{ mm}^3$  voxel resolution (named as CIMES).
- An image of the IBSR dataset [62] with image size  $256 \times 256 \times 128$ , with  $1.5 \times 1.0 \times 1.0 \text{ mm}^3$  voxel resolution (named as IBSR).

### 3.4. Performance measures

In order to evaluate the performance of the proposed model three different quality measures were employed: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM) [63] and Bhattacharyya coefficient (BC) [64].

First of all, PSNR focuses on the intensity values obtained from the algorithm when it is compared with

the ground truth image. The unit of measurement is dB (decibel), where higher is better. It is defined as follows:

$$PSNR = 10 \log_{10} \left( \frac{peak^2}{\|\mathbf{Y} - \hat{\mathbf{Y}}\|^2} \right) \quad (9)$$

where *peak* is the maximum possible value of the image and  $\mathbf{Y}$ ,  $\hat{\mathbf{Y}}$  are the GT image and the predicted SR image, respectively. The PSNR accumulates the voxel errors provoked by incorrect estimation of the SR image.

On the other hand, SSIM focuses on structural similarities between images, returning a value between 0 and 1 (higher is better). This measure permits to check whether the edges are correctly preserved and it is formulated as:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \quad (10)$$

where  $\mu_x$  and  $\mu_y$  are the mean value of images  $x$  and  $y$ ,  $\sigma_x$  and  $\sigma_y$  are the standard deviations of images  $x$  and  $y$ ,  $\sigma_{xy}$  is the covariance of  $x$  and  $y$ ,  $c_1 = (k_1L)^2$  and  $c_2 = (k_2L)^2$  (default values were used:  $L = 1$  is the dynamic range,  $k_1 = 0.01$  and  $k_2 = 0.03$ ).

Finally, the BC measures the closeness of the two discrete pixel probability distributions  $P$  and  $\hat{P}$  corresponding to the ground truth (GT) and restored images with values in the range  $[0, 255]$ :

$$BC = \sum_{j=0}^{255} P(j)\hat{P}(j) \quad (11)$$

where  $BC \in [0, 1]$  and higher is better.

From a qualitative point of view, it is useful to analyze the residual images obtained by the subtraction of the GT image  $Y$  and the super-resolved one  $\hat{Y}$ :

$$ResI = |Y - \hat{Y}| \quad (12)$$

The best performance is such that the residual image is the zero matrix. The constant 0.5 was added to the residual images for the sake of clarity, thereby obtaining gray images.

### 3.5. Parameter selection

Our main aim was to attain the best generality at the time of tuning up the parameters of the proposals, so a set of images different from the ones used for

training and testing were used. In the case of the SR-CNN3D network three different images from 3 different datasets were used to fine-tune the model parameters of each cost function:

- Image 0080 of the OASIS-1 dataset.
- Image 01 of the Kirby 21 dataset [61].
- A normal brain T1 image of Brainweb<sup>2</sup> simulated database, acquired with slice thickness 1 mm, 0 % of noise level and  $RF = 0$ .

Regarding the DCSRN network, the MGH HCP Adult Diffusion dataset was used, which comprises 35 young adult structural scans using the MGH Siemens 3T Connectome scanner.

PSNR, SSIM and BC measures of the above-presented images were computed and a ranking was established sorting each tested parameter according to its performance with respect to each image. The assigned points were accumulated among all images and the lowest scores mean that the network is better. Thus, there are two kinds of parameters to be tuned: the  $L_p$ -norms and the weights.

Firstly, we performed a set of experiments fixing weights and varying the  $L_p$ -norms. For the SR-CNN3D, there were many cases where the convergence of the network was not achieved and the best performances were always obtained with  $p = 2.0$  and  $q = 1.9$  norms, which combines respectively the stability of the gold standard cost function and the best  $L_p$ -norm obtained in our previous work [65]. An example of training loss curves for the WSS model is shown in Fig. 2. In the case of DCSRN, we found a better performance of the  $p = 2.0$  and  $q = 1.7$  norms.

Secondly, for both WSS and WCS cost functions a set of weight values  $\lambda_i$  were tested in order to find the best configuration. Both WSS and WCS methodologies depends on two variable weights  $\lambda_1, \lambda_2$ . As the training takes a long time to be completed we aimed to simplify the parameter optimization by making some assumptions, which are detailed next. In the case of the weighted sum scalarization method (WSS), we are dealing with a linear combination of two  $L_p$ -norms, so we assumed that  $\lambda_1 = 1 - \lambda_2$ , thereby removing one variable. In Fig. 3 we can find the tested values for  $\lambda_2$ , from 0.2 until 0.8. We found that the best PSNR and SSIM ranks are achieved for middle-low values of  $\lambda_2$  (0.25-0.35), although values around 0.5 are also fine for BC measure.

<sup>2</sup><http://mouldy.bic.mni.mcgill.ca/brainweb/>

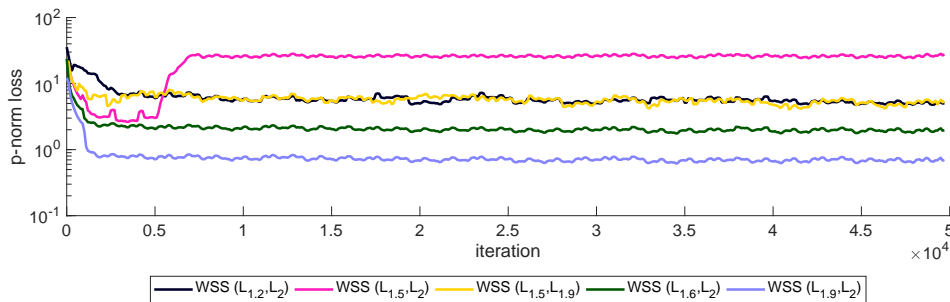


Fig. 2. Training loss curves of the WSS model optimization (logarithmic scale) using SRCNN3D network.

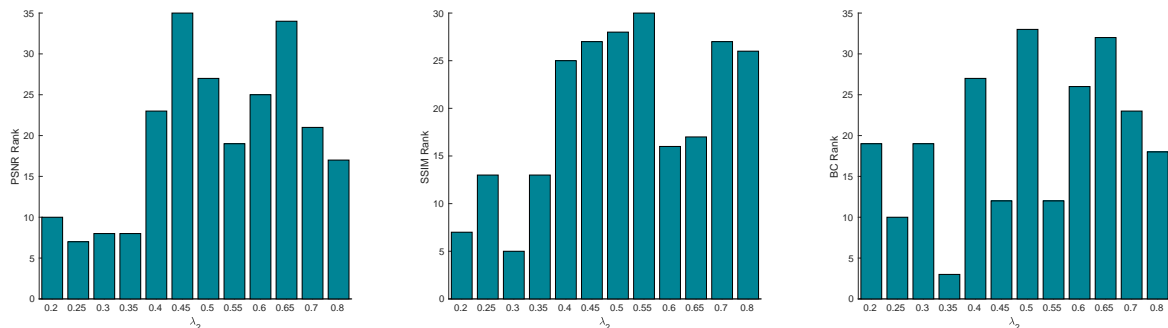


Fig. 3. WSS model optimization for SRCNN3D: PSNR, SSIM, and BC rankings across the three tuning images are shown varying  $\lambda_2$ , with  $\lambda_1 = 1 - \lambda_2$ .

The weighted Chebyshev scalarization (WCS) is based on the maximum value between the two  $L_p$ -norms. Thus, a similar scale between the  $L_p$ -norms cost functions is essential to avoid the scalarization to be dominated always only by one of the cost functions. Assuming that  $\lambda_1 = 1$ , we only need to fix  $\lambda_2$  to balance both norms. Fig. 4 collects the best range of values, showing us that the best ranks are reached between 0.7 and 0.8 depending on the considered quality measure.

In order to make a reasonable selection of the best  $\lambda_2$ , the mean and standard deviation among the three measure ranks were computed, as shown in Fig. 5. That is, the rank values depicted in Fig. 3 (WSS model) are averaged for each value of the parameter, and the same was done with the results of Fig. 4 (WCS model). The best parameter is marked in magenta, considering that the lowest mean rank is the best option to achieve good performance for all the quality measures.

In Table 1 the final configurations of the models based on our previous analysis are summarized. With respect to the DCSRN network, the configurations of the WSS and WCS methods were established taking

into account the previous analysis. Thus, the WSS parameters were set up with the best two  $p$ -norms found and with the same weights. The WCS optimization also used the  $p = 1.7$  norm but modifying the weight  $\lambda_2$  in order to put the norm on the same scale as the squared Euclidean norm.

### 3.6. Significance analysis

First, a Friedman aligned ranks test [66,67] is performed in order to check whether at least two of the methods represent populations with different median values, i.e. the methods have significantly different performance. This technique is a similar version of the Friedman test that can be used under the same circumstances, although the Friedman aligned ranks test is appropriate where the number of methods to be compared is low.

In this technique, if we have  $NK$  test images, we can do  $K$  runs with  $N$  test images per run, so that the images of each run are different. Then we compute the mean of the performance measure for the  $N$  test images of each run. This way we get  $K$  mean perfor-

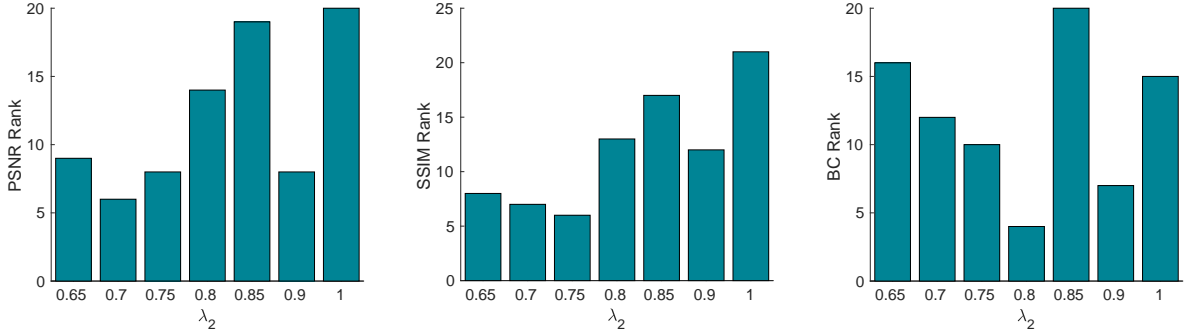


Fig. 4. WCS model optimization for SRCNN3D: PSNR, SSIM, and BC rankings across the three tuning images are shown varying  $\lambda_2$ , with  $\lambda_1 = 1$ .

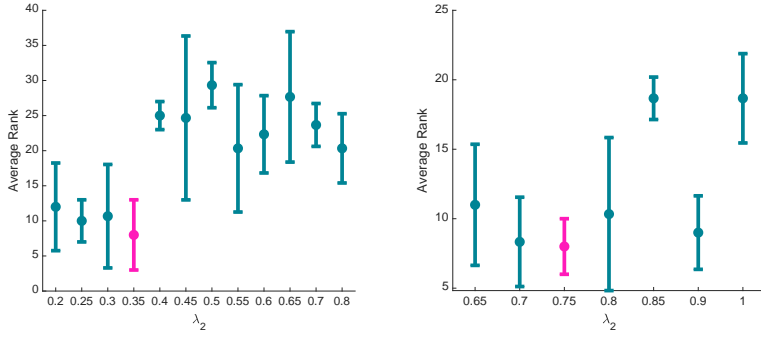


Fig. 5. Mean and standard deviation of the rank values computed among the PNSR, SSIM and BC ranks for both models: WSS, with  $\lambda_1 = 1 - \lambda_2$  (left) and WCS, with  $\lambda_1 = 1$  (right).

Table 1  
Parameter selection of the proposed cost functions.

| Model | Parameters   |  |
|-------|--|--|
|       | SRCNN3D  | DCSRN  |
| WSS   | $p = 2.0, q = 1.9, \lambda_1 = 0.65, \lambda_2 = 0.35$ | $p = 2.0, q = 1.7, \lambda_1 = 0.65, \lambda_2 = 0.35$ |
| WCS   | $p = 2.0, q = 1.9, \lambda_1 = 1.00, \lambda_2 = 0.75$ | $p = 2.0, q = 1.7, \lambda_1 = 1.00, \lambda_2 = 0.45$ |

mance values for each method and performance measure. Then, we calculate the difference between the performance obtained by a method and the mean performance value computed over all methods. This step is repeated for all methods and datasets. The resulting differences are then ranked from 1 to 4 (since there are four methods), where the best performing method is assigned 1 point and the worst performing method is assigned 4 points. After that, the  $K$  ranks associated with each method are accumulated, so that each method obtains an accumulated rank between  $K$  and

$4K$ . Finally, the test statistic is computed:

$$z = \frac{(R_i - R_j)}{\sqrt{k(n+1)/6}} \quad (13)$$

where  $R_i, R_j$  are the average rankings of the compared methods.

Then, if the obtained p-value is smaller than the level of significance  $\alpha = 0.05$ , then the null hypothesis is rejected, i.e. there are at least two methods with significantly different performance. In our work,  $N = 10$

Table 2

Friedman Aligned Rankings of the methods for PSNR measure and for zoom factors 2, 3 and 4, computed for the SRCNN3D network. The last row shows the probability value to reject the null hypothesis.

| PSNR           | Ranking  |          |          |
|----------------|----------|----------|----------|
|                | zoom = 2 | zoom = 3 | zoom = 4 |
| <b>SRCNN3D</b> |          |          |          |
| WCS            | 23.4000  | 20.0999  | 20.5999  |
| WSS            | 20.1000  | 23.3000  | 22.3000  |
| $p = 1.9$      | 14.9000  | 17.5000  | 14.5999  |
| $p = 2$        | 23.5999  | 21.0999  | 24.5000  |
| <b>P-value</b> | 0.0429   | 0.0414   | 0.0422   |

and  $K = 10$ . We used the original software published by the authors<sup>3</sup>.

Additionally, box plots were generated, one box plot per performance measure, where each method is associated to a box, and the  $N$  mean values of the performance measure for a particular method are used to generate the box, i.e. each of the boxes of the box plot is drawn from  $N$  samples.

### 3.7. Results

First, we evaluate the proposed methodology from a quantitative point of view for each experiment. For each quality measure and each zoom factor, a Friedman aligned ranks test was carried out to measure between the different cost functions tested. A total of 10 mean values corresponding to the 10 test repetitions were computed for each method and passed to the Friedman aligned ranks test. The methods were ranked assigning 1,2,3 or 4 points for each repetition and the accumulated results are the ones presented in the following tables.

#### 3.7.1. Results of Experiment 1

PSNR aligned Friedman ranks are shown in Table 2. The lower values for all zoom factors, i.e the best, are achieved always by the model based on a unique  $L_p$ -norm with  $p = 1.9$ . Moreover, the p-value is lower than 0.05 for all cases, which means that the methods are not the same.

There is a bit of variety for SSIM and BC measures, whose statistical analysis is summarized in Tables 3 and 4. For larger scale factors (3 and 4), where the network needs to be more precise to recover the voxel's information, the weighted Chebyshev scalarization is clearly the best cost function, achieving the lowest ranks of all the Friedman analysis performed.

Table 3

Friedman Aligned Rankings of the methods for SSIM measure and for zoom factors 2, 3 and 4, computed for the SRCNN3D network. The last row shows the probability value to reject the null hypothesis.

| SSIM           | Ranking  |          |          |
|----------------|----------|----------|----------|
|                | zoom = 2 | zoom = 3 | zoom = 4 |
| <b>SRCNN3D</b> |          |          |          |
| WCS            | 16.9999  | 13.1000  | 11.5000  |
| WSS            | 23.2000  | 23.3000  | 23.7000  |
| $p = 1.9$      | 14.9000  | 20.1000  | 25.6000  |
| $p = 2$        | 26.9000  | 25.5000  | 21.2000  |
| <b>P-value</b> | 0.0065   | 0.0427   | 0.0440   |

Table 4

Friedman Aligned Rankings of the methods for BC measure and for zoom factors 2, 3 and 4, computed for the SRCNN3D network. The last row shows the probability value to reject the null hypothesis.

| BC             | Ranking  |          |          |
|----------------|----------|----------|----------|
|                | zoom = 2 | zoom = 3 | zoom = 4 |
| <b>SRCNN3D</b> |          |          |          |
| WCS            | 20.4000  | 19.8000  | 19.5999  |
| WSS            | 20.3000  | 20.6000  | 21.1000  |
| $p = 1.9$      | 21.1000  | 21.4000  | 21.7000  |
| $p = 2$        | 20.2000  | 20.2000  | 19.5999  |
| <b>P-value</b> | 0.0412   | 0.0415   | 0.0418   |

This means that the WCS is more suitable for recovering the structural features of the MR image than both the usual squared Euclidean norm and the  $L_{1.9}$ -norm. As shown in Table 3, for scale factor 2 WCS is the second best method and the p-value is significantly lower than 0.05, only surpassed by  $p = 1.9$ , making this optimization model effective for any case.

The differences in terms of BC are closer. In Table 4 all the average rankings have values around 20, although again for zoom factors 3 and 4 the WCS method has the best outcome, corroborated by the p-value. However, we can see that the squared Euclidean norm is the second best cost function. As the BC measures the differences in pixel probability distributions, we can infer that the image histograms obtained by all the tested methods are very similar, thereby avoiding the inclusion of artifacts or anomalous intensity values.

The variances between the 10 runs of 10 different images we executed are depicted as box plots in Figure 6, for scale factors 2, 3 and 4. Analyzing first the zoom 2 (first row), in terms of PSNR and SSIM the  $L_p$ -norm with  $p = 1.9$  the performance is remarkable. The mean and median values (represented as a circle and a line, resp.) overcome the other three cost functions and the variance of the results is quite reduced. Thus, the image intensity values are always very close to the original HR image. There are no meaningful differences

<sup>3</sup><https://sci2s.ugr.es/sicidm>

between the other methods, although the BC box plots show higher differences between the mean and the median except for  $p = 1.9$ . Again, this may indicate that there is an anomalous test image that is better super-resolved on average by  $p = 1.9$ . However, in median, the best method appears to be WCS. The case of scale factor 3 (second row) is quite similar to the previous one. However, the performance of WCS has improved a bit and the one of WSS has worsened. Looking at the median values,  $p = 1.9$  and WCS achieved the best measures over the 10 runs for PSNR and SSIM. When the zoom demand is set to 4 (third row), which is very high, the variance between runs and between methods is reduced. However, the WCS method takes advantage of both the  $L_{1.9}$  and  $L_2$ -norms to improve the quality of the image in terms of either PSNR, SSIM, and BC. The latest measure has reduced considerably its interquartile range compared to the other methods.

### 3.7.2. Results of Experiment 2

Figure 7 summarizes the outcomes obtained by the DCSR network after modifying its cost function. As explained in Subsection 3.2.1, this network was trained by simulating degradation by Fourier transforms similar to zoom factor 4. The most stable method with less dispersion in its results is WSS, which yields better SSIM and BC measures than the squared Euclidean norm. The mean value (circle) of PSNR is on par with the  $L_2$ -norm, although in general, the results are not as good as expected. Nevertheless, the results generated by WCS model are higher for PSNR and also quite acceptable for the rest of the measures. The  $L_{1.7}$ -norm has the lowest dispersion but the results are not the best.

The statistical analysis carried out to check if the methods are significantly different is presented in Table 5. The average rankings computed by the Aligned Friedman test showed that WCS is the first method for PSNR followed by WSS model, and this one is the best for SSIM and BC. It should be remarked the difference obtained with respect to the  $L_2$ -norm, which is the last method in the tests. The  $p$ -value lower than 0.5 corroborates that the methods are significantly different.

The WSS and WCS results of this set of experiments confirm the idea of improving the performance by the combination of two  $p$ -norms. Both cost functions are appropriate for the degradation model based on Fourier transform, which simulates the acquisition of MR images with lower resolution.

Table 5

Friedman Aligned Rankings of the methods for PSNR, SSIM, and BC measures computed for the DCSR network. Last row shows the probability value to reject the null hypothesis.

| Model          | Ranking |         |         |
|----------------|---------|---------|---------|
|                | PSNR    | SSIM    | BC      |
| DCSRN          |         |         |         |
| WCS            | 18.9999 | 20.5000 | 22.2000 |
| WSS            | 20.2000 | 19.0000 | 16.4000 |
| $p = 1.7$      | 21.5000 | 20.0999 | 20.4000 |
| $p = 2$        | 21.2999 | 22.4000 | 23.0000 |
| <b>P-value</b> | 0.0393  | 0.0396  | 0.0394  |

### 3.7.3. Anisotropic Super-Resolution

The last set of experiments deals with anisotropic images. The quantitative outcomes of the zoomed images are collected in Table 6. The values of the scale factors are referred to the SR applied to the third dimension only. The rows show the results obtained for each of the four tested images and the columns represent the measures obtained for each scale. In blue are highlighted the best values for each measure and zoom factor.

The best optimization method is shared by the  $L_{1.9}$ -norm and WSS. From one side, WSS performs better with MPRAGE10 and IBSR images improving the PSNR and SSIM in some tenths. MPRAGE11 and CIMES have been restored better by the single  $p$ -norm, although the second-best method is again WSS. This fact indicates that the weighted linear combination of  $p$ -norms can be a great option to also improve anisotropic images. Regarding the BC measure, either  $p = 1.9$ ,  $p = 2$  or WCS reach good values.

On the other hand, the differences between method performances increase when the scale factor applied in the SR process is higher, due to the necessity of recovering more information from the one present in the image. This occurs for every method if we compare them with the squared Euclidean norm, obtaining around 1% of improvement in the quality of the image. We need to remark that the network was only trained for isotropic tasks, so it is easy to think that appropriate training focused only on this type of image may improve substantially the outcomes.

### 3.7.4. Qualitative performance

In terms of qualitative outcomes, three different images of the OASIS dataset are presented. Fig. 8 shows a three-dimensional perspective with one slice of the sagittal, coronal and axial planes of image numbered as 0174, using an augmentation factor of 2. The differences can be seen in the central part of the image,

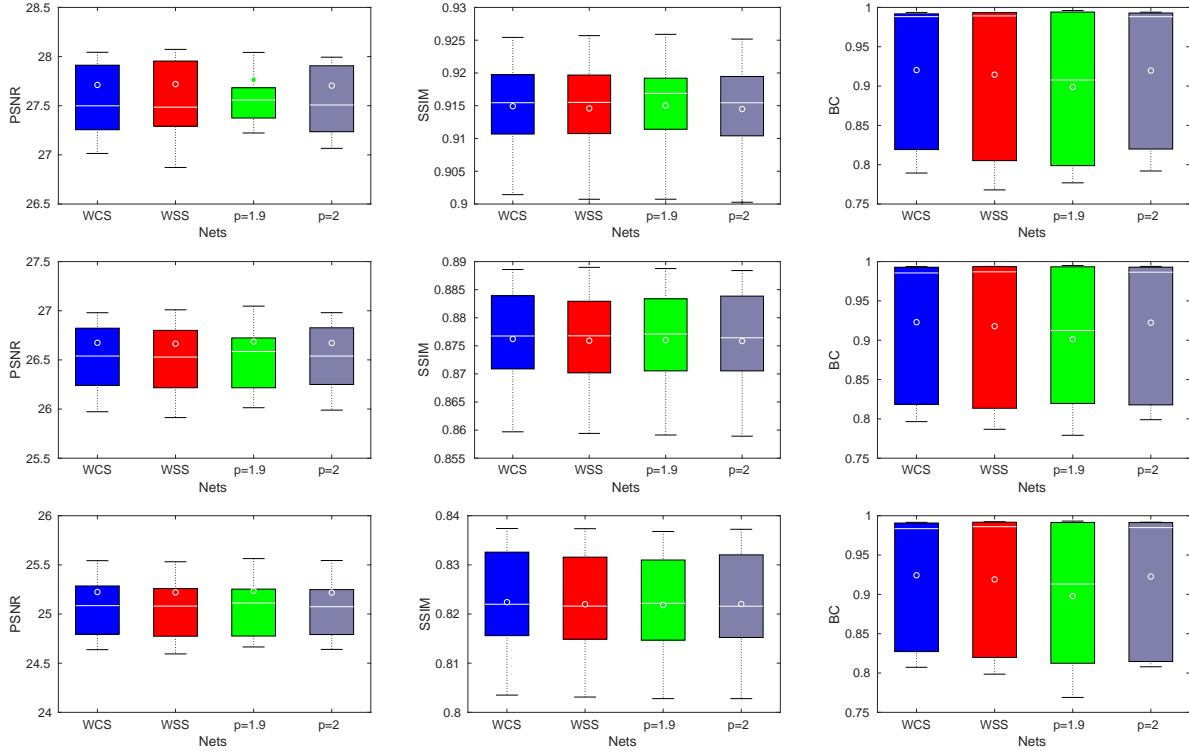


Fig. 6. Comparison of the PSNR, SSIM and BC for the four models using SRCNN3D network and  $zoom = 2$ ,  $zoom = 3$  and  $zoom = 4$  (from top row to bottom). Box plots of 10 runs are displayed, where the medians are plotted as horizontal gray lines, while the means are plotted as gray circles.

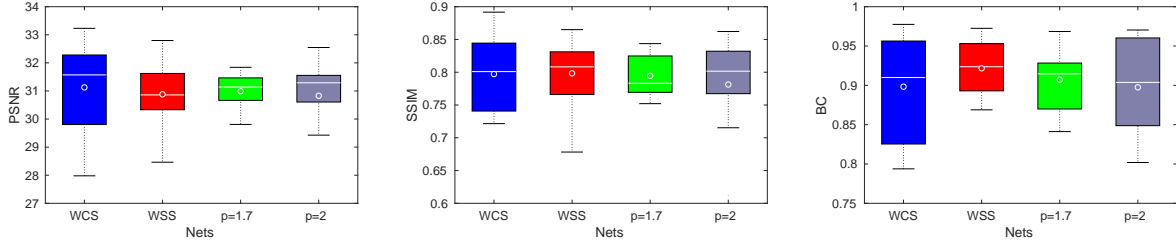


Fig. 7. Comparison of the PSNR, SSIM, and BC for the four models using DCSR network. Box plots of 10 runs are displayed, where the medians are plotted as horizontal gray lines, while the means are plotted as gray circles.

where the intensity of gray varies from one method to another. If we focus on the residual images, the whitest image should be the best approximation to the original HR image. Here, the method based on the  $L_{1.9}$ -norm clearly outperforms the other methods since there are fewer dark curves in the central part of the image. This means that fewer structures were removed. This result matches the previous quantitative analysis, where its values of PSNR and SSIM are the best.

In Fig. 9 a section of the coronal plane of the OASIS-0177 image is shown. In this case, the restora-

tion factor was 3. Here the effect of the  $p = 1.9$  based method is the opposite in terms of similarity structure. The darkest parts of the image are refined while in the original ground-truth image there some gray level irregularities, removing possible abnormalities of the scan or the brain structure. On the other hand, the squared Euclidean norm over-smooths the output of the network with respect to the ideal image. Both WSS and WCS techniques find an equilibrium between both cost functions, as we can see in the middle-left part of

Table 6

PSNR, SSIM and BC measures computed for with SRCNN3D network varying the scale factor of the third dimension of the anisotropic LR image. The blue color represents the best method.

| Image    | Zoom      | zoom = 2 |        |        | zoom = 3 |        |        | zoom = 4 |        |        |
|----------|-----------|----------|--------|--------|----------|--------|--------|----------|--------|--------|
|          |           | Model    | PSNR   | SSIM   | BC       | PSNR   | SSIM   | BC       | PSNR   | SSIM   |
| MPRAGE10 | WCS       | 33.4432  | 0.9580 | 0.9957 | 31.9899  | 0.9426 | 0.9955 | 30.8662  | 0.9266 | 0.9954 |
|          | WSS       | 33.4923  | 0.9606 | 0.9742 | 32.0180  | 0.9452 | 0.9749 | 30.8850  | 0.9292 | 0.9753 |
|          | $p = 1.9$ | 33.4870  | 0.9605 | 0.9966 | 31.9956  | 0.9448 | 0.9964 | 30.8595  | 0.9287 | 0.9963 |
|          | $p = 2$   | 33.4587  | 0.9579 | 0.9960 | 32.0001  | 0.9424 | 0.9958 | 30.8737  | 0.9263 | 0.9957 |
| MPRAGE11 | WCS       | 35.9565  | 0.9641 | 0.9967 | 34.5699  | 0.9507 | 0.9963 | 33.3593  | 0.9350 | 0.9961 |
|          | WSS       | 36.0155  | 0.9655 | 0.9579 | 34.5953  | 0.9518 | 0.9583 | 33.3875  | 0.9362 | 0.9579 |
|          | $p = 1.9$ | 36.0548  | 0.9655 | 0.9971 | 34.6155  | 0.9516 | 0.9968 | 33.3990  | 0.9360 | 0.9966 |
|          | $p = 2$   | 35.9883  | 0.9641 | 0.9968 | 34.5899  | 0.9507 | 0.9964 | 33.3713  | 0.9349 | 0.9962 |
| IBSR     | WCS       | 33.4739  | 0.9730 | 0.9269 | 31.0103  | 0.9560 | 0.9254 | 29.3085  | 0.9367 | 0.9268 |
|          | WSS       | 33.5867  | 0.9733 | 0.9293 | 31.0563  | 0.9561 | 0.9274 | 29.3414  | 0.9368 | 0.9283 |
|          | $p = 1.9$ | 33.4919  | 0.9717 | 0.9273 | 31.0001  | 0.9543 | 0.9255 | 29.3022  | 0.9349 | 0.9272 |
|          | $p = 2$   | 33.5348  | 0.9731 | 0.9270 | 31.0455  | 0.9561 | 0.9253 | 29.3251  | 0.9368 | 0.9268 |
| CIMES    | WCS       | 35.5276  | 0.9568 | 0.9975 | 33.4533  | 0.9306 | 0.9971 | 32.0381  | 0.9026 | 0.9960 |
|          | WSS       | 35.5835  | 0.9579 | 0.9636 | 33.4834  | 0.9313 | 0.9633 | 32.0693  | 0.9032 | 0.9610 |
|          | $p = 1.9$ | 35.6078  | 0.9581 | 0.9980 | 33.4932  | 0.9315 | 0.9977 | 32.0748  | 0.9036 | 0.9969 |
|          | $p = 2$   | 35.5522  | 0.9568 | 0.9975 | 33.4691  | 0.9306 | 0.9970 | 32.0455  | 0.9025 | 0.9960 |

the residual images, where the gray is more homogeneous.

Finally, the visual outcome of DCSRNN network is analyzed in Fig. 10. Here a section of the test image with ID 206929 is displayed. The amount of information to be recovered is quite high, provoking distortions in the enhanced image. Focusing on the residual images we can see better the different performance of the optimization methods. WCS method tends to resolve better the intensities because large gray surfaces are clearer than the other methods. On the other hand, fine details are remarked by WSS and  $p = 2$  since connections between those homogeneous parts are darker, although they introduce some noise in the image. The  $L_{1.7}$ -norm does not achieve to restore the image adequately. This alternation of performance is present along all the volume, making the quality measures in concordance with the box plot analysis.

#### 4. Discussion

The above-presented experiments have demonstrated that the multiobjective optimization of the cost function makes the SR networks more precise. Two different ways to combine the  $L_p$ -norm are proposed, WSS and WCS, although its performance varies depending on the dataset and neural network used.

The WCS performed well with SRCNN3D, the SSIM and BC values improved with respect to the squared Euclidean norm. Nevertheless, the WSS method worked better than WCS with the DCSRNN network. In this case, WSS yielded good outcomes for either PSNR, SSIM, and BC, but also WCS was good restoring the images. There are two factors that may affect the results of the experiments: the type of the neural network and the data used for training.

Firstly, the effect of the backpropagation in the learning procedure may be crucial in the performance of the methods. With a small network like SRCNN3D (only three convolutional layers), the Chebyshev scalarization achieved more stability. However, when a larger, densely connected network is trained, this methodology loses efficiency and the weighted sum of norms outperforms the rest of the models. Thus, the different layers of the network learned better the features of the images because they were interconnected. This fact may indicate that the larger the network used, the more efficient might be the combination of  $p$ -norms.

Secondly, the amount of data and patch sizes were different in each case. For DCSRNN, a larger patch is used, which covers more details of the image, and the minimization of the errors is more effective if the cost function is more complex. The WCS is essentially one  $p$ -norm that varies depending on the maximum value

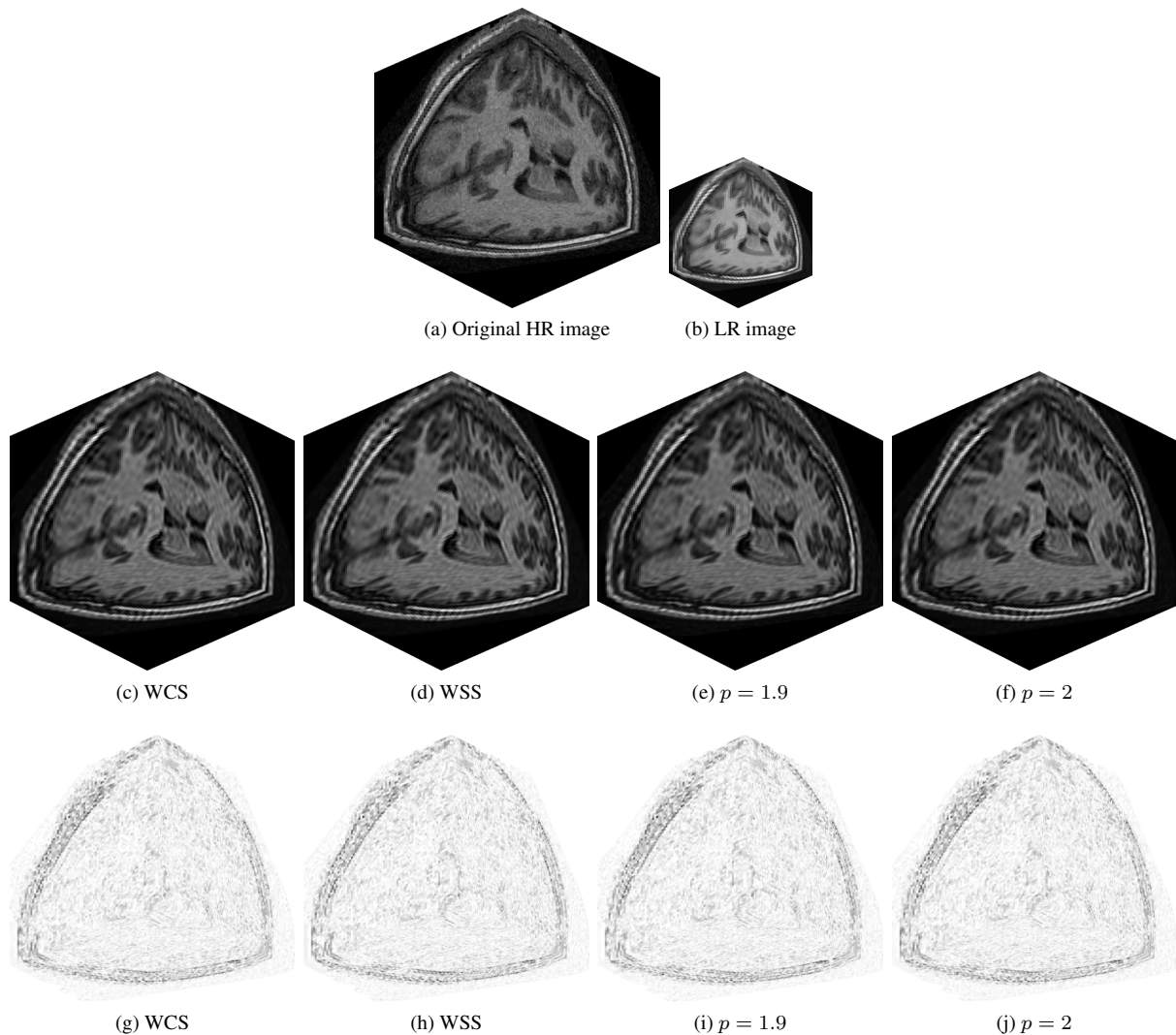


Fig. 8. Qualitative results for OASIS-0174 image for each model, applied with zoom factor 2. Slices of sagittal, coronal and axial views are showed in a 3D representation. The second row shows the reconstructed image by each algorithm and the third row shows residual images between the reconstructed and the original HR image.

reached. Thus, when less information is present, that is, smaller patches as the ones used in SRCNN3D, one norm is enough to minimize the error.

The qualitative outcomes showed that the scalarization methods can provide refined results. The differences among methods were more noticeable in the DC-SRN network than in SRCNN3D. The reason might be the degradation model on which they are based. SCRNN3D carries out an initial interpolation that can smooth the effect of the restoration, while DCSRNN is created to enhance LR images based on a low-pass filtering-like, i.e. the number of voxels are the same and there is not an intermediate blurring effect.

## 5. Conclusions and future works

This work presents a multiobjective optimization model for deep super-resolution neural networks. With the aim of improving the brain magnetic resonance image super-resolution, the usual squared Euclidean loss layer is substituted by combinations of  $L_p$ -norm cost functions using the weighted sum and the weighted Chebyshev scalarizations. The optimization function is defined with  $p < 2$  to reduce the effect error of extreme values of the errors and enhance the behavior of the training.

SRCNN3D and DCSRNN models, and OASIS and HCP datasets were employed for experiments. Three

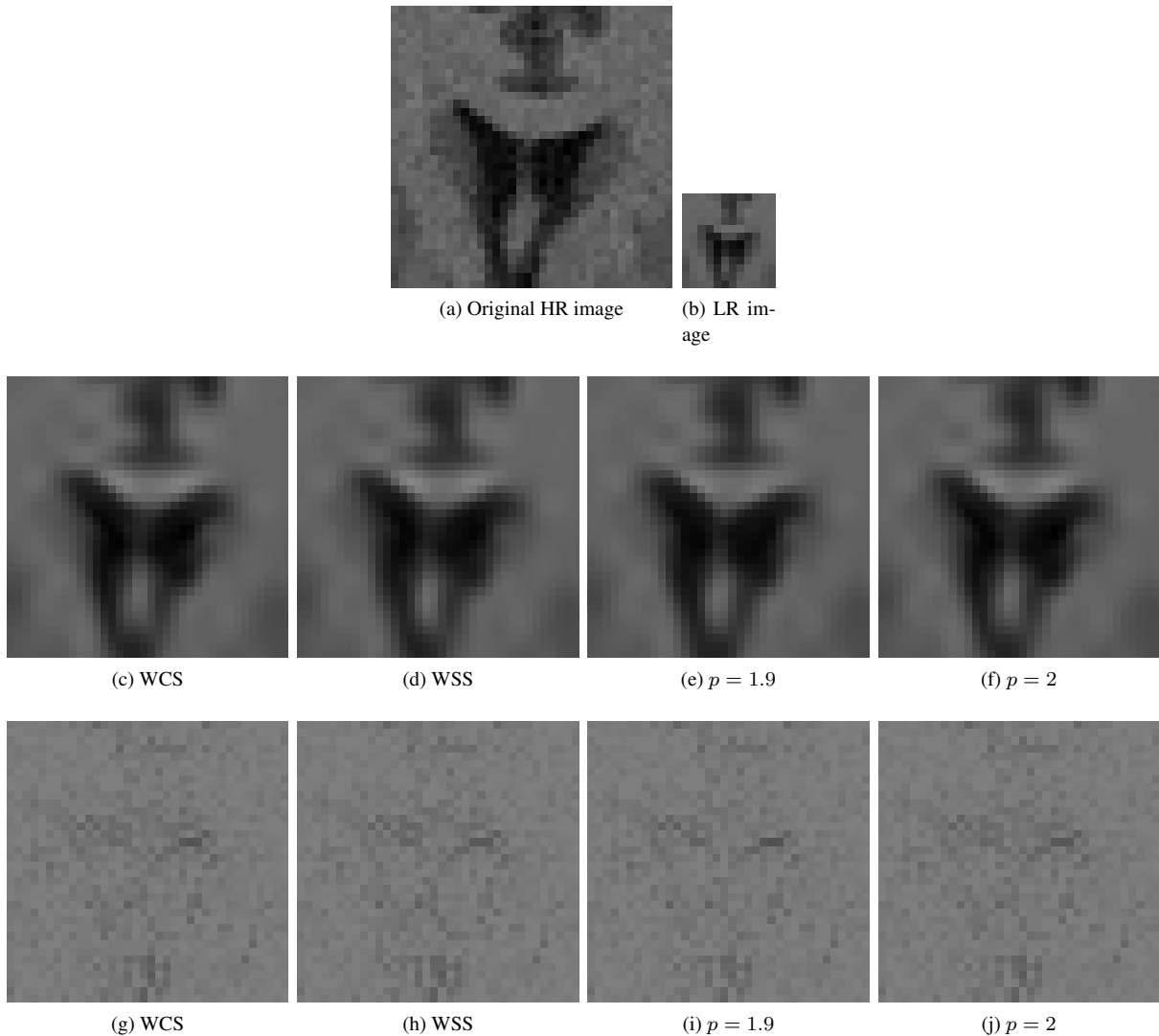


Fig. 9. Qualitative results for a section of the coronal view of the OASIS-0177 image for each model, applied with zoom factor 3. The second row shows the reconstructed image by each algorithm and the third row shows residual images between the reconstructed and the original HR image.

different models were compared to the  $L_2$ -norm and results show that the squared Euclidean loss layer is not the best norm in most cases. When we duplicate the resolution, the  $L_{1.9}$ -norm evince good results in all measures and if large scale factors are used, WCS and WSS methods outperform the SSIM and BC values. A Friedman aligned ranks test indicates that the differences with respect to the gold standard are significant at 95%. Qualitatively, image restoration by WCS better preserves structural information.

In future work, we will extend and apply the proposed idea to other machine learning tasks, such as recommendation task and person tracking [68]. The

proposed approach could be extended to other neural networks in order to improve the quality of the outputs in any other task like noise removal or segmentation. Moreover, the depth of the network seems to be the key to a correct back-propagation of the proposed cost function errors. An extensive analysis with deeper neural networks may improve the performance with lower values of  $p$ . Another line of research is the usage of three or more cost functions to be optimized rather than two. Adding more  $p$ -norms might improve the quality of the super-resolved images, but the selection of the  $p$ -norms to be included in the set of cost functions to be optimized is a difficult optimization

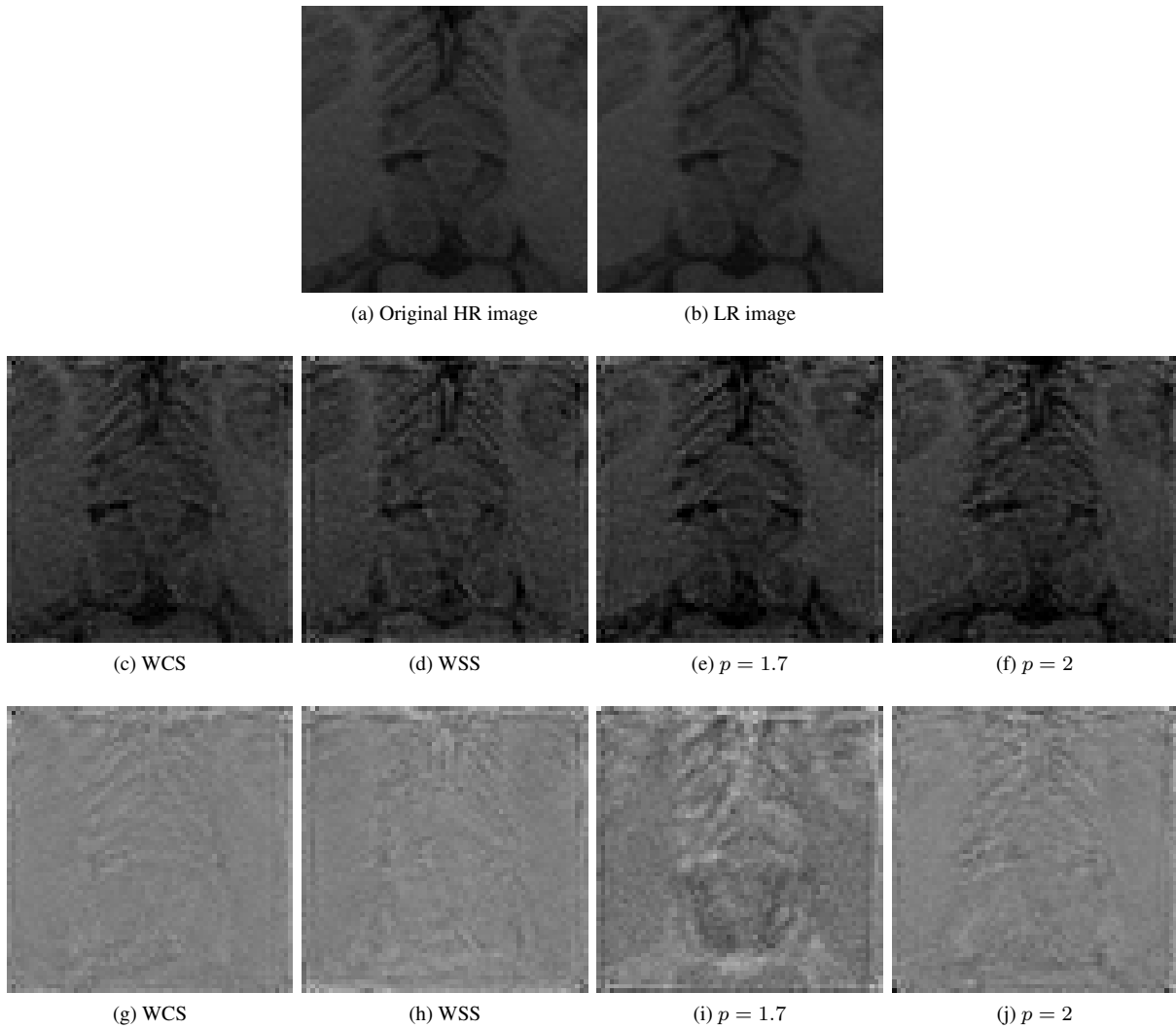


Fig. 10. Qualitative results for a section of a reconstructed patch of the image ID 206929 from HCP dataset for each model, applied with zoom factor 4. The second row shows the reconstructed image by each algorithm and the third row shows residual images between the reconstructed and the original HR image.

problem in itself. This is why we have restricted our attention in this work to the two cost functions case, while the other cases are left for future extensions of our approach.

### Acknowledgments

This work is partially supported by the Ministry of Economy and Competitiveness of Spain under grants TIN2016-75097-P and PPIT.UMA.B1.2017. It is also partially supported by the Ministry of Science, Innovation and Universities of Spain (grant number RTI2018-094645-B-I00), project name Automated de-

tection with low cost hardware of unusual activities in video sequences. It is also partially supported by the Autonomous Government of Andalusia (Spain) under grant MA18-FEDERJA-084, project name Detection of anomalous behavior agents by deep learning in low cost video surveillance intelligent systems. All of them include funds from the European Regional Development Fund (ERDF). The authors thankfully acknowledge the computer resources, technical expertise and assistance provided by the SCBI (Supercomputing and Bioinformatics) center of the University of Málaga. They have also been supported by the Biomedical Research Institute of Málaga (IBIMA). They also gratefully acknowledge the support of NVIDIA

Corporation with the donation of two Titan X GPUs used for this research. The authors also thankfully acknowledge the grant of the Universidad de Málaga. Karl Thurnhofer-Hemsi (FPU15/06512) is funded by a PhD scholarship from the Spanish Ministry of Education, Culture and Sport under the FPU program. The authors acknowledge the funding from the following grants, which was used to develop the OASIS database by its creators: P50 AG05681, P01 AG03991, R01 AG021910, P50 MH071616, U24 RR021382, R01 MH56584. HCP data were provided [in part] by the Human Connectome Project, WU-Minn Consortium (Principal Investigators: David Van Essen and Kamil Ugurbil; 1U54MH091657) funded by the 16 NIH Institutes and Centers that support the NIH Blueprint for Neuroscience Research; and by the McDonnell Center for Systems Neuroscience at Washington University.

## References

- [1] C.-H. Pham, C. Tor-Díez, H. Meunier, N. Bednarek, R. Fablet, N. Passat and F. Rousseau, Multiscale brain MRI super-resolution using deep 3D convolutional networks, *Computerized Medical Imaging and Graphics* **77** (2019), 101647.
- [2] G. Mirzaei and H. Adeli, Segmentation and clustering in brain MRI imaging, *Reviews in the Neurosciences* **30**(1) (2018), 31–44.
- [3] J.J.R. Chong, Deep-Learning Super-Resolution MRI: Getting Something From Nothing, *Journal of Magnetic Resonance Imaging* (2019).
- [4] J.L. Prince, A. Carass, C. Zhao, B.E. Dewey, S. Roy and D.L. Pham, Chapter 1 - Image synthesis and superresolution in medical imaging, in: *Handbook of Medical Image Computing and Computer Assisted Intervention*, S.K. Zhou, D. Rueckert and G. Fichtinger, eds, Academic Press, 2020, pp. 1–24.
- [5] X. Zhao, Y. Zhang, T. Zhang and X. Zou, Channel Splitting Network for Single MR Image Super-Resolution, *IEEE Transactions on Image Processing* **28**(11) (2019), 5649–5662.
- [6] A. Rueda, N. Malpica and E. Romero, Single-image super-resolution of brain MR images using overcomplete dictionaries, *Medical Image Analysis* **17**(1) (2013), 113–132.
- [7] C. Dong, C.C. Loy, K. He and X. Tang, Image Super-Resolution Using Deep Convolutional Networks, 2014.
- [8] C.H. Pham, A. Ducournau, R. Fablet and F. Rousseau, Brain MRI super-resolution using deep 3D convolutional networks, in: *2017 IEEE 14th International Symposium on Biomedical Imaging (ISBI 2017)*, 2017, pp. 197–200.
- [9] M.H. Rafiei and H. Adeli, A New Neural Dynamic Classification Algorithm, *IEEE Transactions on Neural Networks and Learning Systems* **28**(12) (2017), 3074–3083.
- [10] C. Hua, H. Wang, H. Wang, S. Lu, C. Liu and S.M. Khalid, A Novel Method of Building Functional Brain Network Using Deep Learning Algorithm with Application in Proficiency Detection, *International Journal of Neural Systems* **29**(01) (2019), 1850015.
- [11] M. Koziarski and B. Cyganek, Image recognition with deep neural networks in presence of noise—Dealing with and taking advantage of distortions, *Integrated Computer-Aided Engineering* **24**(4) (2017), 337–349.
- [12] M.A. Molina-Cabello, R.M. Luque-Baena, E. López-Rubio and K. Thurnhofer-Hemsi, Vehicle type detection by ensembles of convolutional neural networks operating on super-resolved images, *Integrated Computer-Aided Engineering* **25**(4) (2018), 321–333.
- [13] P. Wang and X. Bai, Regional parallel structure based CNN for thermal infrared face identification, *Integrated Computer-Aided Engineering* **25**(3) (2018), 247–260.
- [14] F. Ortega-Zamorano, J.M. Jerez, I. Gómez and L. Franco, Layer multiplexing FPGA implementation for deep back-propagation learning, *Integrated Computer-Aided Engineering* **24**(2) (2017), 171–185.
- [15] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu and F.E. Alsaadi, A survey of deep neural network architectures and their applications, *Neurocomputing* **234** (2017), 11–26.
- [16] J. Shi, Z. Li, S. Ying, C. Wang, Q. Liu, Q. Zhang and P. Yan, MR Image Super-Resolution via Wide Residual Networks With Fixed Skip Connection, *IEEE Journal of Biomedical and Health Informatics* **23**(3) (2019), 1129–1140.
- [17] Z. Li, J. Yu, Y. Wang, H. Zhou, H. Yang and Z. Qiao, Deep-Volume: Brain Structure and Spatial Connection-Aware Network for Brain MRI Super-Resolution, *IEEE Transactions on Cybernetics* (2019), 1–14.
- [18] Y. Chen, H.C. So, E.E. Kuruoglu and X.L. Yang, Variance analysis of unbiased complex-valued Lp-norm minimizer, *Signal Processing* **135** (2017), 17–25.
- [19] L. Gao, Numerical algorithms for nonlinear Lp-norm problem and its extreme case, *Journal of Computational and Applied Mathematics* **129**(1) (2001), 139–150.
- [20] Y. Chen, H.C. So and E.E. Kuruoglu, Variance analysis of unbiased least Lp-norm estimator in non-Gaussian noise, *Signal Processing* **122** (2016), 190–203.
- [21] A.J. Grove, N. Littlestone and D. Schuurmans, General Convergence Results for Linear Discriminant Updates, *Machine Learning* **43**(3) (2001), 173–210.
- [22] C. Gentile, The Robustness of the p-Norm Algorithms, *Machine Learning* **53**(3) (2003), 265–299.
- [23] D. Blueschke and I. Savin, No such thing as a perfect hammer: comparing different objective function specifications for optimal control, *Central European Journal of Operations Research* **25**(2) (2017), 377–392.
- [24] X. Chen, F. Xu and Y. Ye, Lower Bound Theory of Nonzero Entries in Solutions of l2-lp Minimization, *SIAM Journal on Scientific Computing* **32**(5) (2010), 2832–2852.
- [25] Z. Li, J. Tang and X. He, Robust Structured Nonnegative Matrix Factorization for Image Representation, *IEEE Transactions on Neural Networks and Learning Systems* **29**(5) (2018), 1947–1960.
- [26] C. Zhang, D. Li and J. Tan, The Support Vector Regression with Adaptive Norms, *Procedia Computer Science* **18** (2013), 1730–1736.
- [27] F. Abramovich, Y. Benjamini, D.L. Donoho and I.M. Johnstone, Adapting to unknown sparsity by controlling the false discovery rate, *The Annals of Statistics* **34**(2) (2006), 584–653.
- [28] Y.-F. Ye, Y.-H. Shao, N.-Y. Deng, C.-N. Li and X.-Y. Hua, Robust Lp-norm least squares support vector regression with feature selection, *Applied Mathematics and Computation* **305**

- (2017), 32–52.
- [29] M.T.M. Emmerich and A.H. Deutz, A tutorial on multiobjective optimization: fundamentals and evolutionary methods, *Natural Computing* **17**(3) (2018), 585–609.
- [30] A. Zhou, B.-Y. Qu, H. Li, S.-Z. Zhao, P.N. Suganthan and Q. Zhang, Multiobjective evolutionary algorithms: A survey of the state of the art, *Swarm and Evolutionary Computation* **1**(1) (2011), 32–49.
- [31] L. Su, K. Yang, H. Hu and Z. Yang, Long-Term Hydropower Generation Scheduling of Large-Scale Cascade Reservoirs Using Chaotic Adaptive Multi-Objective Bat Algorithm, *Water* **11**(11) (2019), 2373.
- [32] H. Li, F. He, Y. Liang and Q. Quan, A dividing-based many-objective evolutionary algorithm for large-scale feature selection, *Soft Computing* (2019).
- [33] M. Ehrgott and D.M. Ryan, Constructing robust crew schedules with bicriteria optimization, *Journal of Multi-Criteria Decision Analysis* **11**(3) (2002), 139–150.
- [34] R.E. Steuer and E.-U. Choo, An interactive weighted Tchebycheff procedure for multiple objective programming, *Mathematical programming* **26**(3) (1983), 326–344.
- [35] G. Eichfelder, *Adaptive scalarization methods in multiobjective optimization*, Vol. 436, Springer, 2008.
- [36] M. Datar, A. Gionis, P. Indyk and R. Motwani, Maintaining stream statistics over sliding windows, *SIAM Journal on Computing* **31**(6) (2002), 1794–1813.
- [37] J. Han, J.S. Kim, C.K. Chung and K.S. Park, Evaluation of smoothing in an iterative Lp-norm minimization algorithm for surface-based source localization of MEG, *Physics in Medicine and Biology* **52**(16) (2007), 4791–4803.
- [38] I. Giagkiozis and P.J. Fleming, Methods for multi-objective optimization: An analysis, *Information Sciences* **293** (2015), 338–350.
- [39] Z. Bar-Yossef, T.S. Jayram, R. Kumar and D. Sivakumar, An information statistics approach to data stream and communication complexity, *Journal of Computer and System Sciences* **68**(4) (2004), 702–732.
- [40] M.A. Kerahroodi, A. Aubry, A. De Maio, M.M. Naghsh and M. Modarres-Hashemi, A coordinate-descent framework to design low PSL/ISL sequences, *IEEE Transactions on Signal Processing* **65**(22) (2017), 5942–5956.
- [41] E.E. Kuruoğlu, P.J.W. Rayner and W.J. Fitzgerald, Least Lp-norm estimation of autoregressive model coefficients of symmetric  $\alpha$ -stable processes, *IEEE Signal Processing Letters* **4**(7) (1997), 201–203.
- [42] Y. Zhang, L. Comerford, I.A. Kougioumtzoglou and M. Beer, Lp-norm minimization for stochastic process power spectrum estimation subject to incomplete data, *Mechanical Systems and Signal Processing* **101** (2018), 361–376.
- [43] J.M. Bioucas-Dias and G. Valadão, Phase unwrapping via graph cuts, *IEEE Transactions on Image Processing* **16**(3) (2007), 698–709.
- [44] M. Unser, A. Aldroubi and M. Eden, On the Asymptotic Convergence of B-spline Wavelets to Gabor Functions, *IEEE Transactions on Information Theory* **38**(2) (1992), 864–872.
- [45] R.J. Hathaway, J.C. Bezdek and Y. Hu, Generalized fuzzy c-means clustering strategies using Lp norm distances, *IEEE Transactions on Fuzzy Systems* **8**(5) (2000), 576–582.
- [46] S. Park and N. Kwak, Independent component analysis by Lp-norm optimization, *Pattern Recognition* **76** (2018), 752–760.
- [47] R. Kasimbeyli, Z.K. Ozturk, N. Kasimbeyli, G.D. Yalcin and B.I. Erdem, Comparison of Some Scalarization Methods in Multiobjective Optimization, *Bulletin of the Malaysian Mathematical Sciences Society* **42**(5) (2019), 1875–1905.
- [48] J. Cheng, G. Zhang, F. Caraffini and F. Neri, Multicriteria adaptive differential evolution for global numerical optimization, *Integrated Computer-Aided Engineering* **22**(2) (2015), 103–107.
- [49] S. Gass and T. Saaty, The computational algorithm for the parametric objective function, *Naval Research Logistics Quarterly* **2**(1–2) (1955), 39–45.
- [50] X.H. Gong, Chebyshev scalarization of solutions to the vector equilibrium problems, *Journal of Global Optimization* **49**(4) (2011), 607–622.
- [51] V.J. Bowman, On the Relationship of the Tchebycheff Norm and the Efficient Frontier of Multiple-Criteria Objectives, in: *Multiple Criteria Decision Making*, H. Thiriez and S. Zionts, eds, Springer Berlin Heidelberg, Berlin, Heidelberg, 1976, pp. 76–86.
- [52] H. Greenspan, Super-Resolution in Medical Imaging, *The Computer Journal* **52**(1) (2008), 43–63.
- [53] F. Shi, J. Cheng, L. Wang, P. Yap and D. Shen, LRTV: MR Image Super-Resolution With Low-Rank and Total Variation Regularizations, *IEEE Transactions on Medical Imaging* **34**(12) (2015), 2459–2466.
- [54] X. Du and Y. He, Gradient-Guided Convolutional Neural Network for MRI Image Super-Resolution, *Applied Sciences* **9**(22) (2019), 4874.
- [55] K. Zeng, H. Zheng, C. Cai, Y. Yang, K. Zhang and Z. Chen, Simultaneous single- and multi-contrast super-resolution for brain MRI images based on a convolutional neural network, *Computers in Biology and Medicine* **99** (2018), 133–141.
- [56] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama and T. Darrell, Caffe: Convolutional Architecture for Fast Feature Embedding, *arXiv preprint arXiv:1408.5093* (2014).
- [57] Y. Chen, Y. Xie, Z. Zhou, F. Shi, A.G. Christodoulou and D. Li, Brain MRI super resolution using 3D deep densely connected neural networks, in: *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, 2018, pp. 739–742.
- [58] D.C.V. Essen, S.M. Smith, D.M. Barch, T.E.J. Behrens, E. Yacoub and K. Ugurbil, The WU-Minn Human Connectome Project: An overview, *NeuroImage* **80** (2013), 62–79, Mapping the Connectome.
- [59] D. Marcus, J. Harwell, T. Olsen, M. Hodge, M. Glasser, F. Prior, M. Jenkinson, T. Laumann, S. Curtiss and D. Van Essen, Informatics and Data Mining Tools and Strategies for the Human Connectome Project, *Frontiers in Neuroinformatics* **5** (2011), 4.
- [60] C.B. Paschal and H.D. Morris, K-space in the clinic, *Journal of Magnetic Resonance Imaging* **19**(2) (2004), 145–159.
- [61] B.A. Landman, A.J. Huang, A. Gifford, D.S. Vikram, I.A.L. Lim, J.A. Farrell, J.A. Bogovic, J. Hua, M. Chen, S. Jarso et al., Multi-parametric neuroimaging reproducibility: a 3-T resource study, *NeuroImage* **54**(4) (2011), 2854–2866.
- [62] A.J. Worth, MGH CMA Internet Brain Segmentation Repository (IBSR), 2010.
- [63] Z. Wang, A.C. Bovik, H.R. Sheikh and E.P. Simoncelli, Image Quality Assessment: From Error Visibility to Structural Similarity, *IEEE Transactions on Image Processing* **13**(4) (2004), 600–612.
- [64] A. Bhattacharyya, On a Measure of Divergence between Two

- Multinomial Populations, *Sankhyā: The Indian Journal of Statistics (1933-1960)* **7**(4) (1946), 401–406.
- [65] K. Thurnhofer-Hemsi, E. López-Rubio, N. Roé-Vellvé and M.A. Molina-Cabello, Deep Learning Networks with p-norm Loss Layers for Spatial Resolution Enhancement of 3D Medical Images, in: *From Bioinspired Systems and Biomedical Applications to Machine Learning*, J.M. Ferrández Vicente, J.R. Álvarez-Sánchez, F. de la Paz López, J. Toledo Moreo and H. Adeli, eds, Springer International Publishing, Cham, 2019, pp. 287–296.
- [66] S. García, A. Fernández, J. Luengo and F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power, *Information Sciences* **180**(10) (2010), 2044–2064.
- [67] J. Derrac, S. García, D. Molina and F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, *Swarm and Evolutionary Computation* **1**(1) (2011), 3–18.
- [68] M.J. Gómez-Silva, E. Izquierdo, A.d.l. Escalera and J.M. Armingol, Transferring learning from multi-person tracking to person re-identification, *Integrated Computer-Aided Engineering* **26**(4) (2019), 329–344.