Article

QMRNet: Quality Metric Regression for EO Image Quality Assessment and Super-Resolution

David Berga [1,*], Pau Gallés [2,†], Katalin Takáts [2,†], Eva Mohedano [2,†], Laura Riordan-Chen [2,†], Clara Garcia-Moll [2,†], David Vilaseca [2,†] and Javier Marín [2,†]

1 Eurecat, Centre Tecnològic de Catalunya, Tecnologies Multimèdia, 08005 Barcelona, Spain
2 Satellogic Inc., Davidson, NC 28036, USA
* Correspondence: david.berga@eurecat.org; Tel.: +34-932-38-14-00
† These authors contributed equally to this work.

Abstract: The latest advances in super-resolution have been tested with general-purpose images such as faces, landscapes and objects, but mainly unused for the task of super-resolving earth observation images. In this research paper, we benchmark state-of-the-art SR algorithms for distinct EO datasets using both full-reference and no-reference image quality assessment metrics. We also propose a novel Quality Metric Regression Network (QMRNet) that is able to predict the quality (as a no-reference metric) by training on any property of the image (e.g., its resolution, its distortions, etc.) and also able to optimize SR algorithms for a specific metric objective. This work is part of the implementation of the framework IQUAFLOW, which has been developed for the evaluation of image quality and the detection and classification of objects as well as image compression in EO use cases. We integrated our experimentation and tested our QMRNet algorithm on predicting features such as blur, sharpness, snr, rer and ground sampling distance and obtained validation medRs below 1.0 (out of $N = 50$) and recall rates above 95%. The overall benchmark shows promising results for LIIF, CAR and MSRN and also the potential use of QMRNet as a loss for optimizing SR predictions. Due to its simplicity, QMRNet could also be used for other use cases and image domains, as its architecture and data processing is fully scalable.

Keywords: super-resolution; quality assessment; benchmark; denoising; regression; autoencoder networks; generative adversarial networks; self-supervision; regularization; optimization; earth observation

1. Introduction

One of the main issues in observing and analyzing earth observation (EO) images is to estimate its quality. However, this main issue is twofold. First, images are captured with distinct image modifications and distortions, such as optical diffractions and aberrations, detector spacing and footprints, atmospheric turbulence, platform vibration, blurring, target motion, and postprocessing. Second, EO image resolution is very limited due to the sensor's optical resolution, the satellite's and connection's capacity to send high-quality images to the ground as well as the captured ground sampling distance (GSD) [1]. These limitations make the image quality assessment (IQA) hard to evaluate for EO particularly, as there are no comparable fine-grained baselines in broad EO domains.

We will tackle these problems by defining a network that acts as a no-reference (blind) metric, assessing the quality and optimizing the super-resolution of EO images at any scale and modification.

Below are summarized our **main contributions**:

- We train and validate a novel network (QMRNet) for EO imagery that is able to predict any type of image based on its quality and distortion
- (Case 1) We benchmark distinct super-resolution models with QMRNet and compare the results with full-reference, no-reference and feature-based metrics

- (Case 2) We benchmark distinct EO datasets with QMRNet scores
- (Case 3) We propose to use QMRNet as a loss for optimizing the quality of super-resolution models

Super-resolution (SR) consists of estimating a high-resolution image (I_{HR}) given a low-resolution one (I_{LR}). In the deep learning era, deep networks have been used to classify images, obtaining a high precision in their predictions. For the specific SR task, one can design a network (**autoencoder**) whose convolutional layers (feature extractor) encode the patches of the image in order to build a feature vector (encoder) from the image, then add deconvolutional layers to reconstruct the original image (decoder). The instances of the predicted images are compared with the original ones in order to re-train the autoencoder network until they converge to an HR objective. The SRCNN and FSRCNN models [2,3] are based on a network of three blocks (patch extraction and representation, nonlinear mapping and reconstruction). The authors also mention the use of rotation, scaling and noise transformations as data augmentation prior to training the network. The authors use downscaling with a low-pass filter to obtain the I_{LR} images and use a bicubic interpolation for the upscaling during reconstruction to obtain the I_{SR} (the model's prediction of I_{HR}). SRCNN has been used by MC-SRCNN [4] to super-resolve multi-spectral images by changing the architecture's input channels and adding pan-sharpening filters (modulating the smoothing/sharpening intensity). These design principles used in autoencoders, however, have a drawback in that they work differently over feature-size frequencies and features at distinct resolutions. For that, multi-scale architectures are proposed. The Multi-Scale Residual Network (MSRN) [5] uses residual connections in multiple residual blocks at different scale bands, non-exclusive to ResNets. It ables the equalization of the information bottleneck in deeper layers (high-level features) where the spatial information in some cases tends to diminish or either vanish. Traditional convolutional filters in primary layers have a fixed and narrow field of view, which creates dependencies to the learning of spatial long-range connections and deeper layers. However, multi-scale blocks cope with this drawback by analyzing the image domain at different resolution scales to be later merged in a high-dimensional multiband latent space. This allows a better abstraction at deeper layers and, therefore, the reconstruction of spatial information. This is a remarkable advantage when using EO images, which come with distinct resolutions and GSD.

Novel state-of-the-art SR models are based on **generative adversarial networks (GANs)**. These networks are composed of two networks, a generator that generates an image estimate (SR) and a discriminator that decides whether the generated image is real or fake under certain categorical or metric objectives with respect to the classification of a set of images "I". Usually, the generator is a deconvolutional network that is fed with a latent vector that represents the distribution for each image. In the SR problem, the LR (I_{LR}) is considered as the input latent space while the HR image is considered as the real image I_{HR} to obtain the adversarial loss. For the case of the popular SRGAN [6], it has been designed with adversarial loss through VGG and ResNet (SRResnet) with residual connections and perceptual loss. The ESRGAN [7] is an improved version of the SRGAN, although it uses adversarial loss relaxation, adds training upon perceptual loss and has some some residual connections in its architecture. The main intrinsic difference between GANs and other architectures is that the image probability distribution is intrinsically learned. This makes these architectures suffer from unknown artifacts and hallucinations; however, their SR estimates are usually sharper than autoencoder-type architectures. Some mentioned generative techniques for SR, such as SRGAN/SRResnet, ESRGAN and Enlighten-GAN [8], and convolutional SR autoencoders, such as VDSR [9], SRCNN/FRSCNN and MSRN, do not adapt their feature generation to optimize a loss based on a specific quality standard that considers all quality properties of the image (both structural and pixel-to-pixel). However, the predictions show typical distortions such as blurring (from downscaling the input) or GAN artifacts from the training domain objective. Most of these GAN-based models build the I_{LR} inputs of the network from downsampled data from the original I_{HR}. This

I_{LR} generation from downsampling I_{HR} limits the training of these models to perform the reverse transformation of the modification; however, the type of distortions and variations from any test image are a combination of much more diverse modifications. The only way to mitigate this limitation, but only partially due to overfitting, is to augment the I_{LR} samples to distinct transformations simultaneously.

Some **self-supervised** techniques can learn to solve the ill-posed inverse problem from the observed measurements, without any knowledge of the underlying distribution assuming its invariance to the transformations. The Content Adaptive Resampler (CAR) [10] was proposed, in which a join-learnable downscaling pre-step block together with an upscaling block (SRNet) is trained separately. It is able to learn the downscaling step (through a ResamplerNet) by learning the statistics of kernels from the I_{HR}, then it learns the upscaling blocks with another net (SRNet/EDSR) to obtain the SR images. CAR has been able to improve the experimental results of SR by considering the intrinsic divergences between I_{LR} and I_{HR}. The Local Implicit Image Function (LIIF) [11] is able to generate super-resolved pixels considering 2D deep features around these coordinates as inputs. In LIIF, an encoder is jointly trained in a self-supervised super-resolution task maintaining high fidelity at higher resolutions. Since the coordinates are continuous, LIIF can be presented in any arbitrary resolution. Here, the main advantage is that the SR is represented in a resolution without resizing I_{HR}, making it invariant to the transformations performed to the I_{LR}. This enables LIIF to extrapolate SR upon factors up to x30.

In order to assess the quality of an image, there are distinct strategies. Full-reference metrics consider the difference between an estimated or modified image (I_{SR}) and the reference image (I_{HR}). In contrast, no-reference metrics assess the specific statistical properties of the estimated image without any reference image. Other more novel metrics calculate the high-level characteristics of the estimated I_{SR} by comparing its distribution distance with respect to either a preprocessed dataset or the reference I_{HR} in a feature-based space.

The similarity between the predicted images I_{SR} and the reference high-resolution images I_{HR} is estimated by looking at the pixel-wise differences responsive to reflectance, sharpness, structure, noise, etc. Very well-known examples of **pixel-level (or full-reference) metrics** are the root-mean-square error (RMSE) [12], Spearman's rank order correlation coefficient (SRCC or SROCC), Pearson's linear correlation coefficient (PLCC), Kendall's rank order correlation coefficient (KROCC), the peak signal-to-noise ratio (PSNR) [13], the structural similarity metric (SSIM/MSSIM) [14], the Haar perceptual similarity index (HAARPSI) [15], the gradient magnitude similarity deviation (GMSD) [16] and the mean deviation similarity index (MDSI) [17]. PSNR calculates the power of the signal-to-noise ratio considering the noise error with respect to the I_{HR}. Some metrics such as the SSIM specifically measure the means and covariances locally for each region at a specific size (e.g., 8×8 patches; multi-scale patches for MSSIM) affecting the overall metric score. The GMSD calculates the global variation similarity of the gradient based on a local quality map combined with a pooling strategy. Most comparative studies use these metrics to measure the actual I_{SR} quality, mostly relying on PSNR, although there is no evidence that these measurements are the best for EO cases, as some of these are not sensitive to local perturbations (i.e., blurring, over-sharpening) and local changes (i.e., artifacts, hallucinations) to the image. The HAARPSI calculates an index based on the difference (absolute or in mutual information) using the sum of a set of wavelet coefficients processed over the I_{SR}-I_{HR} images. Other cases of metrics combine some of the pinpointed parameters simultaneously. For instance, the MDSI compares jointly the gradient similarity, chromaticity similarity and deviation pooling. The latest metric design, LCSA [18], uses linear combinations of full-reference metrics, i.e., VSI, FSIM, IFC, MAD, MSSSIM, NQM, PSNR, SSIM and VIF.

Pixel-reference metrics have a main requirement, which is that the ground-truth HR images are needed to assess a specific quality standard. For the case of **no-reference (or blind) metrics**, no explicit reference is needed. These rely on a parametric characterization of the enhanced signal based on statistic descriptors, usually linked to the noise or sharpness, embedded in high-frequency bands. Some examples are the variance, entropy (He),

or high-end spectrum (FFT). The main popular metric in EO is the modulation transfer function (MTF), which measures impulse responses in the spatial domain and transfer functions in the frequency domain. This varies upon overall local pixel characteristics mostly present on contours, corners and sharp features in general [19]. Here, the MTF is very sensitive to local changes such as those aforementioned (e.g., optical diffractions and aberrations, blurring, motion, etc). Other metrics would use statistics from image patches in combination with multivariate filtering methods to propose score indexes for a given predefined image given its geo-referenced parameter standards. Such methods include NIQE [20], PIQE [21] and GIQE [22]. The latter is considered for official evaluation of NIIRS ratings (https://irp.fas.org/imint/niirs.htm, accessed on 10 October 2022) considering the ground sampling distance (GSD), the signal-to-noise ratio (SNR) and the relative edge response (RER) in distinct effective focal lengths of EO images [23–25]. Note that RER measures the line spread function (LSF), which corresponds to the absolute impulse response also computed by the MTF. The relative edge response measures the slope in the edge response (transition). The lower the metric, the blurrier the image. Taking the derivative of the normalized edge response produces the line spread function (LSF). The LSF is a 1D representation of the system point sparsity function (PSF). The width of the LSF at half the height is called the full width at half maximum (FWHM). The Fourier transform of the LSF produces the modulation transfer function (MTF). The MTF is determined across all spatial frequencies, but can be evaluated at a single spatial frequency, such as the Nyquist frequency. The value of the MTF at Nyquist provides a measure of the resolvable contrast at the highest 'alias-free' spatial frequency.

In [26], the authors argued that the conventional IQA evaluation methods are not valid for EO as the degradation functions and operation hardware conditions do not meet the operational conditions. Through advances in DL in that aspect, deeper network representations have been shown to improve the perceptual quality of images, although with higher requirements. The concept of **feature-based metrics** (i.e., perceptual similarity) is defined by the score reference on these trained features (i.e., the generator or reconstruction network). These metrics compare the distances between latent features from the predicted image and the reference image. Some state-of-the-art methods of perceptual similarity include the VGGLoss [27] and the Learned Perceptual Image Patch Similarity (LPIPS) [28], which measure the feature maps obtained by the n-th convolution after activation (image-reference layer n) and then calculate the similarity using the Euclidean distance between the predicted I_{SR} model features and the reference image features. Some other metrics such as the sliced Wasserstein distance (SWD) [29] and the Fréchet inception distance (FID) [30] assume a non-linear space modelling for the feature representations to compare, and therefore can adapt better with larger variability or a lack of samples in the training image domains.

2. Datasets and Related Work

Most non-feature-based metrics are fully unsupervised, namely, there are no current models that specifically can assess the image quality invariably from the specific modifications made on images specific to a certain domain. Blind quality ranking and assessment of images has been useful for applications such as avoiding forgetting in continual learning adaptation tasks [31] and many others, such as compression evaluation and mean opinion scores (MOS/CMOS). One novel strategy, ProxIQA [32], tries to evaluate the quality of an image by adapting the underlying distribution of a GAN given a compressed input. This method has been shown to improve the quality when tested on images from compression datasets from Kodak, Teknick and NFLX, although the results may vary among the trained image distributions, as shown by the JPEG2000, VMAFp and HEVC metrics. Traditional blind IQA methods (i.e., BLIINDS-II [33], BRISQUE [34], CORNIA [35], HOSA [36] and RankIQA [37]) as well as the latest deep blind image-quality assessment models such as WaDIQaM (deepIQA) [38], IQA-MCNN [39], Meta-IQA [40] and GraphIQA [41] propose to benchmark distortion-aware datasets (e.g., LIVE, LIVEC, CSIQ, KonIQ10k, TID2013 and KADID-10k) with already-distorted images and MOS/CMOS. These train and assess upon

annotated exemplars such as Gaussian blur, lens blur, motion blur, color quantization, color saturation, etc. SRIF [42], RealSRQ [43] and DeepSRQ [44] explore deep learning strategies as no-reference metrics for SR quality estimation, although they have only been tested in generic datasets such as CVIU, SRID, QADS and Waterloo. Additionally, most of these models do not integrate their own modifiers that are able to customize ranking metrics (i.e., are limited to the available synthetic annotations from the aforementioned datasets). Some of these could include geo-reference annotations from the actual EO missions, such as the GSD, Nadir angle, atmospheric data, etc. The usage of customizable modifiers allows the fine-tuning on distortions on any existing domain, in our case, HR EO images. It has also not been demonstrated for IQA methods to integrate with super-resolution model benchmarking and re-training. Understanding and building the mechanics of distortions (geometrics and modifiers) is thus key for the generation of the necessary samples to train a network with enough samples to represent the whole domain.

Very few studies on SR use EO images obtained from current worldwide satellites such as DigitalGlobe WorldView-4 (https://earth.esa.int/eogateway/missions/worldview-4, accessed on 10 October 2022), SPOT (https://earth.esa.int/eogateway/missions/spot, accessed on 10 October 2022), Sentinel-2 (https://sentinels.copernicus.eu/web/sentinel/missions/sentinel-2, accessed on 10 October 2022), Landsat-8 (https://www.usgs.gov/landsat-missions/landsat-8, accessed on 10 October 2022), Hyperion/ EO-1 (https://www.usgs.gov/centers/eros/science/usgs-eros-archive-earth-observing-one-eo-1-hyperion, accessed on 10 October 2022), SkySat (https://earth.esa.int/eogateway/missions/skysat, accessed on 10 October 2022), Planetscope (https://earth.esa.int/eogateway/missions/planetscope, accessed on 10 October 2022), RedEye (https://space.skyrocket.de/docs_dat/red-eye.htm, accessed on 10 October 2022), QuickBird (https://earth.esa.int/eogateway/missions/quickbird-2, accessed on 10 October 2022), CBERS (https://www.satimagingcorp.com/satellite-sensors/other-satellite-sensors/cbers-2/, accessed on 10 October 2022), Himawari-8 (https://www.data.jma.go.jp/mscweb/data/himawari/, accessed on 10 October 2022), DSCOVR EPIC (https://epic.gsfc.nasa.gov/, accessed on 10 October 2022) or PRISMA (https://www.asi.it/en/earth-science/prisma/, accessed on 10 October 2022). In our study we selected a variety of subsets (see Table 1) from distinct online general public domain satellite imagery datasets with high resolution (around 30 cm/px). Most of these are used for land use classification tasks, with coverage category annotations and some with object segmentation. The Inria Aerial Image Labeling Dataset [45] (Inria-AILD) (https://project.inria.fr/aerialimagelabeling/, accessed on 10 October 2022) contains 180 training and 180 test images covering $405 + 405$ km^2 of US (Austin, Chicago, Kitsap County, Bellingham, Bloomington, San Francisco) and Austrian (Innsbruck Eastern/Western Tyrol, Vienna) regions. Inria-AILD was used for a semantic segmentation of buildings contest. Some land cover categories are considered for aerial scene classification in DeepGlobe (http://deepglobe.org/, accessed on 10 October 2022) (Urban, Agriculture, Rangeland, Forest, Water or Barren), USGS (https://data.usgs.gov/datacatalog/, accessed on 10 October 2022) and UCMerced (http://weegee.vision.ucmerced.edu/datasets/landuse.html, accessed on 10 October 2022) with 21 classes (i.e., agricultural, airplane, baseball diamond, beach, buildings, chaparral, dense residential, forest, freeway, golf course, harbor, intersection, medium residential, mobile home park, overpass, parking lot, river, runway, sparse residential, storage tanks and tennis court). The latter has been captured for many US regions, i.e., Birmingham, Boston, Buffalo, Columbus, Dallas, Harrisburg, Houston, Jacksonville, Las Vegas, Los Angeles, Miami, Napa, New York, Reno, San Diego, Santa Barbara, Seattle, Tampa, Tucson and Ventura. XView (http://xviewdataset.org/, accessed on 10 October 2022) contains 1.400 km^2 RGB pan-sharpened images from DigitalGlobe WorldView-3 with 1 million labeled objects and 60 classes (e.g., Building, Hangar, Train, Airplane, Vehicle, Parking Lot) annotated both with bounding boxes and segmentation. Kaggle Shipsnet (https://www.kaggle.com/datasets/rhammell/ships-in-satellite-imagery, accessed on 10 October 2022) contains seven San Francisco Bay harbor images and 4000 individual crops of ships captured in the dataset. The ECODSE

competition dataset (https://zenodo.org/record/1206101, accessed on 10 October 2022), (https://www.ecodse.org/task3_classification.html, accessed on 10 October 2022) has been considered for EO hyperspectral image classification [46], delineation (segmentation) and alignment of trees. ECODSE has available NEON photographs, LiDAR data for assessing canopy height and hyperspectral images with 426 bands. The terrain is photographed with a mean altitude of 45 m.a.s.l. and the mean canopy height is approximately 23 m.

Table 1. List of datasets used in our experimentation. We show 12 subsets collected from 8 datasets provided by 5 satellites and EO stations.

Dataset-Subset	#Set/#Total	GSD	Resolution	Spatial Coverage	Year	Provider
USGS	279/279	30 cm/px	5000×5000	349 km^2 (US regions)	2000	USGS (LandSat)
UCMerced-380	380/2100	30 cm/px	256×256	1022/5652 (US regions)	2010	USGS (LandSat)
UCMerced-2100	2100/2100	30 cm/px	232×232	5652 km^2 (US regions)	2010	USGS (LandSat)
Inria-AILD-180-train	100/360	30 cm/px	5000×5000	405/810 km^2 (US and Austria)	2017	arcGIS
Inria-AILD-180-val	20/360	30 cm/px	5000×5000	405/810 km^2 (US and Austria)	2017	arcGIS
Inria-AILD-180-test	180/360	30 cm/px	5000×5000	405/810 km^2 (US and Austria)	2017	arcGIS
ECODSE-hs (C = 426)	43/129	~60 cm/px	80×80	37/37 km^2 (Florida, US)	2018	OSBS
Shipsnet-Scenes	7/7	3 m/px	3000×1500	28 km^2 (San Francisco Bay)	2018	Open California
Shipsnet-Ships	4000/4000	3 m/px	80×80	28 km^2 (San Francisco Bay)	2018	(Planetscope)
DeepGlobe	469/1146	31 cm/px	2448×2448	703/1.717 km^2 (Germany)	2018	Worldview-3
Xview-train	846/1127	30 cm/px	5000×5000	1050/1.400 km^2 (Global)	2018	WorldView-3
Xview-validation	281/1127	30 cm/px	5000×5000	349/1.400 km^2 (Global)	2017	WorldView-3

3. Proposed Method

3.1. Iquaflow Modifiers and Metrics

We have developed a novel framework IQUAFLOW [47,48] (code available at https://github.com/satellogic/iquaflow, accessed on 10 October 2022) with set of modifiers (https://github.com/satellogic/iquaflow/tree/main/iquaflow/datasets, accessed on 10 October 2022) that apply a specific type of distortion in EO images. In the modifiers list (see Table 2) we describe 5 modifiers we developed for our experimentation, 3 of which have been integrated from common libraries (Pytorch (https://pytorch.org/vision/stable/transforms.html, accessed on 10 October 2022), PIL (https://pillow.readthedocs.io/en/stable/reference/Image.html, accessed on 10 October 2022), such as the *blur* (σ), sharpness factor (*F*) and detected ground sampling distance (*GSD*), and 2 (*snr* and *rer*) that we developed to represent the RER and SNR metric modifications. For the case of *blur*, we build a Gaussian filter with kernel 7×7 and we parameterize the σ. For the case of *F*, similarly, we build a function that is modulated by a Gaussian factor (similar to a σ). If the factor is higher than 1.0 (i.e., from 1.0 to 10.0), the image is sharpened (high-pass filter, with negative values on the sides of the kernel). However, if the factor is lower than 1.0 (i.e., from 0.0 to 1.0) then the image is blurred through a Gaussian function (low-pass filter with Gaussian shape). For the case of GSD, we apply a bilinear interpolation on the original image to a specific scaling (e.g., $\times 1.5$, $\times 2$), which will increase the scaling resolution of objects. In this case, an interpolated version of a 5000×5000 image of GSD 30 cm/px will be 10,000 $\times$ 10,000 and its GSD 60 cm/px, as its resolution has changed but the (oversampled/fake) sampling distance is doubled (worse). For the case of RER, we obtain the real RER value from the ground truth and calculate the LSF and max value of edge response. From that, we build a Gaussian function that is adapted to the expected RER coefficients and then filter the image. For SNR, similarly to RER, we require annotation of the base SNR from the original dataset. From that, we build a randomness regime that is adapted to a Gaussian shape that will be summed to the original image (adding randomness with a specific σ slope probability).

Table 2. List of modifier parameters used in QMRNet. These modify the input images and annotate them to provide training and test data for the QMRNet. Distinct intervals have been selected according to the precision and variability of the modification. ↑↓ represents closest values to *GT*.

Algorithm	Acronym	Parameters	#Intervals (N)	Range	Properties
Gaussian Blur	*blur*	Blur Sigma (σ)	50	0.0 to 2.5	*Quality ↓, Distortion ↑*
Gaussian Sharpness	*F*	Sharpness Factor (F)	9	1.0 to 10.0	*Quality ↑↓, Distortion ↑*
Ground Sampling Distance	GSD	GSD or scaling	10	0.30 to 0.60 ($\times 1 \ldots \times 2$)	*Quality ↓, Distortion ↑*
Relative Edge Response	*rer*	RER (MTF-Sharpness)	40	0.15 to 0.55	*Quality ↑↓, Distortion ↓*
Signal-to-Noise Ratio	*snr*	Noise (Gaussian) Ratio	40	15 to 30	*Quality ↑↓, Distortion ↓*

3.2. QMRNet: Classifier of Modifier Metric Parameters

We have designed the Quality Metric Regression Network (QMRNet) to be able to regress the quality parameters upon the modification or distortion (see Tables 2 and 3) applied to single images (code for QMRNet in https://github.com/satellogic/iquaflow/tree/main/iquaflow/quality_metrics, accessed on 10 October 2022). Given a set of images, modified through a Gaussian blur (σ), sharpness (Gaussian factor *F*), a rescaling to a distinct GSD, noise (SNR), or any kind of distortion, the images are annotated with that parameter. These annotations can be used by training and validating the network upon classifying the intervals corresponding to the annotated parameters.

QMRNet is a feed-forward neural network that takes the architecture of an encoder with a parametrizable classifier (see Figure 1) upon numerical class intervals (can be set as binary, categorical or continuous according to the N intervals). It trains upon the predicted interval differences and the annotated parameters of the ground truth (GT) and requires a HEAD for each parameter to predict. In Figure 2 we designed 2 mechanisms of assessing the quality from several parameters simultaneously (multiparameter prediction): multibranch (MB) and multihead (MH). For MB a single encoder and head is required for each parameter to predict, while MH requires a head for each parameter but only one encoder. Therefore, QMRNet-MH considers one encoder and #N classifiers per EO parameter while QMRNet-MB considers #N encoders + #N classifiers (a whole QMRNet per parameter). The QMRNet-MH predicts all parameters simultaneously (faster) but its capacity is lower (can lead to lower accuracy) from the encoder part.

For our experiments with QMRNet we have used an encoder based on ResNet18 (backbone) composed of a convolutional layer (3×3) and 4 residual blocks (each composed of 4 convolutional layers) of 64, 128, 256 and 512 pixels of resolution. Our network is scalable to distinct crop resolutions as well as regression parameters (N intervals), adapting the HEAD to the number of classes to predict. The output of the HEAD after pooling is a continuous value of probability of each class interval, and through softmax and thresholding we can filter (one-hot) which class or classes have been predicted (1) and not (0) for each image sample crop. By default, we utilize the Binary Cross Entropy Loss (BCELoss) as the classification error and Stochastic Gradient Descend as the optimizer. For the case of multiclass regression, we designed the multibranch QMRNet (QMRNet-MB), in which we train each network individually with its set of parameterized modification intervals for each sample. QMRNet-MB is trained individually but can be run once per sample (parallel threads per branch), especially to obtain fast multi-class metric calculations (see *score* in Section 4.1).

Table 3. Examples of Inria-AILD crops from modified images for each modifier (see Table 2).

	Original	← Lower	*Distortion*	→ Higher
Blur (σ)	σ = 1.0	σ = 1.5	σ = 2.0	σ = 2.5
Sharpness (F)	F = 1.0	F = 2.0	F = 5.0	F = 10.0
GSD	30 zoom (×800)	$36.\widehat{6}$ zoom (×720)	50 zoom (×506)	60 zoom (×400)
RER	0.55	0.30	0.25	0.15
SNR	30	25	20	15

Note that for processing irregular or inequivalent crops in our design of the network input, in the case of having the encoder input resolution R lower than the input image crops (e.g., 5000 × 5000 for GT and 256 × 256 for the network input), we crop the image to the QMRNet input R by C crops. C is the number of crops to generate for each sample (e.g., 10, 20, 50, 100, 200). In the case of the crops being smaller than the encoder backbone input (e.g., 232 × 232 for the GT and 256 × 256 for the network input), we apply a circular padding on each border (width and/or height) to obtain a real image that preserves the scaling and domain. The total number of hyperparameters to specify the design of a specific QMRNet architecture is N × R and it can be trained with a distinct combination of hyperparameters (N × C × R). To train the QMRNet's regressor, we select a training set and generate a set of distorted cases, which are parameterizable through our modifiers. The total number of training samples (dataset size) can be calculated by the product of the dataset images (I) and N × C (number of parameter intervals and crops per sample). We can set distinct possible hyperparameters specifically to train and validate, such as the

number of epochs (e), batch size (bs), learning rate (lr), weight decay (wd), momentum, soft thresholding, etc.
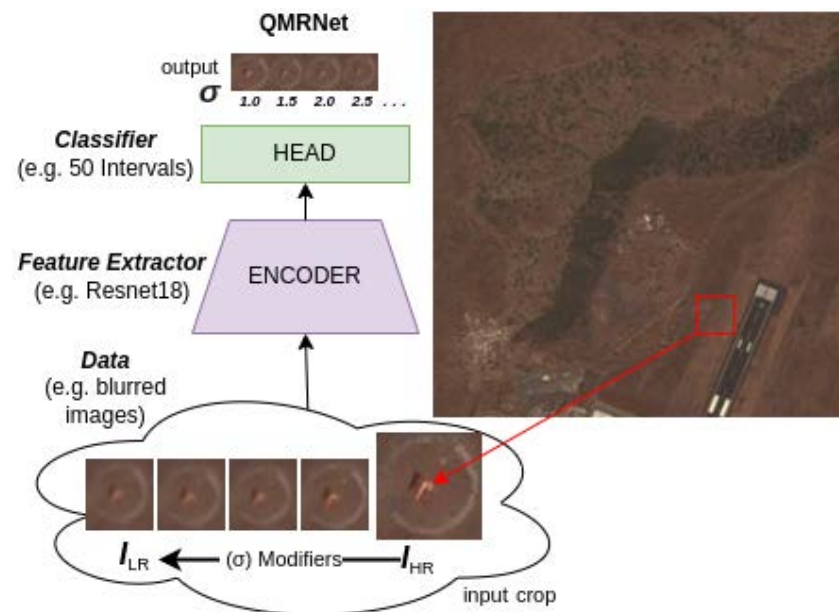
Figure 1. Architecture of QMRNet. Single-parameter QMRNet for using modified/annotated data from a unique parameter/modifier (in this case, from blur-modified images).

(**a**) QMRNet-MB   (**b**) QMRNet-MH
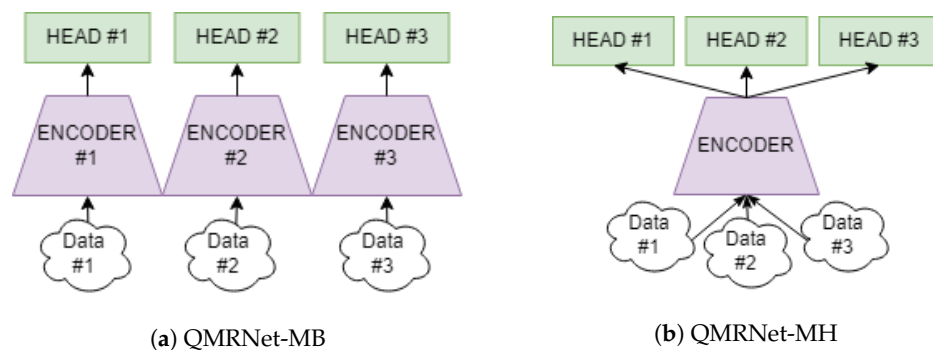
Figure 2. Multiparameter architectures to simultaneously predict several distortions in one run. (**a**) Multibranch QMRNet (QMRNet-MB), example with 3 stacked QMRNets (3 encoders with 1 head each). (**b**) Multihead QMRNet (QMRNet-MH), example with 3 heads.

3.3. QMRLoss: Learning Quality Metric Regression as Loss in SR

We designed a novel objective function that is able to optimize super-resolution algorithms upon a specific quality objective using QMRNet (see Figure 3). Given a GAN or autoencoder network, we can add an ad hoc module based on a specific (or several) parameters of QMRNet. The QMRLoss is obtained by computing the classification error between the I_{SR} prediction and the original I_{HR}. This classification error determines whether the SR image is distinct in terms of a quality parameter objective (i.e., *blur* σ, *F*, GSD, *rer* or *snr*) with respect the HR. The QMRLoss has been designed to use any classification error (i.e., BCE, L1 or L2) and can be summated to the perceptual or content loss of the generator (decoder for autoencoders) in order to tune the SR to the quality objective.

The objective function for image generation algorithms is based on minimizing the generator (G) error (which compares I_{HR} and I_{SR}) while maximizing the discriminator (D) error (which tests whether the SR image is true or fake).

$$\min_{G} \max_{D} = \mathbb{E}_{HR}\big[log(D(I_{HR})\big] + \mathbb{E}_{LR}\big[log(1 - D(I_{SR})\big] \qquad (1)$$
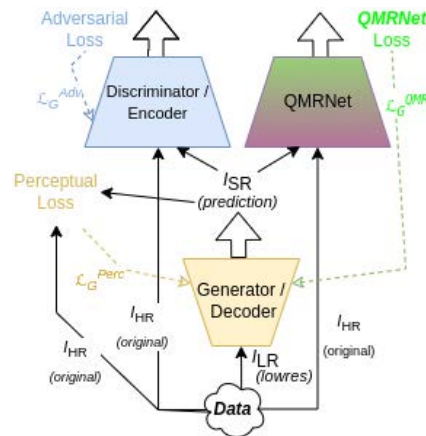
Figure 3. Super-resolution model pipeline (encoder–decoder for autoencoders and generator–discriminator for GANs) with ad hoc QMRNet loss optimization. Note that all losses (i.e., $\mathcal{L}_D^{Adv}$, $\mathcal{L}_G^{Perc}$ and $\mathcal{L}_G^{QMR}$) are considered for the case of MSRN optimization with QMRNet; see Section 4.4.

During training, G is optimized upon $\mathcal{L}^{SR}$, which considers $\mathcal{L}_G^{Perc}$ and $\mathcal{L}^{Adv}$. We added a new term, $\mathcal{L}_G^{QMR}$, which will be our loss function based on the quality objectives (QMRNet). Note that here we consider I_{SR} as the prediction image $G(I_{LR})$.

$$\mathcal{L}^{SR} = \mathcal{L}_G^{Perc} + \mathcal{L}_D^{Adv} + \mathcal{L}_G^{QMR}\lambda_{QMR} \tag{2}$$

$$\mathcal{L}_D^{Adv} = \sum -logD(I_{SR}) \tag{3}$$

$$\mathcal{L}_G^{Perc} = \frac{1}{n}\sum(I_{HR} - I_{SR})^{\{1,2\}} \tag{4}$$

Below, we define the term $\mathcal{L}^{QMR}$, which calculates the parameter difference between the I_{HR} images and I_{SR} images, regularized by the constant λ_{QMR}. This is performed by computing the classification error (L1, L2 or BCE) between the output of the heads for each case:

$$
\begin{aligned}
\mathcal{L}_{L1}^{QMR} &= \frac{1}{n}\sum(QMRNet(I_{HR}) - QMRNet(I_{SR})) \\
\mathcal{L}_{L2}^{QMR} &= \frac{1}{n}\sum(QMRNet(I_{HR}) - QMRNet(I_{SR}))^2 \\
\mathcal{L}_{BCE}^{QMR} &= -\frac{1}{n}\sum QMRNet(I_{HR})log(QMRNet(I_{SR})) \\
&\quad +(1 - QMRNet(I_{HR}))log(1 - QMRNet(I_{SR}))
\end{aligned}
\tag{5}
$$

4. Experiments

4.1. Experimental Setup

For training the QMRNet we collected 30cm/pixel data from the Inria Aerial Image Labeling Dataset (both training and validation using Inria-AILD sets). For testing our network, we selected all 11 subsets from the distinct EO datasets, USGS, UCMerced, Inria, DeepGlobe, Shipsnet, ECODSE and Xview (see Table 1).

4.1.1. Evaluation Metrics

In order to validate the training regime, we set several evaluation metrics (Tables 4–6) that provide interval dependencies for each prediction, namely, that intervals that are closer to the target interval are considered better predictions that further ones. This means that given an unblurred image (*blur* $\sigma = 1.0$) the prediction of $\sigma = 2.5$ will be a worse prediction than predictions closer to the GT (e.g., $\sigma = 1.03$, $\sigma = 1.2$). For this, we considered retrieval

metrics (which are N-rank order classification) such as medR or recall rate K (R@K) [49,50] as well as performance statistics (precision, recall, accuracy, F-score) at different intervals close to the target (Precision@K, Recall@K, Accuracy@K, F-Score@K) and overall Area Under ROC (AUC). The retrieval metric medR measures the median absolute interval difference between classes, namely, that for 10 classes and modifier GSD (30, 33.$\widehat{3}$, 36.$\widehat{6}$,..., 60), if the targets (modified) are 33.$\widehat{3}$ and predictions are 36.$\widehat{6}$ then there is a medR of 1.0, while if predictions are 60 then medR is 9.0. R@K measures the total recall (whether prediction in an interval distance from the target is lower than K) over a target window (i.e., if there are 40 classes and K is 10, only the 10 classes around the target label are considered for evaluation).

In Tables 7, 8 and 11 we add another quality metric in addition to the modifier-based ones, which is the *score*. For this *score* we defined a basis that describes the overall quality ranking (set from 0.0 to 1.0) of an image or dataset. This is calculated by measuring the weighted mean of the metrics, each metric with its own objective target (min↓ or max↑) as described in Table 2.

$$M_{score} = \frac{M_{range} - |(M_{objective} - M_{prediction})}{M_{range}} \tag{6}$$

$$score = \sum_{m=1}^{m=5} \omega_M^m M_{score}^m \tag{7}$$

For a specific quality metric we define the total M_{range} of the metric (i.e., for σ it would be $2.5 - 1.0$, namely, 1.5), an objective $M_{objective}$ value (i.e., for σ it would be the minimum, as best the *Quality* goes toward minimizing $\sigma \downarrow$, namely, 1.0) and the weights ω_M for the total weighted sum of the *Score* (by default if we keep the same importance for each metric, $\omega_M = \frac{1}{b}$, where m is the total number of modifiers, for our case $m = 5$).

4.1.2. Training and Validation

We trained our network with Inria-AILD sets of 180 images each for the training, validation and test subsets (Inria-AILD-180-train, Inria-AILD-180-val Inria-AILD-180-test, respectively), selecting 100 images for training and 20 for validation (proportional to 45% and 12% of the total, respectively). We processed all samples of the dataset with distinct intervals for each modifier (thus, we annotated each sample with that modification interval) and built our network with distinct heads: $N_\sigma = 50$, $N_F = 9$, $N_{GSD} = 10$, $N_{rer} = 40$, $N_{snr} = 40$. We selected a distinct set of crops for each resolution (C $\times$ R), in this case 10 crops of 1024 $\times$ 1024, 20 crops of 512 $\times$ 512, 50 crops of 256 $\times$ 256, 100 crops of 128 $\times$ 128 and 200 crops of 64 $\times$ 64. Thus, we generated datasets with different input resolutions but adapting the total domain capacity. The total number of trained images becomes 180x$N_{50,9,10,40,40}$x$C_{10,20,50,100,200}$ (e.g., a *blur* σ 64 $\times$ 64 image set contains 1.8M crop samples).

We ran our training and validation experiments for 200 epochs with distinct hyperparameters: $lr = [10^{-2}, 10^{-3}, 10^{-4}, 10^{-5}]$, $wd = [10^{-3}, 10^{-4}, 10^{-5}]$, momentum = 0.9 and soft threshold 0.3 (to filter out soft to hard/one-hot labels). Due to the computational capacity, the training batch sizes were selected according to the resolution for each set: $bs_{R=64\times64,128\times128} = [32, 64, 128, 256]$, $bs_{R=256\times256} = [16, 32, 64, 128]$, $bs_{R=512\times512} = [8, 16, 32, 64]$ and $bs_{R=1024\times1024} = [4, 8, 16, 32]$.

In Table 4 we show the validation results (Inria-AILD-180-test) with the trained QMR-Net using a ResNet18 backbone with the Inria-AILD-180-train data. We can observe that the overall medRs are around 1.0 (predictions are about one interval of distance with respect to the targets) and recall rates (exact match) are for top-1 (R@1) around 70% and for R@5 and R@10 (prediction is in an interval below 5 and 10 of the distance with respect to the target, respectively) around 100%. This means our network is able to predict the parameter data (*blur* σ, sharpness F, GSD, *snr*, *rer*) with a very high retrieval precision, even when the parameters are fine-grained (e.g., 40 or 50 class intervals). The best results appear for low N parameters (smaller classification tasks), such as F and GSD. Here, GSD is mostly an

easy task, as the scaling of objects is constant whether the images are distorted or not. In terms of the crop size, the best results are mostly in a higher input resolution of networks ($R = 1024 \times 1024$); this may vary on the selected backbone for the encoder (here, Resnet18 is mostly used with input R around 256×256).

In Table 5 are shown the validation results for multiparameter prediction, in which we tested multibranch QMRNet (QMRNet-MB) and multihead QMRNet (QMRNet-MH). Here, the performance between the two is similar to the single-parameter prediction (Table 4), where medR is around 1.0 and the recall rates around 70% for R@1 and >90% for R@5 and R@10. We tested predictions for two simultaneous parameters (*blur + rer*, *F + GSD* and *snr + rer*) and overall QMRNet-MB obtains better results for *blur + rer* and *snr + rer* but slightly worse for *F + GSD* than QMRNet-MH.

Table 4. Validation metrics for QMRNet (ResNet18) with all modifiers in Inria-AILD-180-test. Note that R (height × width) defines the resolution input of the network, in each case 1024 × 1024, 512 × 512, 256 × 256, 128 × 128 and 64 × 64. <u>Underline</u> represents top-1 best performance. *Italics* represents same value for most cases.

Parameter	R (H × W)	medR	R@1	R@5	R@10	F-Score	AUC
blur	64 × 64	2.170	38.37%	88.51%	97.96%	16.55%	59.03%
(N = 50)	128 × 128	1.021	64.42%	98.44%	99.85%	25.82%	62.20%
	256 × 256	0.936	73.05%	99.35%	99.91%	33.40%	66.04%
	512 × 512	0.989	70.27%	99.32%	*100.0*%	36.11%	72.18%
	1024 × 1024	<u>0.788</u>	<u>83.04%</u>	<u>99.65%</u>	*100.0*%	<u>42.56%</u>	<u>72.83%</u>
F	64 × 64	1.131	60.01%	99.25%	*100.0*%	31.30%	62.22%
(N = 9)	128 × 128	1.002	64.78%	<u>99.92%</u>	*100.0*%	33.73%	63.60%
	256 × 256	1.021	63.66%	99.54%	*100.0*%	35.22%	64.62%
	512 × 512	0.849	72.59%	99.76%	*100.0*%	40.56%	68.65%
	1024 × 1024	<u>0.643</u>	<u>80.28%</u>	99.85%	*100.0*%	<u>50.96%</u>	<u>75.45%</u>
GSD	64 × 64	*0.000*	*100.0*%	*100.0*%	*100.0*%	*100.0*%	*100.0*%
(N = 10)	128 × 128	*0.000*	*100.0*%	*100.0*%	*100.0*%	*100.0*%	*100.0*%
	256 × 256	*0.000*	*100.0*%	*100.0*%	*100.0*%	*100.0*%	*100.0*%
	512 × 512	*0.000*	*100.0*%	*100.0*%	*100.0*%	*100.0*%	*100.0*%
	1024 × 1024	*0.000*	*100.0*%	*100.0*%	*100.0*%	*100.0*%	*100.0*%
snr	64 × 64	1.374	51.44%	84.92%	97.97%	25.57%	63.06%
(N = 40)	128 × 128	1.396	52.97%	87.82%	98.35%	27.66%	64.75%
	256 × 256	1.113	62.65%	90.12%	97.25%	35.60%	68.93%
	512 × 512	1.073	68.30%	99.43%	*100.0*%	33.29%	67.50%
	1024 × 1024	<u>0.924</u>	<u>75.69%</u>	<u>99.95%</u>	*100.0*%	<u>35.93%</u>	<u>70.52%</u>
rer	64 × 64	1.512	49.90%	89.33%	98.84%	22.95%	62.06%
(N = 40)	128 × 128	5.319	18.79%	53.78%	77.79%	6.95%	52.28%
	256 × 256	1.328	52.91%	93.92%	99.64%	24.97%	63.68%
	512 × 512	1.268	57.71%	94.83%	99.76%	28.71%	<u>68.06%</u>
	1024 × 1024	<u>1.130</u>	<u>63.06%</u>	<u>96.53%</u>	99.98%	<u>28.88%</u>	65.00%

In Table 6 are shown the validation results for QMRNet's prediction of *snr* and *rer* in hyperspectral images (ECODSE RSData with 426 bands per pixel). By changing the first convolutional layer of QMRNet's encoder backbone for multiband input channels we can classify the quality metric with multichannel and hyperspectral images. Overall, medR is around 5.0 and the recall rates around 20% for R@1 (exact match), 56% for R@5 (five closest categories) and 80% for R@10 (ten closest categories). Here, the precision is lower due to the hardness of the approximation task, given the hyperspectral resolution (i.e., 80 × 80 at 60 cm/px) and the very few examples (43 examples). Despite the hardness of the dataset task, QMRNet with ResNet18 is able measure whether a parameter is in a specific range of *snr* or *rer* in hyperspectral images.

In Figure 4 we can see that most of the worst predictions for blur, sharpness, rer and snr appear mainly when attempting to predict over crops with sparse or homogeneous features, namely, when most of the image has limited or little pixel information (i.e., with similar pixel values), such as the sea or flat terrain surfaces. This is because the preprocessed samples have few or no dissimilarities in each modifier parameter. This has an effect on evaluating the datasets: when the surfaces are more sparse, predictions become harder.

Table 5. Validation metrics for multiparameter prediction with QMRNet-MH (multihead) and QMRNet-MB (multibranch) in Inria-AILD-180-test. Note: QMRNet-MB (one QMRNet branch per parameter) validation is equivalent to running several parameters from Figure 4 jointly. Underline represents top-1 best performance. *Italics* represents same value for most cases.

	Parameter	R (H × W)	medR	R@1	R@5	R@10	F-Score	AUC
QMRNet-MH	blur + rer (N = 50 + 40)	128 × 128	1.849	45.56%	89.64%	98.72%	20.35%	60.97%
		256 × 256	1.427	53.63%	95.69%	99.71%	25.49%	64.61%
		512 × 512	<u>1.365</u>	<u>57.70%</u>	<u>93.52%</u>	<u>95.65%</u>	<u>28.80%</u>	<u>65.33%</u>
	F + GSD (N = 9 + 10)	128 × 128	<u>0.055</u>	<u>98.39%</u>	<u>*100.0*%</u>	*100.0*%	<u>88.61%</u>	<u>95.47%</u>
		256 × 256	0.521	82.75%	99.93%	*100.0*%	66.17%	81.32%
		512 × 512	0.674	78.93%	99.42%	*100.0*%	64.28%	80.23%
	snr + rer (N = 40 + 40)	128 × 128	1.998	44.24%	86.56%	97.60%	20.90%	61.10%
		256 × 256	2.109	44.37%	85.33%	96.96%	20.88%	60.87%
		512 × 512	<u>1.588</u>	<u>52.55%</u>	<u>92.18%</u>	<u>98.85%</u>	<u>26.67%</u>	<u>65.17%</u>
QMRNet-MB	blur + rer (N = 50 + 40)	128 × 128	3.170	41.61%	76.11%	88.82%	16.39%	57.24%
		256 × 256	1.132	62.98%	96.64%	99.78%	29.19%	64.86%
		512 × 512	<u>1.128</u>	<u>63.99%</u>	<u>97.08%</u>	<u>99.88%</u>	<u>32.41%</u>	<u>70.12%</u>
	F + GSD (N = 9 + 10)	128 × 128	0.501	82.39%	99.96%	*100.0*%	66.87%	81.8%
		256 × 256	0.510	81.83%	99.77%	*100.0*%	67.61%	82.31%
		512 × 512	<u>0.424</u>	<u>86.30%</u>	<u>99.88%</u>	*100.0*%	<u>70.28%</u>	<u>84.33%</u>
	snr + rer (N = 40 + 40)	128 × 128	3.357	35.88%	70.80%	88.07%	17.31%	58.52%
		256 × 256	1.220	57.78%	92.02%	98.45%	30.29%	66.30%
		512 × 512	<u>1.170</u>	<u>63.01%</u>	<u>97.13%</u>	<u>99.88%</u>	<u>31.0%</u>	<u>67.78%</u>

Table 6. Validation metrics in ECODSE Competition hyperspectral image dataset with crops of 80 × 80 and 426 bands ranging from 383 to 2.512 nm with a spectral resolution of five nm.

	Parameter	R (H × W)	medR	R@1	R@5	R@10	F-Score	AUC
hs (C = 426)	rer (N = 40)	80 × 80	5.45	16.75%	51.69%	73.94%	6.41%	52.00%
	snr (N = 40)	80 × 80	4.70	18.75%	57.81%	89.45%	6.48%	52.04%

GT$_\sigma$ = 1.0/2.5
Pred$_\sigma$ = 1.0/2.5
rank(err) = 0/50

GT$_\sigma$ = 2.44/2.5
Pred$_\sigma$ = 2.44/2.5
rank(err) = 0/50

GT$_\sigma$ = 1.98/2.5
Pred$_\sigma$ = 1.34/2.5
rank(err) = 21/50

GT$_F$ = 1/10
Pred$_F$ = 1/10
rank(err) = 0/9

GT$_F$ = 10/10
Pred$_F$ = 10/10
rank(err) = 0/9

GT$_F$ = 10/10
Pred$_F$ = 3/10
rank(err) = 6/9

GT$_{rer}$ = 0.20/0.55
Pred$_{rer}$ = 0.20/0.55
rank(err) = 0/40

GT$_{rer}$ = 0.54/0.55
Pred$_{rer}$ = 0.054/0.55
rank(err) = 0/40

GT$_{rer}$ = 0.55/0.55
Pred$_{rer}$ = 0.37/0.55
rank(err) = 18/40

GT$_{snr}$ = 15/30
Pred$_{snr}$ = 15/30
rank(err) = 0/40

GT$_{snr}$ = 30/30
Pred$_{snr}$ = 30/30
rank(err) = 0/40

GT$_{snr}$ = 15/30
Pred$_{snr}$ = 30/30
rank(err) = 40/40

Figure 4. Correct and incorrect prediction examples of QMRNet on Inria-AILD-180 validation (crop resolution, i.e., R = 128 × 128) given interval rank error (classification label distance between GT and prediction, maximum is N for each net, i.e., 50 for blur, 10 for sharpness and 40 for snr and rer). medR is the overall median rank (err).

4.2. Results on QMRNet for IQA: Benchmarking Image Datasets

We ran our QMRNet with a ResNet18 backbone over the sets (See the EO dataset evaluation use case in https://github.com/dberga/iquaflow-qmr-eo, accessed on 10 October 2022) described in Table 1. Given our network trained uniquely on Inria-AILD-180-train, we see how our network is able to adapt due to the prediction of feasible quality metrics (*blur* σ, GSD, sharpness *F*, *snr* and *rer*) over each of the distinct datasets. We see that with fine-tuning QMRNet over Inria-AILD-180-train, the overall σ for most of the datasets appears to be $\sigma = 1.0$ (originally unblurred from the ground truth) except for USGS279 and Inria-AILD-180-test, which is around $\sigma = 1.02$. For the case of the sharpness factor *F*, the overall values for most datasets is $F = 1.0$ (without oversharpening) but for cases such as UCMerced380 and Shipnset, it appears to be oversharpened ($F > 1.5$ and $F > 3.0$, respectively). Most datasets present an overall predicted *snr* of $M(snr) = 28.67$ and *rer* of $M(rer) = 0.4896$. The highest *Score* datasets are Inria-AILD-180-train, UCMerced2100 and USGS279, here considering the same weight ω_M for each modifier metric *M*.

Table 7. Mean IQA results of datasets given QMRNet(ResNet18) trained over 180 images (Inria-AILD-180-train) and 5 modifiers. <u>Underline</u> represents top-1 best performance. *Italics* represents same value for most cases.

Dataset	blur↓	snr↑↓	rer↑↓	F↑↓	GSD↓	Score↑
USGS	1.019	26.111	0.467	*1.000*	*0.300*	0.896
UCMerced-380	*1.000*	28.121	0.470	1.563	*0.300*	0.878
UCMerced-2100	*1.000*	24.994	0.459	1.194	*0.300*	0.896
Inria-AILD-180-test	1.021	<u>*30.0*</u>	0.488	*1.000*	*0.300*	0.887
Inria-AILD-180-train	*1.000*	<u>*30.0*</u>	<u>0.515</u>	*1.000*	*0.300*	<u>0.904</u>
Shipsnet-Ships	*1.000*	27.516	0.483	1.281	*0.300*	0.881
shipsnet-Scenes	*1.000*	<u>*30.00*</u>	0.499	3.250	*0.300*	0.846
DeepGlobe	*1.000*	<u>*30.0*</u>	0.505	1.281	*0.300*	0.892
XView-train	*1.000*	<u>*30.0*</u>	0.507	*1.000*	*0.300*	0.899
XView-validation	*1.000*	<u>*30.0*</u>	0.503	*1.000*	*0.300*	0.898

4.3. Results on QMRNet for IQA: Benchmarking Image Super-Resolution

Here we selected a set of super-resolution algorithms that have been previously tested to super-resolve high-quality real-image SR dataset benchmarks such as BSD100, Urban100 or Set5 × 2 (https://paperswithcode.com/task/image-super-resolution, accessed on 10 October 2022) but here we want to apply them to EO data and metrics. For this we want to benchmark their performance considering full-reference, no-reference and our QMRNet-based metric (See the use case of super-resolution benchmark at https://github.com/dberga/iquaflow-qmr-sisr, accessed on 10 October 2022). QMRNet allows us to check the amount of each distortion for every transformation (LR) applied to the original image (HR), if it is either the usual x2, x3 or x4 downsampling or a specific distortion such as blurring.

Concretely, we tested our UCMerced subset of 380 images with crops of 256 × 256 with autoencoder algorithms (FSRCNN and MSRN) and GAN-based and self-supervised architectures such as SRGAN, ESRGAN, CAR and LIIF. All model checkpoints are selected as vanilla (default hyperparameter settings) except for the input scaling (x2, x3, x4) and also for the case of MSRN, for which we computed three versions of the vanilla MSRN (architecture with four scales), one without fine-tuning ($MSRN_1$), one with fine-tuning and added noise ($MSRN_2$) and one ($MSRN_3$) with fine-tuning (over Inria-AILD-180-train).

In Table 8 we have evaluated each type of modifier parameter for every single super-resolution algorithm as well as the overall score for all quality metric regression. Here, we tested the algorithms considering x2, x3 and x4 downsampling input ($LR_{x2,x3,x4}$), as well as considering the case of adding a blur filter with a scaled σ. Here, the QMRNet is able to predict that I_{LR} gives the worst ranking for most metrics. FSRCNN and SRGAN

give similar results in most metrics, with SRGAN being slightly better in the *blur* and *snr* metrics. MSRN shows the best results in *snr* and *F*, mainly when the inputs have a higher resolution (i.e., x2, x3). For the overall scores, CAR presents the best results in *blur* and *rer*, with the highest score ranking in most downsampling cases. However, CAR has the worst ranking in the noise and sharpness metrics (*snr* and *F*). As we mentioned earlier, CAR presents oversharpening and hallucinations, which can trick some metrics that measure blur but becomes worse for those that predict unusual signal-to-noise ratios and illusory edges. In contrast, LIIF presents a bad performance in the *blur* and *rer* metrics (meaning LIIF's I_{SR} images appearing slightly blurred), although LIIF acquires a overall good performance for the rest of the modifier metrics. We want to pinpoint that in some metrics (i.e., *snr*, *rer* and *F*), some of the tested algorithms (Tables 8–10) show lower distortion values than the original I_{HR}. This phenomenon means that our metrics can demonstrate if an image has a present distortion effect (whether it is oversharpening, blur or noise) beyond its image quality, unlike full-reference metrics, which are limited to the quality of the I_{HR} samples.

In Table 9 we show a benchmark of known full-reference metrics. In super-resolving x2, MSRN (concretely, $MSRN_2$ and $MSRN_3$) has the best results for full-reference metrics, including SSIM, PSNR, SWD, FID, MSSIM, HAARPSI and MDSI. In x3 and x4, LIIF and CAR have the best results for most of these metrics, including PSNR, FID, GMSD and MDSI, being top-3 with most metric evaluations. Here we have to pinpoint that LIIF does not perform as well when the input (I_{LR}) has been blurred; see here that CAR is able to deblur the input better than other algorithms as it is oversharpening the originally downscaled and/or blurred I_{LR}. In Table 10 we show the no-reference metric results, here for SNR, RER, MTF and FWHM. SRGAN, MSRN and LIIF present significantly better results for SNR than other algorithms. This means these algorithms in general do not add noise to the input, namely, the generated images do not contain artifacts that were not present in the original I_{HR}. In this case, CAR outperforms in RER, MTF and FWHM.

In Figures 5 and 6 we super-resolve the original UCMerced and XView images x3 and we can observe that some algorithms, such as FSRCNN, SRGAN, $MSRN_1$, ESRGAN and LIIF, present a similar (blurred) output, while others, such as $MSRN_2$, $MSRN_3$ and CAR, present a higher noise and oversharpening of borders, trying to enhance the features of the image (here, attempting to generate features beyond the I_{LR} content). The noise and oversharpening are distinguishable in colormaps of buildings (e.g., Figure 5, row 10 and Figure 7, row 6).

In our results for low-resolution LR_{x3} inputs we can qualitatively see (Figure 7) that FSRCNN, SRGAN, $MSRN_1$ and LIIF present blurred outputs, similar to the I_{LR}. ESRGAN does not change much the appearance with respect to the original image (see differences in colormaps), but simply adds some residual noise at the edges. CAR, however, seems to acquire better results but it appears in some cases to be oversharpened (similar to MSRN$_3$). We can observe that MSRN algorithms do not perform well when super-resolving very-low-resolution images (i.e., the downsampled I_{LR}). Its original training set might not have considered very-low-resolution image samples. See Section 4.4 for MSRN optimization using QMRNet.

Table 8. Mean no-reference Quality Metric Regression (QMRNet trained on Inria-AILD-180-train) metrics on super-resolution of downsampled inputs in UCMerced-380. **Bold** represents having lower distortion than HR. <u>Underline</u> represents top-1 best performance. *Italics* represents the same value for most cases.

	Algorithm	blur↓	snr↑↓	rer↑↓	F↑↓	GSD↓	Score↑
	HR	*1.000*	*28.121*	*0.470*	*1.563*	*0.300*	*0.878*
x2	LR$_{x2}$	1.103	**28.997**	0.366	<u>***1.000***</u>	*0.300*	0.820
	FSRCNN	<u>*1.000*</u>	<u>***30.0***</u>	**0.490**	2.699	*0.300*	0.853
	SRGAN	<u>*1.000*</u>	<u>***30.0***</u>	0.411	**1.160**	*0.300*	0.848
	MSRN$_1$	1.141	**29.12**	0.344	<u>***1.000***</u>	*0.300*	0.804
	MSRN$_2$	1.036	**28.69**	0.431	**1.018**	*0.300*	0.863
	MSRN$_3$	1.109	<u>***30.0***</u>	0.341	<u>***1.000***</u>	*0.300*	0.802
	ESRGAN	1.084	**28.874**	0.358	<u>***1.000***</u>	*0.300*	0.820
	CAR	<u>*1.000*</u>	26.061	**0.499**	2.776	*0.300*	<u>0.876</u>
	LIIF	1.089	**29.558**	<u>0.348</u>	<u>***1.000***</u>	*0.300*	0.810
x3	LR$_{x3}$	1.149	**29.937**	0.274	<u>***1.000***</u>	*0.300*	0.763
	FSRCNN	1.114	**29.937**	0.323	<u>***1.000***</u>	*0.300*	0.793
	SRGAN	1.074	<u>***30.0***</u>	0.347	<u>***1.000***</u>	*0.300*	0.809
	MSRN$_1$	1.142	<u>***30.0***</u>	0.277	<u>***1.000***</u>	*0.300*	0.765
	MSRN$_2$	1.025	<u>***30.0***</u>	0.310	<u>***1.000***</u>	*0.300*	0.798
	MSRN$_3$	1.034	<u>***30.0***</u>	0.310	<u>***1.000***</u>	*0.300*	0.796
	ESRGAN	1.332	**29.561**	0.309	**1.030**	*0.300*	0.758
	CAR	<u>*1.000*</u>	28.145	<u>0.420</u>	**1.071**	*0.300*	<u>0.864</u>
	LIIF	1.089	**29.558**	<u>0.348</u>	<u>***1.000***</u>	*0.300*	0.810
x4	LR$_{x4}$	1.620	<u>***30.0***</u>	0.202	<u>***1.000***</u>	*0.300*	0.664
	FSRCNN	1.563	**29.937**	0.287	<u>***1.000***</u>	*0.300*	0.715
	SRGAN	1.368	<u>***30.0***</u>	0.290	<u>***1.000***</u>	*0.300*	0.741
	MSRN$_1$	1.582	<u>***30.0***</u>	0.206	<u>***1.000***</u>	*0.300*	0.672
	MSRN$_2$	1.505	<u>***30.0***</u>	0.185	<u>***1.000***</u>	*0.300*	0.671
	MSRN$_3$	1.484	<u>***30.0***</u>	0.231	<u>***1.000***</u>	*0.300*	0.697
	ESRGAN	1.332	**29.561**	0.309	**1.030**	*0.300*	0.758
	CAR	<u>1.039</u>	<u>***30.0***</u>	<u>0.371</u>	<u>***1.000***</u>	*0.300*	<u>0.826</u>
	LIIF	1.467	**29.495**	0.293	<u>***1.000***</u>	*0.300*	0.733
	Algorithm	blur↓	snr↑↓	rer↑↓	F↑↓	GSD↓	Score↑
	HR	*1.000*	*28.121*	*0.470*	*1.563*	*0.300*	*0.878*
x2 + blur	LR$_{x2+blur}$	1.444	**29.684**	0.285	<u>***1.000***</u>	*0.300*	0.731
	FSRCNN	<u>1.002</u>	<u>***30.0***</u>	**0.479**	**1.524**	*0.300*	<u>0.873</u>
	SRGAN	1.076	<u>***30.0***</u>	0.338	<u>***1.000***</u>	*0.300*	0.805
	MSRN$_1$	1.473	**29.75**	0.274	<u>***1.000***</u>	*0.300*	0.721
	MSRN$_2$	1.434	**29.62**	0.286	<u>***1.000***</u>	*0.300*	0.733
	MSRN$_3$	1.434	<u>***30.0***</u>	0.279	<u>***1.000***</u>	*0.300*	0.728
	ESRGAN	1.208	<u>***30.0***</u>	0.282	<u>***1.000***</u>	*0.300*	0.759
	CAR	1.013	**28.750**	0.382	**1.071**	*0.300*	0.840
	LIIF	1.568	<u>***30.0***</u>	0.237	<u>***1.000***</u>	*0.300*	0.689
x3 + blur	LR$_{x3+blur}$	2.420	<u>***30.0***</u>	0.198	**1.000**	*0.300*	0.556
	FSRCNN	1.649	<u>***30.0***</u>	0.229	**1.000**	*0.300*	0.674
	SRGAN	1.273	<u>***30.0***</u>	0.243	**1.000**	*0.300*	0.731
	MSRN$_1$	2.339	<u>***30.0***</u>	0.198	**1.000**	*0.300*	0.566
	MSRN$_2$	2.324	<u>***30.0***</u>	0.178	**1.000**	*0.300*	0.559
	MSRN$_3$	2.244	<u>***30.0***</u>	0.210	**1.000**	*0.300*	0.586
	ESRGAN	1.559	<u>***30.0***</u>	0.242	**1.000**	*0.300*	0.692
	CAR	<u>1.116</u>	**29.937**	<u>0.312</u>	**1.000**	*0.300*	<u>0.787</u>
	LIIF	1.725	<u>***30.0***</u>	0.228	**1.000**	*0.300*	0.663
x4 + blur	LR$_{x4+blur}$	1.840	*30.0*	0.159	*1.000*	*0.300*	0.613
	FSRCNN	1.649	*30.0*	0.229	*1.000*	*0.300*	0.674
	SRGAN	1.625	*30.0*	0.175	*1.000*	*0.300*	0.650
	MSRN$_1$	1.696	*30.0*	0.161	*1.000*	*0.300*	0.633
	MSRN$_2$	1.606	*30.0*	0.155	*1.000*	*0.300*	0.642
	MSRN$_3$	1.630	*30.0*	0.168	*1.000*	*0.300*	0.645
	ESRGAN	1.559	*30.0*	0.242	*1.000*	*0.300*	0.692
	CAR	<u>1.329</u>	*30.0*	<u>0.258</u>	*1.000*	*0.300*	<u>0.731</u>
	LIIF	1.725	*30.0*	0.228	*1.000*	*0.300*	0.663

Table 9. Mean full-reference metrics on super-resolution of downsampled inputs in UCMerced-380. Underline represents top-1 best performance.

Group	Algorithm	ssim↑	psnr↑	swd↓	fid↓	mssim↑	haarpsi↑	gmsd↓	mdsi↑
	HR	*1.00*	*80.000*	-	-	*1.00*	*1.00*	-	-
x2	LR_{x2}	0.901	30.628	1125	0.211	0.990	0.954	0.014	0.330
	FSRCNN	0.438	16.682	2316	4.47	0.718	0.552	0.155	0.427
	SRGAN	0.919	31.534	1010	0.177	0.991	0.925	0.015	0.308
	$MSRN_1$	0.901	30.178	1103	0.222	0.990	0.950	0.014	0.329
	$MSRN_2$	0.917	31.750	1017	0.174	0.991	0.951	0.013	0.315
	$MSRN_3$	0.892	30.417	1167	0.217	0.987	0.934	0.016	0.339
	ESRGAN	0.793	26.693	1462	0.353	0.959	0.737	0.073	0.370
	CAR	0.827	26.285	1282	0.422	0.968	0.831	0.064	0.354
	LIIF	0.860	29.645	1236	0.220	0.978	0.892	0.036	0.360
x3	LR_{x3}	0.778	27.004	1619	0.386	0.956	0.801	0.072	0.401
	FSRCNN	0.839	28.982	1328	0.243	0.973	0.865	0.042	0.367
	SRGAN	0.811	27.633	1456	0.332	0.961	0.796	0.053	0.386
	$MSRN_1$	0.700	24.368	1864	0.502	0.918	0.666	0.128	0.420
	$MSRN_2$	0.699	24.169	1800	0.513	0.918	0.663	0.128	0.415
	$MSRN_3$	0.701	24.261	1838	0.488	0.918	0.662	0.128	0.418
	ESRGAN	0.825	28.387	1371	0.262	0.970	0.848	0.049	0.366
	CAR	0.721	23.273	1678	0.708	0.925	0.700	0.111	0.394
	LIIF	0.860	29.645	1245	0.220	0.978	0.892	0.036	0.360
x4	LR_{x4}	0.683	25.031	1973	0.569	0.925	0.703	0.121	0.440
	FSRCNN	0.819	28.223	1401	0.278	0.969	0.843	0.050	0.372
	SRGAN	0.721	25.844	1750	0.468	0.936	0.716	0.096	0.428
	$MSRN_1$	0.600	22.691	2142	0.743	0.869	0.573	0.164	0.453
	$MSRN_2$	0.599	22.582	2094	0.752	0.870	0.570	0.164	0.451
	$MSRN_3$	0.602	22.651	2156	0.726	0.871	0.569	0.165	0.454
	ESRGAN	0.825	28.387	1349	0.262	0.970	0.848	0.049	0.366
	CAR	0.624	21.825	1953	0.910	0.887	0.620	0.150	0.421
	LIIF	0.841	28.708	1316	0.254	0.974	0.866	0.043	0.367

Group	Algorithm	ssim↑	psnr↑	swd↓	fid↓	mssim↑	haarpsi↑	gmsd↓	mdsi↑
	HR	*1.00*	*80.000*	-	-	*1.00*	*1.00*	-	-
x2 + blur	$LR_{x2+blur}$	0.822	27.876	1504	0.356	0.968	0.854	0.051	0.385
	FSRCNN	0.372	16.425	2495	4.89	0.662	0.502	0.184	0.447
	SRGAN	0.836	28.135	1398	0.349	0.966	0.826	0.052	0.376
	$MSRN_1$	0.825	27.574	1485	0.377	0.968	0.855	0.049	0.383
	$MSRN_2$	0.846	28.637	1409	0.307	0.972	0.867	0.045	0.372
	$MSRN_3$	0.817	27.852	1529	0.355	0.965	0.840	0.053	0.389
	ESRGAN	0.774	26.754	1657	0.401	0.955	0.738	0.075	0.404
	CAR	0.903	30.716	1156	0.197	0.984	0.915	0.034	0.326
	LIIF	0.748	26.312	1769	0.508	0.939	0.774	0.088	0.422
x3 + blur	$LR_{x3+blur}$	0.691	25.054	2003	0.614	0.918	0.716	0.115	0.444
	FSCNN	0.741	26.107	1804	0.513	0.938	0.764	0.088	0.423
	SRGAN	0.705	25.089	1911	0.637	0.915	0.703	0.107	0.443
	$MSRN_1$	0.645	23.803	2131	0.706	0.892	0.639	0.143	0.455
	$MSRN_2$	0.649	23.731	2050	0.714	0.895	0.641	0.141	0.450
	$MSRN_3$	0.649	23.832	2113	0.681	0.894	0.639	0.142	0.454
	ESRGAN	0.752	26.314	1770	0.488	0.941	0.770	0.085	0.419
	CAR	0.783	26.909	1616	0.378	0.955	0.801	0.070	0.405
	LIIF	0.748	26.337	1798	0.500	0.939	0.777	0.086	0.421
x4 + blur	$LR_{x4+blur}$	0.972	38.599	897	0.046	0.992	0.940	0.031	0.248
	FSRCNN	0.977	37.210	834	0.062	0.992	0.950	0.022	0.226
	SRGAN	0.962	34.761	1050	0.083	0.986	0.867	0.033	0.265
	$MSRN_1$	0.909	30.115	1277	0.112	0.955	0.756	0.095	0.316
	$MSRN_2$	0.901	29.513	1350	0.150	0.955	0.750	0.096	0.317
	$MSRN_3$	0.909	29.888	1281	0.120	0.955	0.749	0.096	0.319
	ESRGAN	0.973	37.202	876	0.062	0.992	0.945	0.024	0.236
	CAR	0.916	30.067	1371	0.213	0.964	0.831	0.074	0.309
	LIIF	0.994	47.317	420	0.032	0.999	0.993	0.003	0.166

Table 10. Mean no-reference noise (SNR) and contour sharpness (RER, MTF, FWHM) metrics on super-resolution of downsampled inputs in UCMerced-380. <u>Underline</u> represents top-1 best performance.

|	Algorithm	SNR$_{Mdn}$ ↓	SNR$_M$ ↓	RER$_{XY}$ ↑	MTF$_{XY}$ ↑	FWHM$_{XY}$ ↓
	HR	*20.788*	*28.814*	*503.5*	*124.5*	*1692*
x2	LR$_{x2}$	31.361	43.217	367.5	30	2379.5
	FSRCNN	**<u>10.830</u>**	**<u>11.016</u>**	471.5	437	3038.5
	SRGAN	28.699	35.223	497	119.5	1730
	MSRN$_1$	33.188	45.941	356	24.5	2450
	MSRN$_2$	30.114	40.626	376	35	2329.5
	MSRN$_3$	34.217	43.851	367.5	29.5	2374
	ESRGAN	23.916	31.614	382	35	2269
	CAR	**15.660**	**26.506**	<u>553</u>	<u>166</u>	<u>1484</u>
	LIIF	44.273	56.133	459.5	92	1909
x3	LR$_{x3}$	45.33	54.317	317.5	93	2754
	FSRCNN	39.72	45.69	222.5	**191**	2132
	SRGAN	43.75	49.17	432.5	**187**	2015.5
	MSRN$_1$	43.882	52.050	321.5	15.5	2743.5
	MSRN$_2$	37.707	46.747	340.5	19	2571
	MSRN$_3$	44.579	52.747	345.5	20.5	2532.5
	ESRGAN	28.58	39.97	340	115	2562.5
	CAR	<u>25.20</u>	<u>39.45</u>	**<u>522.5</u>**	**<u>261.5</u>**	**<u>1617</u>**
	LIIF	44.27	56.13	460.5	252	1903.5
x4	LR$_{x4}$	49.183	57.351	279	6	3150
	FSRCNN	30.797	41.584	325.5	14	2678
	SRGAN	50.258	55.282	366	28.5	2385.5
	MSRN$_1$	51.875	60.043	281	6.5	3113
	MSRN$_2$	45.084	52.373	293	8	2987
	MSRN$_3$	53.523	61.691	298	8	2936
	ESRGAN	28.584	<u>39.974</u>	340	18.5	2560
	CAR	<u>30.193</u>	47.106	<u>485.5</u>	<u>113.5</u>	<u>1793</u>
	LIIF	35.375	49.543	342	10	2546
	Algorithm	**SNR$_{Mdn}$ ↓**	**SNR$_M$ ↓**	**RER$_{XY}$ ↑**	**MTF$_{XY}$ ↑**	**FWHM$_{XY}$ ↓**
	HR	*20.788*	*28.814*	*503.5*	*124.5*	*1692*
x2 + blur	LR$_{x2+blur}$	40.864	49	299	13	2804.5
	FSRCNN	**<u>11.314</u>**	**<u>11.529</u>**	289.5	8.5	3046.5
	SRGAN	41.630	49.499	400	56	2258
	MSRN$_1$	43.346	53.858	306	10.5	2865.5
	MSRN$_2$	39.287	52.993	317.5	14	2766
	MSRN$_3$	44.007	53.984	314.5	12.5	2791
	ESRGAN	42.656	55.710	318	13	2770.5
	CAR	33.737	47.754	<u>446.5</u>	<u>76.5</u>	<u>1939</u>
	LIIF	58.289	73.030	298	11.5	2975
x3 + blur	LR$_{x3+blur}$	57.193	65.361	287.5	5	3107.5
	FSRCNN	52.598	66.357	285.5	8	3083.5
	SRGAN	55.658	64.598	354.5	27.5	2515
	MSRN$_1$	54.601	62.769	290.5	11.5	3076.5
	MSRN$_2$	<u>50.257</u>	<u>60.81</u>	297	12	2997.5
	MSRN$_3$	58.377	66.545	302	13	2954.5
	ESRGAN	51.330	66.647	291	10	3036
	CAR	50.696	66.709	<u>398.5</u>	<u>48</u>	<u>2209.5</u>
	LIIF	56.194	64.362	283	7	3119.5
x4 + blur	LR$_{x4+blur}$	65.089	73.257	268.5	7.5	3311.5
	FSRCNN	<u>53.430</u>	68.246	290	8	3038.5
	SGAN	62.236	70.854	316.5	14	2806
	MSRN$_1$	63.810	71.978	279.5	13	3579.5
	MSRN$_2$	54.682	63.786	282.5	8.5	3445
	MSRN$_3$	70.048	78.216	288.5	9.5	3308
	ESRGAN	53.559	67.793	292	4	3011.5
	CAR	60.483	83.280	<u>359.5</u>	<u>30</u>	<u>2471</u>
	LIIF	56.194	<u>64.362</u>	282.5	6.5	3120
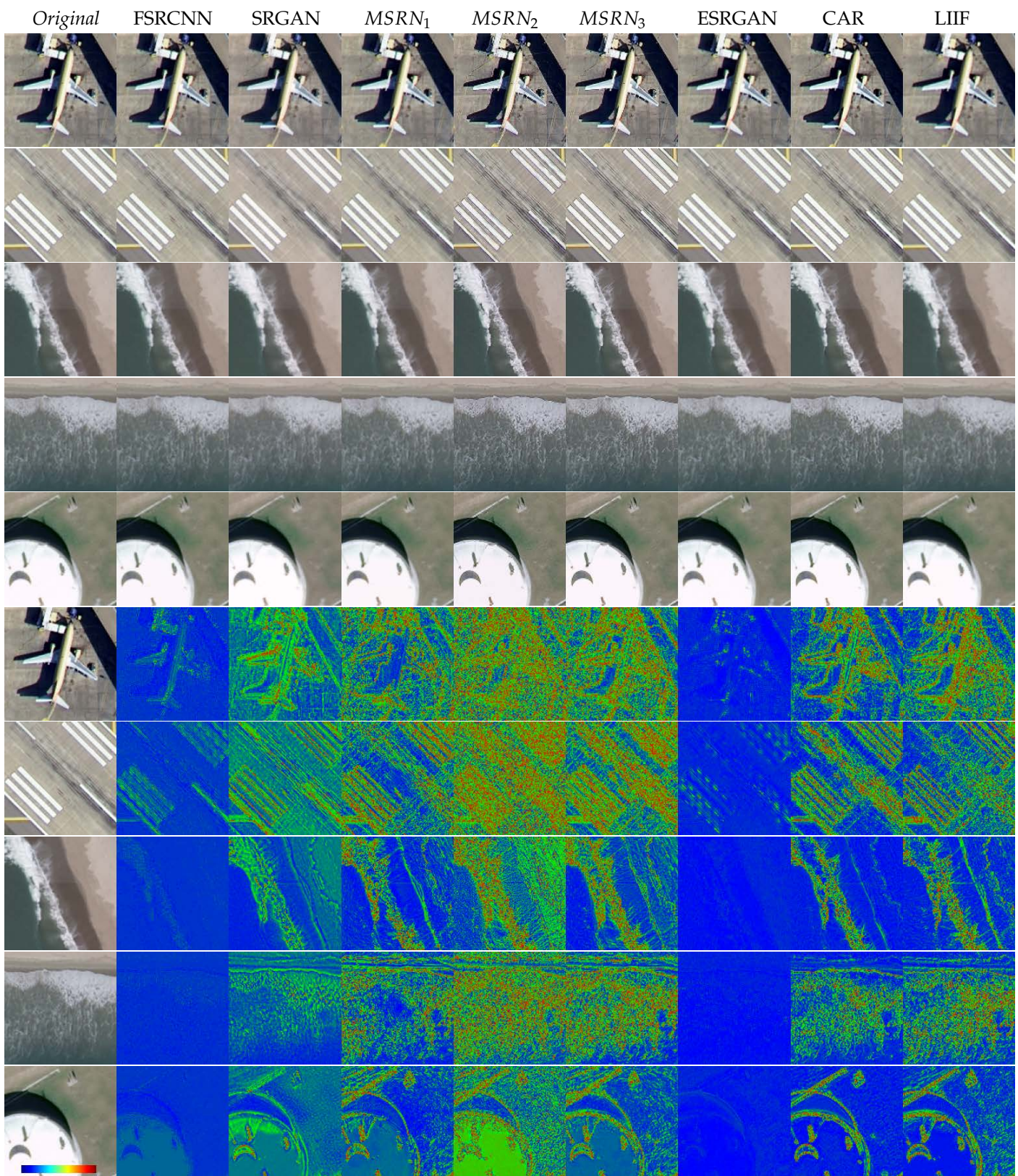
Figure 5. Examples of super-resolving original UCMerced images (with a crop zoom, i.e., 128×128) and each SR algorithm output. Inputs (*Original* I_{HR}) without downsampling (i.e., super-resolving x3). In rows 6–10 are colormaps of the sum of differences ($\Delta R + \Delta G + \Delta B$) with respect to *Original* I_{HR}.

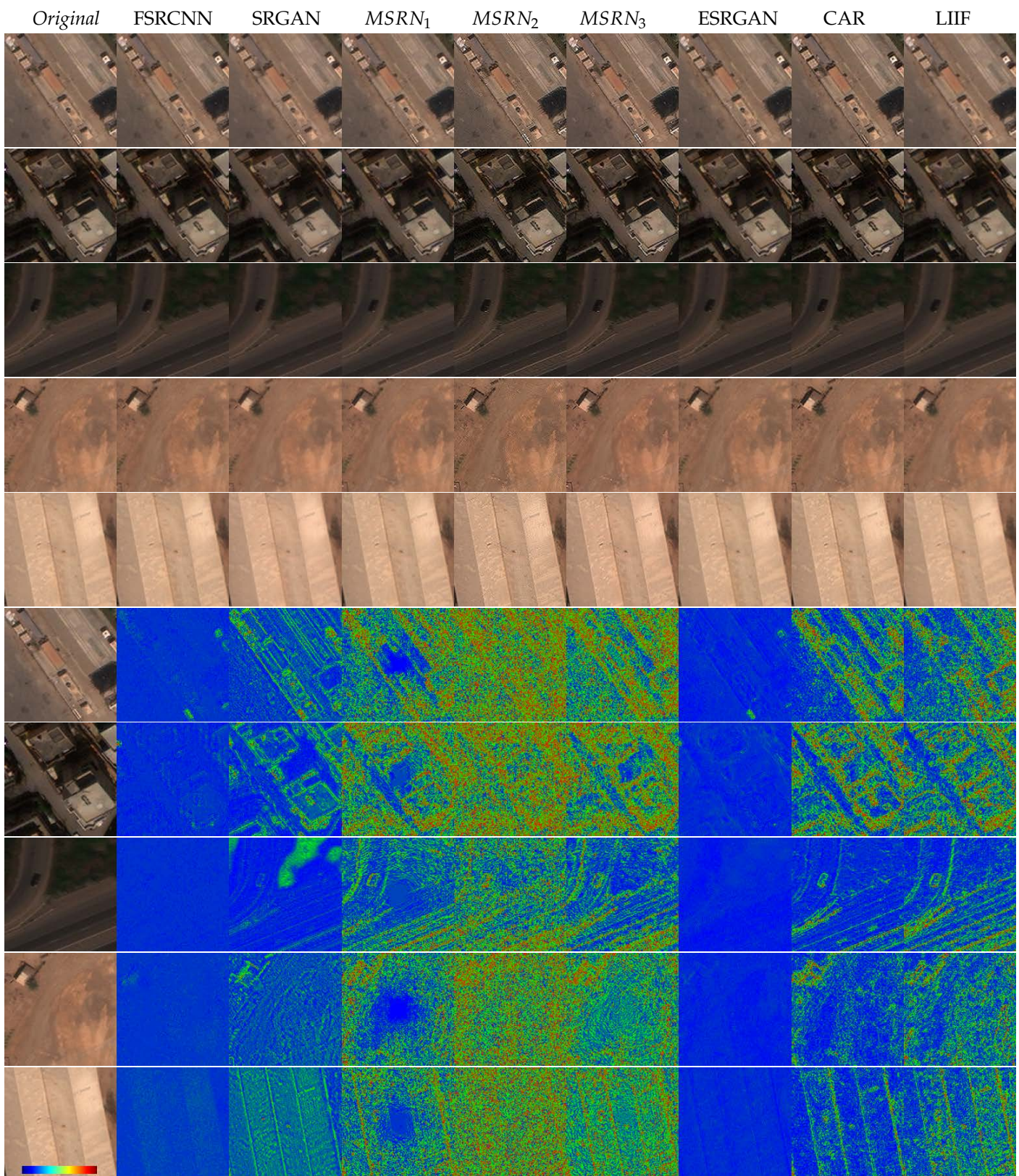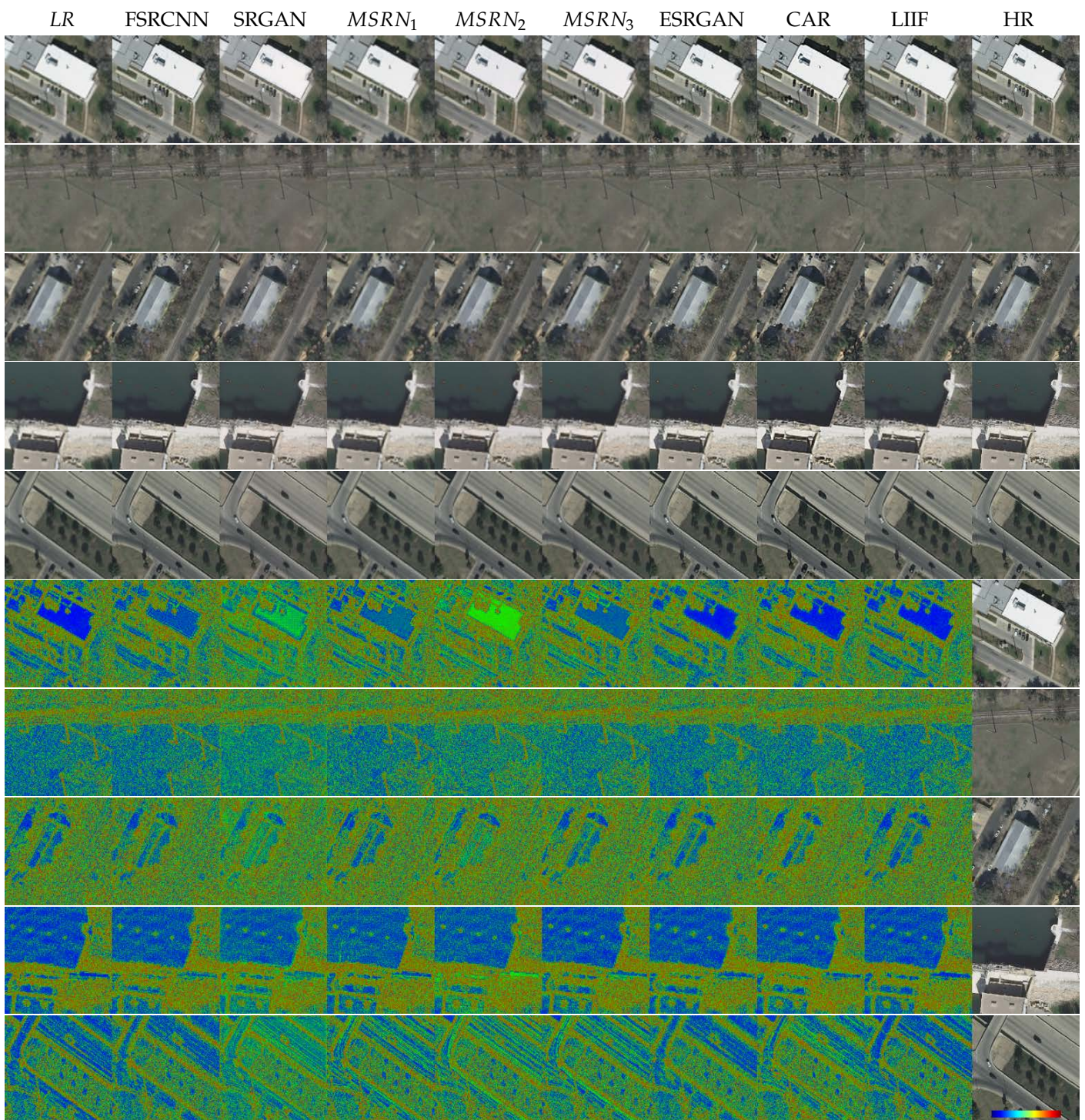**Figure 6.** Examples of super-resolving original XView images (with a crop zoom, i.e., 128 × 128) and each SR algorithm output. Inputs (*Original I_{HR}*) without downsampling (i.e., super-resolving x4). In rows 6–10 are colormaps of the sum of differences (ΔR + ΔG + ΔB) with respect to *Original I_{HR}*.

Figure 7. Low-resolution examples of Inria-AILD-180-val images (crops of 256 × 256) and each SR algorithm output. *LR* (corresponding to input on algorithms I_{LR}) is the downsampling x3 of I_{HR}. In rows 6–10 are colormaps of the sum of differences (ΔR + ΔG + ΔB) with respect to *Original* I_{HR}.

Above (Figure 8) we demonstrate the validity of some of our metric results by comparing them with each homologous measurement, namely, the ones measuring similar or the same properties. Here, we compared QMRNet's *snr* ↓ and PSNR↑. These measure the quantity of noise over information of the image. The first subplot shows an anticorrelation ($\nearrow$) on the algorithm values in these two metrics, with LIIF being closest to the I_{HR} (GT) and CAR, $MSRN_2$ and $MSRN_3$ having both the lowest *snr* (best) and PSNR (worst). For the case of QMRNet's *rer* ↑ and measured RER_{other} ↑ (which corresponds to the RER that

measures diagonal contours), there is a positive correlation ($\nearrow$), with CAR, $MSRN_2$ and $MSRN_3$ outperforming the rest of the algorithms. We also compared $FWHM_{other} \downarrow$ and SSIM$\uparrow$ to see how well each algorithm performs when evaluating the diagonal contour width as well as the structural similarity, and it appears that $MSRN_2$, $MSRN_3$ and CAR have the lowest (best) $FWHM$ and most algorithms have the same values of SSIM as the original GT images (unchanged). In the last subplot we compared the QMRNet's *score* $\uparrow$ (composed of the weighted mean of QMRNet's *sigma*, *rer*, *snr*, *GSD* and *F*) and FID$\downarrow$, which measures the Fréchet distribution distance between images. Here $MSRN_2$, $MSRN_3$ and CAR show the highest *score* with higher (worse) FID, while most algorithms are close to the original image (almost unchanged). Note that in these plots we super-resolve x4 the original image so that full-reference metrics can only compare with the original image (thus, there is no downsampling of inputs so that the I_{HR} would be equivalent to the I_{LR} input). Here, we need to consider how the algorithms actually perform in metrics that can evaluate better than the original image.
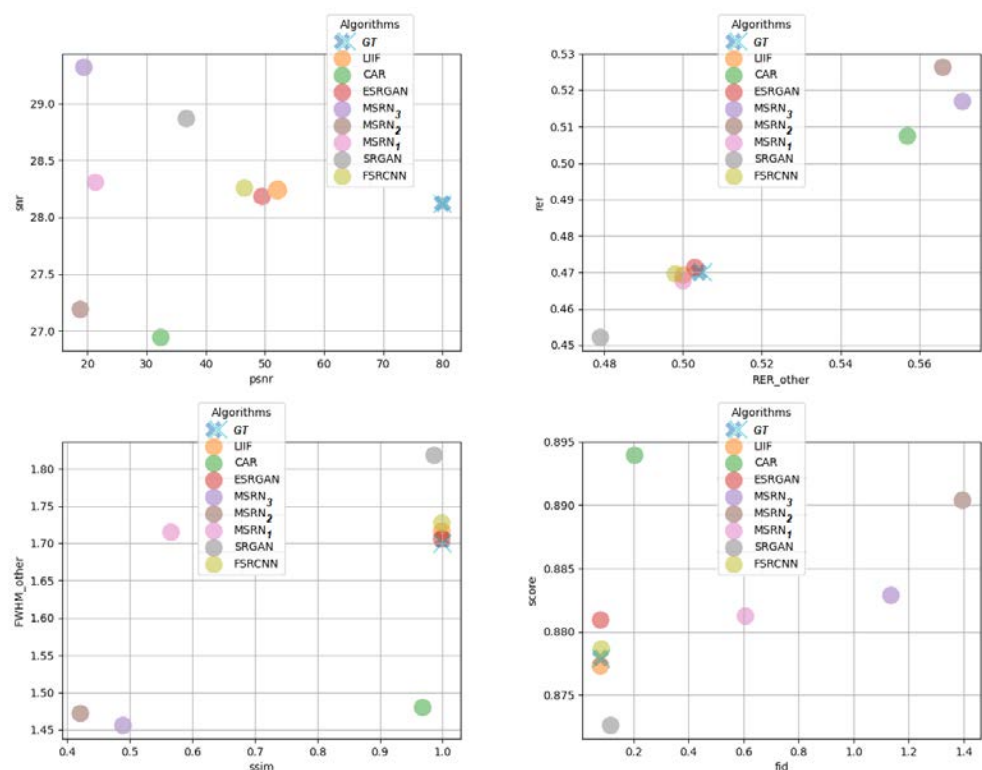

Figure 8. Scatter plots of metric comparison on super-resolving (x4) UCMerced dataset.

4.4. Results on QMRloss: Optimizing Image Super-Resolution

In this section, we integrated the aforementioned QMRLoss as an ad-hoc regularization strategy for optimizing SR algorithms (See the QMRLoss optimization use case at https://github.com/dberga/iquaflow-qmr-loss, accessed on 10 October 2022). For this case, we integrated different loss methods (L1, L2 and BCE) as QMRLoss in different modifiers in MSRN training. We regularized the MSRN architecture by integrating the QMRLoss ($\mathcal{L}_{QMR}$) to the total loss calculation, namely, summed to the adversarial loss $\mathcal{L}_{adv}$ and the perceptual loss $\mathcal{L}_{perc}$ (in this case, VGGLoss). This QMRLoss regularization mechanism will allow MSRN and any other algorithm to avoid quality mismatches considering several metrics that measure distortions simultaneously (see results in Table 11).

In Figure 9 we show that several strategies such as QMRLoss using *rer* (and L1 loss) obtain better results than vanilla MSRN in the PSNR, SSIM and FID metrics. Here, the PSNR improves with QMRNet using L1 loss and crops of 256 × 256 as well as with L2 loss with 512 × 512. It also improves with the *blur* metric both with L1 and L2 loss on 256 × 256 crops. The SSIM improves with L1 loss in QMRNet that uses RER and significatively (almost 1.0)

with *rer* L2 loss with crops of 512×512. For FID, using QMRNet improves MSRN with *rer* and all types of losses (L1, L2 and BCE) using crops of 64×64, here as well using QMRNet with *snr* metric and L1 loss, using crops of 64×64 and 128×128.
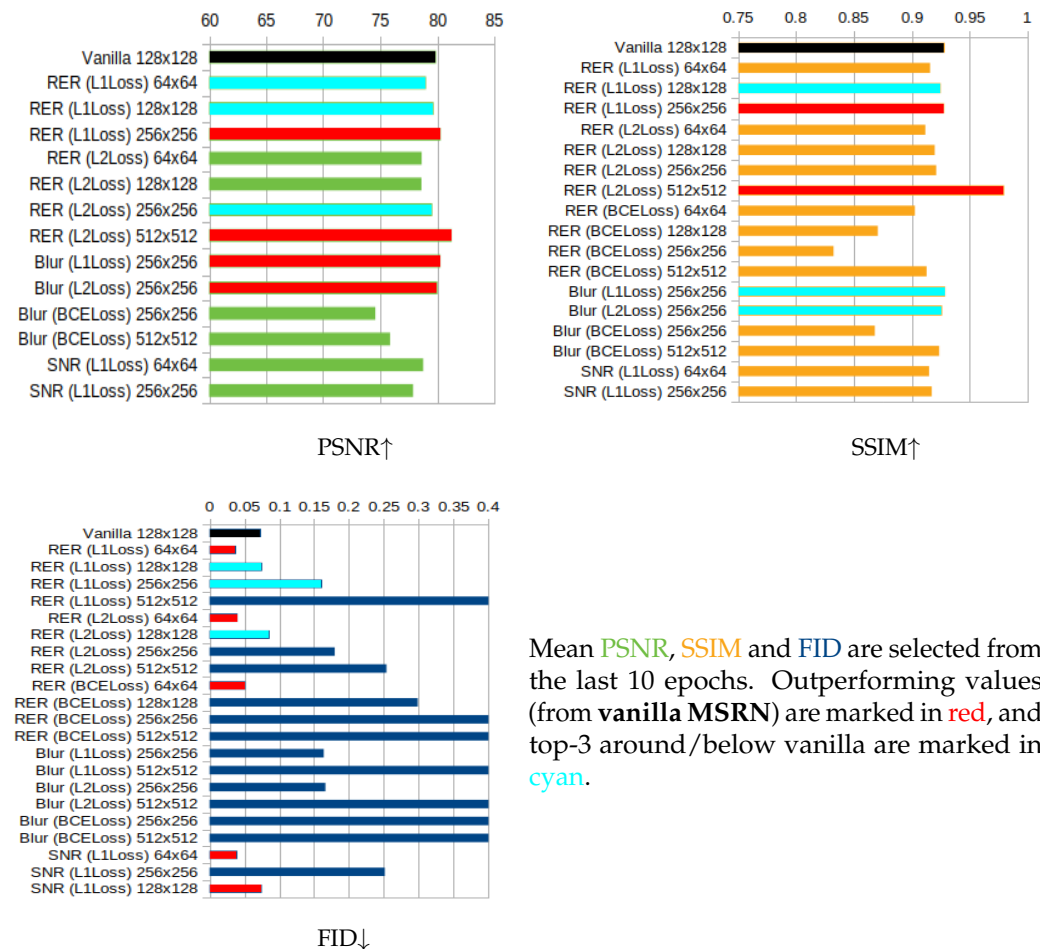

PSNR↑   SSIM↑

FID↓

Mean PSNR, SSIM and FID are selected from the last 10 epochs. Outperforming values (from **vanilla MSRN**) are marked in red, and top-3 around/below vanilla are marked in cyan.

Figure 9. Validation of QMRLoss optimizing MSRN in super-resolution of Inria-AILD-180. Note that the training/validation regime was conducted over Inria-AILD-180 with 100-20 image splits and crops set to 64×64, 128×128, 256×256 and 512×512.

We also tested our MSRN + QMRLoss (adding QMRNet's metric evaluation) generated images with most of our full-reference and no-reference metrics in the UCMerced-380 dataset (outside Inria-AILD's training and validation distribution) with crops of 256×256. Here, vanilla MSRN yields worse results for *blur* ↓, *snr* ↑↓, *rer* ↑↓, SNR_{Mdn} ↓, RER mean of X and Y ↑, MTF mean of X and Y ↑ and FWHM mean of X and Y ↓ in comparison with the optimized $\text{QMRLoss}_{\sigma,L1,256 \times 256}$, $\text{QMRLoss}_{rer,L1,256 \times 256}$ and $\text{QMRLoss}_{F,L1,256 \times 256}$. Here, QMRLoss_{L1} has been able to adapt better when generating contours and predicting blurred objects on testing distinct shapes from the original training. In the case of full-reference metrics, I_{LR} is more similar to the original I_{HR} (although seemingly blurred); this is due to the lack of changes made to the image. In the no-reference metrics, MSRN+QMR significantly improves with respect to I_{LR} and $\text{MSRN}_{vanilla}$.

In Figures 10–12 can be observed the changes of super-resolving UCMerced and Inria-AILD images according to every QMRNet optimization. In most cases of $\text{MSRN}_{vanilla}$ (column 2 colormaps) there is a center bias, especially in sparse/homogeneous regions. In Figure 11, row 3 and Figure 12, row 1 it can be observed that $\text{QMRLoss}_{GSD,L1}$ and $\text{QMRLoss}_{F,L1}$ significatively enhance the noise present in the homogeneous areas (sea/beach), while $\text{QMRLoss}_{\sigma,L1}$, $\text{QMRLoss}_{rer,L1}$ and $\text{QMRLoss}_{snr,L1}$ present a smoother solution whilst having higher oversharpening than $\text{MSRN}_{vanilla}$.

Table 11. Test metrics on super-resolution (super-resolving original input x3 or using downsampled inputs x3) using MSRN backbone +QMRLoss in UCMerced-380. $QMRLoss_{L1}$ computation over Inria-AILD-180-train on distinct QMRNets (for *blur*, *rer* and *snr*) using crops of $R = 256 \times 256$. Note that here we are testing MSRN + QMRLoss over UCMerced samples while QMRNet's training is on Inria-AILD. **Bold** represents having lower distortion than HR. <u>Underline</u> represents top-1 best performance. *Italics* represents the same value for most cases.

|	Algorithm	blur↓	snr↑↓	rer↑↓	F↑↓	GSD↓	Score↑	ssim↑	psnr↑	swd↓	fid↓	SNR_{Mdn}↓	RER_{XY}↑	MTF_{XY}↑	$FWHM_{XY}$↓
	HR	*1.000*	*28.12*	*0.470*	*1.563*	*0.300*	*0.878*	*1.00*	*80.00*	*-*	*0.079*	*20.79*	*0.502*	*0.125*	*1.709*
Original x3	MSRN$_{vanilla}$	*1.000*	<u>28.18</u>	0.464	**<u>1.355</u>**	*0.300*	**0.879**	<u>0.717</u>	<u>23.65</u>	<u>1634</u>	0.411	21.28	0.501	0.124	1.712
	+QMRLoss$_{\sigma,L1}$	*1.000*	26.81	**0.521**	2.285	*0.300*	**0.894**	0.608	21.22	1784	0.815	**13.72**	**0.557**	**0.181**	**1.505**
	+QMRLoss$_{GSD,L1}$	*1.000*	27.62	**0.520**	1.965	*0.300*	**0.896**	0.601	21.40	1799	0.745	**<u>12.81</u>**	**<u>0.573</u>**	**<u>0.195</u>**	**<u>1.447</u>**
	+QMRLoss$_{F,L1}$	*1.000*	26.62	**<u>0.524</u>**	1.947	*0.300*	**<u>0.904</u>**	0.605	21.41	1786	0.790	**13.37**	**0.566**	**0.189**	**1.473**
	+QMRLoss$_{rer,L1}$	*1.000*	26.81	**<u>0.524</u>**	2.178	*0.300*	**0.898**	0.603	21.10	1788	0.850	**13.24**	**0.558**	**0.180**	**1.493**
	+QMRLoss$_{snr,L1}$	*1.000*	26.75	**0.521**	2.232	*0.300*	**0.895**	0.604	21.12	1794	0.850	**13.70**	**0.565**	**0.188**	**1.471**
x3	LR$_{x3}$	1.149	29.94	0.274	*1.000*	*0.300*	0.763	<u>0.778</u>	<u>27.00</u>	<u>1633</u>	0.386	45.33	0.297	0.008	2.946
	MSRN$_{vanilla}$	1.142	***<u>30.00</u>***	0.277	*1.000*	*0.300*	0.765	0.700	24.37	1846	0.502	43.88	0.301	0.010	2.900
	+QMRLoss$_{\sigma,L1}$	<u>1.031</u>	***<u>30.00</u>***	0.307	*1.000*	*0.300*	0.795	0.706	24.35	1804	0.479	36.29	0.314	0.011	2.782
	+QMRLoss$_{GSD,L1}$	1.038	29.94	0.347	*1.000*	*0.300*	0.815	0.701	24.33	1814	0.482	35.02	<u>0.330</u>	<u>0.016</u>	<u>2.664</u>
	+QMRLoss$_{F,L1}$	1.028	29.75	<u>0.412</u>	*1.000*	*0.300*	<u>0.849</u>	0.696	24.26	1803	0.483	<u>34.88</u>	0.328	<u>0.016</u>	2.674
	QMRLoss$_{rer,L1}$	1.036	***<u>30.00</u>***	0.304	*1.000*	*0.300*	0.793	0.704	24.37	1810	0.482	36.32	0.314	0.011	2.787
	+QMRLoss$_{snr,L1}$	1.036	***<u>30.00</u>***	0.305	*1.000*	*0.300*	0.793	0.706	24.34	1797	0.481	35.08	0.315	0.012	2.773
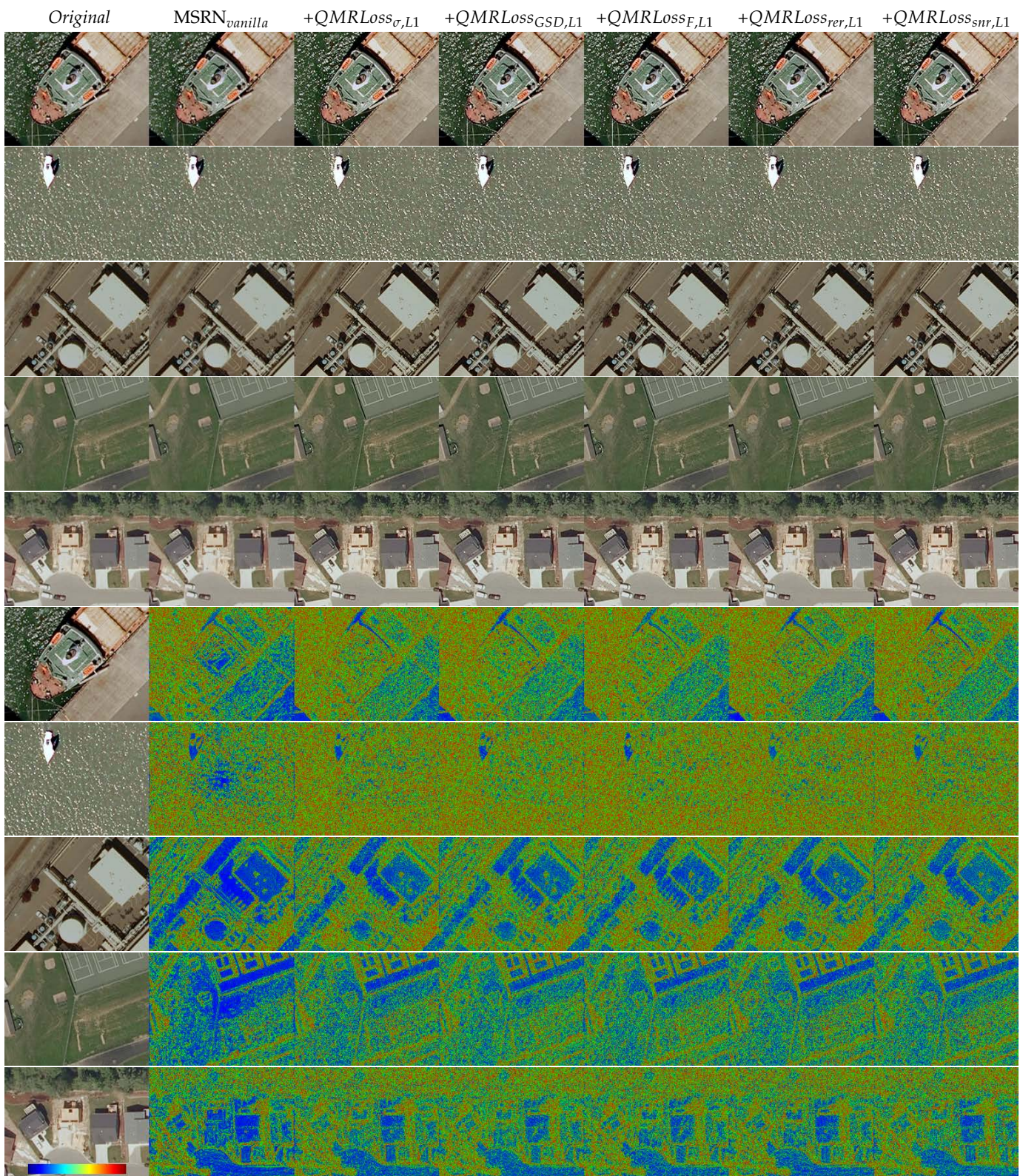
Figure 10. Examples of Inria-AILD-180-test images (with a crop zoom of 256×256) and each QMRNet algorithm output (QMRLoss$_{L1}$). Inputs (*Original I_{HR}*) without downsampling (i.e., super-resolving x3). In rows 6–10 are colormaps of the sum of differences ($\Delta R + \Delta G + \Delta B$) with respect to *Original I_{HR}*.
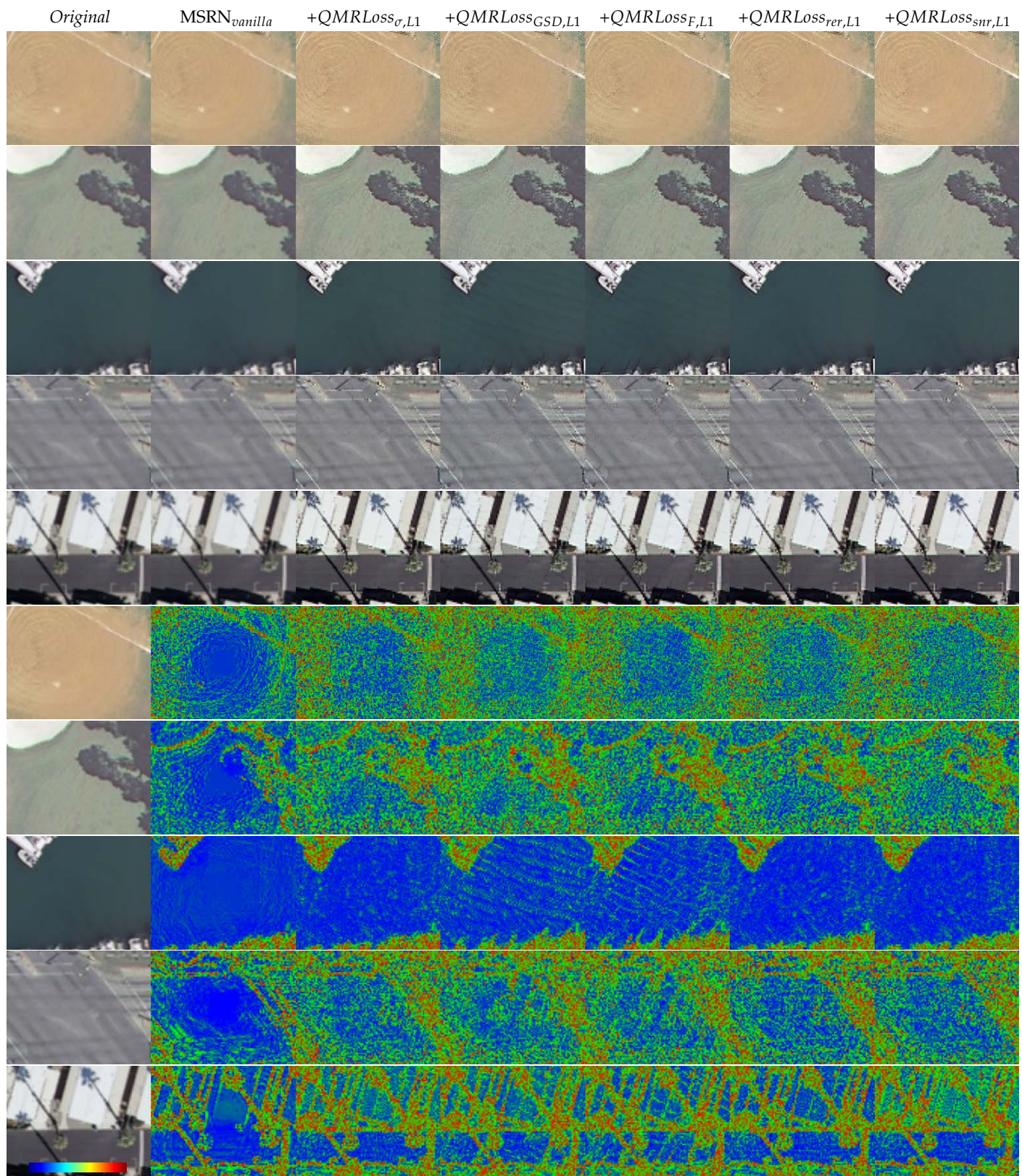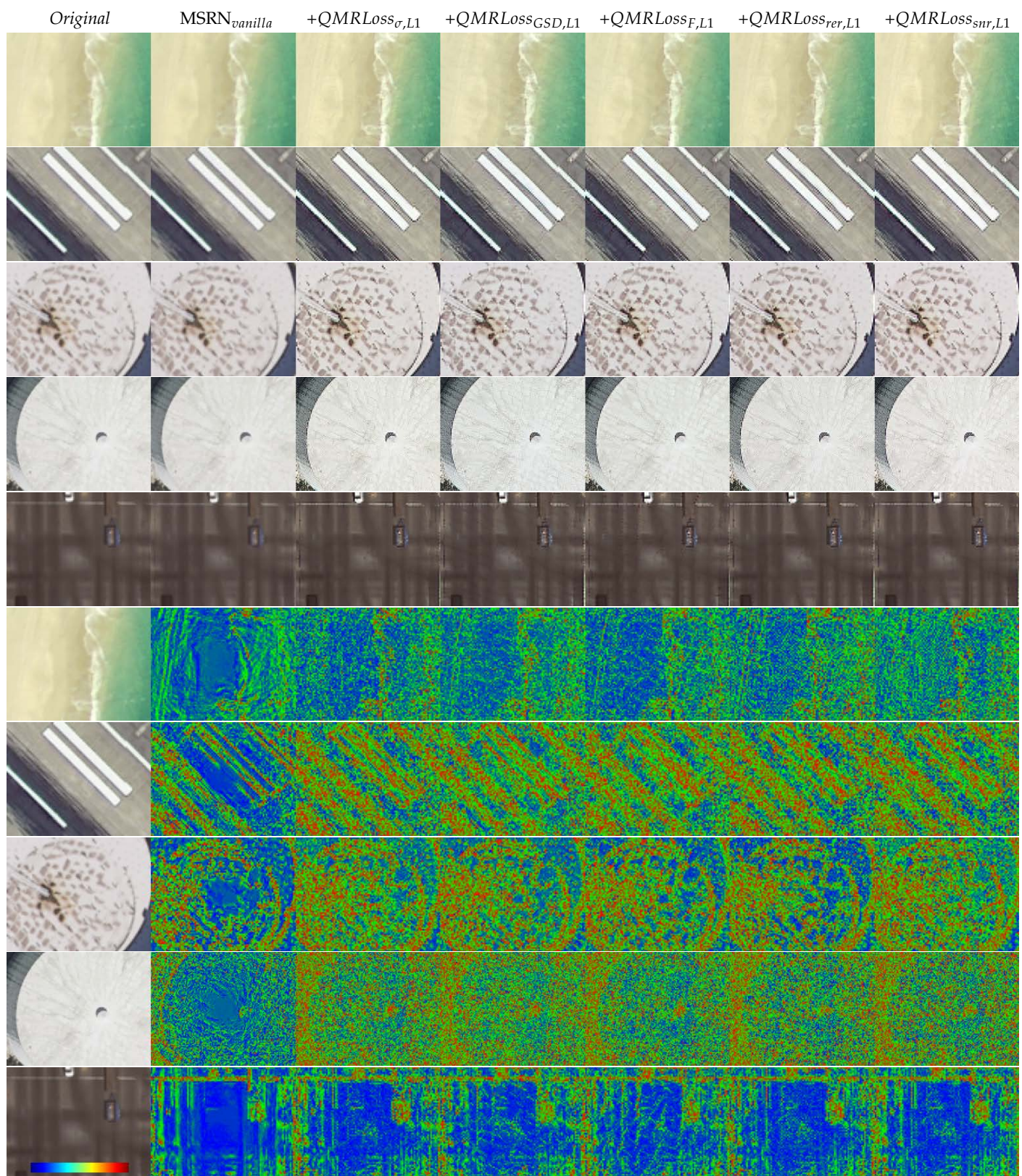
Figure 11. Examples of UCMerced images (with a crop zoom of 96 × 96) and each QMRNet algorithm output (QMRLoss$_{L1}$). Inputs (*Original I$_{HR}$*) without downsampling (i.e., super-resolving x4). In rows 6–10 are colormaps of the sum of differences (ΔR + ΔG + ΔB) with respect to *Original I$_{HR}$*.

Figure 12. Examples of UCMerced images (with a crop zoom of 96 × 96) and each QMRNet algorithm output (QMRLoss$_{L1}$). Inputs (*Original I_{HR}*) without downsampling (i.e., super-resolving x4). In rows 6–10 are colormaps of the sum of differences (ΔR + ΔG + ΔB) with respect to *Original I_{HR}*.

5. Conclusions

In this study, we implement an open-source tool (integrated in the IQUAFLOW framework) developed for assessing quality and modifying EO images. We propose a network architecture (QMRNet) that predicts the amount of distortion for each parameter as a no-reference metric. We also benchmark distinct super-resolution algorithms and datasets with both full-reference and no-reference metrics and propose a novel mechanism for optimizing super-resolution training regimes using QMRLoss, integrating QMRNet metrics with SR algorithm objectives. We tested the performance in single-parameter prediction of *blur*, *rer*, *snr*, *F* and GSD, as well as multiparameter simultaneously. In addition to the high-resolution color EO image computation, we adapted and tested the QMRNet architecture for the prediction of *snr* and *rer* with hyperspectral EO images.

On assessing the image quality of datasets we observe similar overall scores for most datasets, with dissimilarities in the scores of *snr* and *rer*. On assessing the single-image super-resolution we see significantly better results for CAR, LIIF, $MSRN_2$ and $MSRN_3$. Optimizing MSRN with QMRLoss (snr, rer and blur) improves the results on both full-reference and no-reference metrics with respect to the default vanilla MSRN.

We have to point out that our proposed method can be applied to any type of distortion or modification. QMRNet allows us to predict any parameter of the image and also several parameters simultaneously. For instance, training QMRNet to assess compression parameters could be another use case of interest, including other datasets mentioned in Section 2. We also tested the usage of QMRNet as loss for optimizing SR results by regularizing the MSRN network, but it could be extended with distinct algorithm architectures and uses, as QMRLoss allows us to reverse or denoise any modification of the original image. In addition, it is also possible to implement a variation to the QMRLoss objective by forcing the loss calculation to be on a specific interval with maximum quality and minimal distortion for each parameter. In that way, the algorithm could maximize toward a specific metric or *score* objective beyond the output over the GT.

Author Contributions: Conceptualization, D.B., D.V. and J.M.; methodology, D.B.; software, D.B., P.G., L.R.-C., K.T., C.G.-M. and E.M.; validation, D.B.; formal analysis, D.B.; investigation, D.B. and J.M.; resources, D.B. and J.M.; data curation, D.B.; writing—original draft, D.B.; writing—review and editing, D.B., P.G. and J.M.; visualization, D.B.; supervision, D.B., J.M., P.G. and D.V.; project administration, D.B., J.M. and D.V.; funding acquisition, D.B., J.M. and D.V. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement: The research code and data are specified in the following repositories:
- IQUAFLOW https://github.com/satellogic/iquaflow (accessed on 2 April 2023);
- IQUAFLOW-QMRNet https://github.com/satellogic/iquaflow/tree/main/iquaflow/quality_metrics (accessed on 2 April 2023);
- IQUAFLOW-Modifiers https://github.com/satellogic/iquaflow/tree/main/iquaflow/datasets (accessed on 2 April 2023);
- Case 1: Benchmark Datasets https://github.com/dberga/iquaflow-qmr-eo (accessed on 2 April 2023);
- Case 2: Benchmark Super-Resolution https://github.com/dberga/iquaflow-qmr-sisr (accessed on 2 April 2023);
- Case 3: Benchmark QMRLoss https://github.com/dberga/iquaflow-qmr-loss (accessed on 2 April 2023).

Abbreviations

The following abbreviations are used in this manuscript:

SR | I_{SR}	Super-Resolution | SR image
HR | I_{HR}	High-Resolution | HR image
LR | I_{LR}	Low-Resolution | LR image
EO	Earth Observation
IQA	Image Quality Assessment
GSD	Ground Sampling Distance
GAN	Generative Adversarial Networks
SNR	Signal-to-Noise
RER	Relative Edge Response
MTF	Modulation Transfer Function
LSF	Line Spread Function
PSF	Point Sparsity Function
FWHM	Full Width at Half Maximum
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity
MSSIM	Mean Structural Similarity
HAARPSI	Haar Wavelet Perceptual Similarity Index
GMSD	Gradient Magnitude Similarity Deviation
MDSI	Mean Deviation Similarity Index
SWD	Sliced Wasserstein Distance
FID	Fréchet Inception Distance

References

1. Leachtenauer, J.C.; Driggers, R.G. *Surveillance and Reconnaissance Imaging Systems: Modeling and Performance Prediction*; Artech House Optoelectronics Library: Norwood, MA, USA, 2001.
2. Dong, C.; Loy, C.C.; He, K.; Tang, X. Image Super-Resolution Using Deep Convolutional Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 295–307. [CrossRef]
3. Yamanaka, J.; Kuwashima, S.; Kurita, T. Fast and Accurate Image Super Resolution by Deep CNN with Skip Connection and Network in Network. In *Neural Information Processing*; Springer International Publishing: Long Beach, CA, USA, 2017; pp. 217–225. [CrossRef]
4. Müller, M.U.; Ekhtiari, N.; Almeida, R.M.; Rieke, C. Super-resolution of multispectral satellite images using convolutional neural networks. *arXiv* **2020**, arXiv:2002.00580. https://doi.org/10.5194/isprs-annals-v-1-2020-33-2020.
5. Li, J.; Fang, F.; Mei, K.; Zhang, G. Multi-scale Residual Network for Image Super-Resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*; Springer International Publishing: Munich, Germany, 2018; pp. 527–542. [CrossRef]
6. Ledig, C.; Theis, L.; Huszar, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017. [CrossRef]
7. Wang, X.; Yu, K.; Wu, S.; Gu, J.; Liu, Y.; Dong, C.; Qiao, Y.; Loy, C.C. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. In *Workshop of the European Conference on Computer Vision (ECCV)*; Springer International Publishing: Munich, Germany, 2019; pp. 63–79. [CrossRef]
8. Jiang, Y.; Gong, X.; Liu, D.; Cheng, Y.; Fang, C.; Shen, X.; Yang, J.; Zhou, P.; Wang, Z. EnlightenGAN: Deep Light Enhancement Without Paired Supervision. *IEEE Trans. Image Process.* **2021**, *30*, 2340–2349. [CrossRef] [PubMed]
9. Kim, J.; Lee, J.K.; Lee, K.M. Accurate Image Super-Resolution Using Very Deep Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR Oral), Las Vegas, NV, USA, 27–30 June 2016.
10. Sun, W.; Chen, Z. Learned image downscaling for upscaling using content adaptive resampler. *IEEE Trans. Image Process.* **2020**, *29*, 4027–4040. [CrossRef] [PubMed]
11. Chen, Y.; Liu, S.; Wang, X. Learning Continuous Image Representation with Local Implicit Image Function. *arXiv* **2020**, arXiv:2012.09161.
12. Pradham, P.; Younan, N.H.; King, R.L. Concepts of image fusion in remote sensing applications. In *Image Fusion*; Elsevier: Amsterdam, The Netherlands, 2008; pp. 393–428. [CrossRef]
13. Huynh-Thu, Q.; Ghanbari, M. Scope of validity of PSNR in image/video quality assessment. *Electron. Lett.* **2008**, *44*, 800. .:20080522. [CrossRef]
14. Wang, Z.; Bovik, A.; Sheikh, H.; Simoncelli, E. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [CrossRef] [PubMed]

15. Reisenhofer, R.; Bosse, S.; Kutyniok, G.; Wiegand, T. A Haar wavelet-based perceptual similarity index for image quality assessment. *Signal Process. Image Commun.* **2018**, *61*, 33–43. [CrossRef]
16. Xue, W.; Zhang, L.; Mou, X.; Bovik, A.C. Gradient Magnitude Similarity Deviation: A Highly Efficient Perceptual Image Quality Index. *IEEE Trans. Image Process.* **2014**, *23*, 684–695. [CrossRef]
17. Nafchi, H.Z.; Shahkolaei, A.; Hedjam, R.; Cheriet, M. Mean Deviation Similarity Index: Efficient and Reliable Full-Reference Image Quality Evaluator. *IEEE Access* **2016**, *4*, 5579–5590. [CrossRef]
18. Varga, D. Full-Reference Image Quality Assessment Based on an Optimal Linear Combination of Quality Measures Selected by Simulated Annealing. *J. Imaging* **2022**, *8*, 224. [CrossRef] [PubMed]
19. Lim, P.C.; Kim, T.; Na, S.I.; Lee, K.D.; Ahn, H.Y.; Hong, J. Analysis of UAV image quality using edge analysis. *Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2018**, *42*, 359–364. [CrossRef]
20. Mittal, A.; Soundararajan, R.; Bovik, A.C. Making a "Completely Blind" Image Quality Analyzer. *IEEE Signal Process. Lett.* **2013**, *20*, 209–212. [CrossRef]
21. Venkatanath, N.; Praneeth, D.; Bh, M.C.; Channappayya, S.S.; Medasani, S.S. Blind image quality evaluation using perception based features. In Proceedings of the 2015 Twenty First National Conference on Communications (NCC), Mumbai, India, 27 February–1 March 2015. [CrossRef]
22. Leachtenauer, J.C.; Malila, W.; Irvine, J.; Colburn, L.; Salvaggio, N. General Image-Quality Equation: GIQE. *Appl. Opt.* **1997**, *36*, 8322. [CrossRef]
23. Thurman, S.T.; Fienup, J.R. Analysis of the general image quality equation. In *Visual Information Processing XVII*; ur Rahman, Z., Reichenbach, S.E., Neifeld, M.A., Eds.; SPIE: Bellingham, WA, USA, 2008. [CrossRef]
24. Kim, T.; Kim, H.; Kim, H.D. Image-based Estimation and Validation of Niirs for High-resolution Satellite Images. *Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci.* **2008**, *37*, 1–4.
25. Li, L.; Luo, H.; Zhu, H. Estimation of the Image Interpretability of ZY-3 Sensor Corrected Panchromatic Nadir Data. *Remote Sens.* **2014**, *6*, 4409–4429. [CrossRef]
26. Benecki, P.; Kawulok, M.; Kostrzewa, D.; Skonieczny, L. Evaluating super-resolution reconstruction of satellite images. *Acta Astronaut.* **2018**, *153*, 15–25. [CrossRef]
27. Johnson, J.; Alahi, A.; Fei-Fei, L. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *Computer Vision–ECCV 2016*; Springer International Publishing: Amsterdam, The Netherlands, 2016; pp. 694–711. [CrossRef]
28. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018. [CrossRef]
29. Kolouri, S.; Nadjahi, K.; Simsekli, U.; Badeau, R.; Rohde, G. Generalized Sliced Wasserstein Distances. In *Proceedings of the Advances in Neural Information Processing Systems*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Knoxville, TN, USA, 2019; Volume 32.
30. Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A.; Chen, X.; Chen, X. Improved Techniques for Training GANs. In *Proceedings of the Advances in Neural Information Processing Systems*; Lee, D., Sugiyama, M., Luxburg, U., Guyon, I., Garnett, R., Eds.; Curran Associates, Inc.: Knoxville, TN, USA, 2016; Volume 29.
31. Liu, J.; Zhou, W.; Li, X.; Xu, J.; Chen, Z. LIQA: Lifelong Blind Image Quality Assessment. *IEEE Trans. Multimed.* 2022, *Early Access*. [CrossRef]
32. Chen, L.H.; Bampis, C.G.; Li, Z.; Norkin, A.; Bovik, A.C. ProxIQA: A Proxy Approach to Perceptual Optimization of Learned Image Compression. *IEEE Trans. Image Process.* **2021**, *30*, 360–373. [CrossRef]
33. Saad, M.A.; Bovik, A.C.; Charrier, C. Blind Image Quality Assessment: A Natural Scene Statistics Approach in the DCT Domain. *IEEE Trans. Image Process.* **2012**, *21*, 3339–3352. [CrossRef]
34. Mittal, A.; Moorthy, A.K.; Bovik, A.C. No-reference image quality assessment in the spatial domain. *IEEE Trans. Image Process.* **2012**, *21*, 4695–4708. [CrossRef] [PubMed]
35. Ye, P.; Kumar, J.; Kang, L.; Doermann, D. Unsupervised feature learning framework for no-reference image quality assessment. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 1098–1105. [CrossRef]
36. Xu, J.; Ye, P.; Li, Q.; Du, H.; Liu, Y.; Doermann, D. Blind Image Quality Assessment Based on High Order Statistics Aggregation. *IEEE Trans. Image Process.* **2016**, *25*, 4444–4457. [CrossRef] [PubMed]
37. Liu, X.; van de Weijer, J.; Bagdanov, A.D. RankIQA: Learning From Rankings for No-Reference Image Quality Assessment. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
38. Bosse, S.; Maniry, D.; Muller, K.R.; Wiegand, T.; Samek, W. Deep Neural Networks for No-Reference and Full-Reference Image Quality Assessment. *IEEE Trans. Image Process.* **2018**, *27*, 206–219. [CrossRef]
39. Fan, C.; Zhang, Y.; Feng, L.; Jiang, Q. No Reference Image Quality Assessment based on Multi-Expert Convolutional Neural Networks. *IEEE Access* **2018**, *6*, 8934–8943. [CrossRef]
40. Zhu, H.; Li, L.; Wu, J.; Dong, W.; Shi, G. MetaIQA: Deep Meta-Learning for No-Reference Image Quality Assessment. In Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020; pp. 14143–14152.

41. Sun, S.; Yu, T.; Xu, J.; Zhou, W.; Chen, Z. Graphiqa: Learning distortion graph representations for blind image quality assessment. *IEEE Trans. Multimed.* 2022, *Early Access*. [CrossRef]

42. Zhou, W.; Wang, Z. Quality Assessment of Image Super-Resolution: Balancing Deterministic and Statistical Fidelity. *arXiv* **2022**, arXiv:2207.08689.

43. Jiang, Q.; Liu, Z.; Gu, K.; Shao, F.; Zhang, X.; Liu, H.; Lin, W. Single Image Super-Resolution Quality Assessment: A Real-World Dataset, Subjective Studies, and an Objective Metric. *IEEE Trans. Image Process.* **2022**, *31*, 2279–2294. [CrossRef]

44. Zhou, W.; Jiang, Q.; Wang, Y.; Chen, Z.; Li, W. Blind quality assessment for image superresolution using deep two-stream convolutional networks. *Inf. Sci.* **2020**, *528*, 205–218. [CrossRef]

45. Maggiori, E.; Tarabalka, Y.; Charpiat, G.; Alliez, P. Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark. In Proceedings of the 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Fort Worth, TX, USA, 23–28 July 2017; pp. 3226–3229. [CrossRef]

46. Dalponte, M.; Frizzera, L.; Gianelle, D. Individual tree crown delineation and tree species classification with hyperspectral and LiDAR data. *PeerJ* **2019**, *6*, e6227. [CrossRef]

47. Gallés, P.; Takáts, K.; Hernández-Cabronero, M.; Berga, D.; Pega, L.; Riordan-Chen, L.; Garcia-Moll, C.; Becker, G.; Garriga, A.; Bukva, A.; et al. iquaflow: A new framework to measure image quality. *arXiv* **2022**, arXiv:2210.13269.

48. Gallés, P.; Takáts, K.; Marín, J. Object Detection Performance Variation on Compressed Satellite Image Datasets with Iquaflow. *arXiv* **2023**, arXiv:2301.05892.

49. Carvalho, M.; Cadène, R.; Picard, D.; Soulier, L.; Thome, N.; Cord, M. Cross-Modal Retrieval in the Cooking Context. In Proceedings of the The 41st International ACM SIGIR Conference on Research Development in Information Retrieval, Ann Arbor, MI, USA, 8–12 July 2018. [CrossRef]

50. Salvador, A.; Hynes, N.; Aytar, Y.; Marin, J.; Ofli, F.; Weber, I.; Torralba, A. Learning Cross-Modal Embeddings for Cooking Recipes and Food Images. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017. [CrossRef]