An Investigation of Metrics to Evaluate the Sharpness in AI-Generated Meteorological Imagery (Draft version - Jan 26, 2024)

Imme Ebert-Uphoff [1,2], Lander Ver Hoef [1], John S. Schreck [3], Jason Stock [4], Maria J. Molina [5,3], Amy McGovern [6], Michael Yu [6], Bill Petzke [3], Kyle Hilburn [1], David M. Hall [7], David J. Gagne [3], Sam Scheuerman [8]

[1]Cooperative Institute for Research in the Atmosphere (CIRA), Colorado State University, Fort Collins, CO, USA.
[2]Electrical and Computer Engineering, Colorado State University, Fort Collins, CO, USA.
[3]National Center for Atmospheric Research (NCAR), Boulder, CO, USA.
[4]Computer Science, Colorado State University, Fort Collins, CO, USA.
[5]Department of Atmospheric and Oceanic Science, University of Maryland, College Park, MD, USA.
[6]School of Computer Science and School of Meteorology, University of Oklahoma, Norman, OK, USA.
[7]NVIDIA, Santa Clara, CA
[8]Mathematics, Colorado State University, Fort Collins, CO, USA

Key Points:

- AI-based estimates of meteorological images, e.g., for forecasting applications, often lack sharpness, but there are no well established metrics to measure sharpness of meteorological imagery.
- This manuscript seeks to close this gap by selecting and exploring different sharpness metrics for meteorological imagery, and by providing guidelines for their use and interpretation
- We hope that the tools provided here will aid the development of AI algorithms to provide more realistic meteorological imagery.

Corresponding author: Imme Ebert-Uphoff, iebert@colostate.edu

Abstract

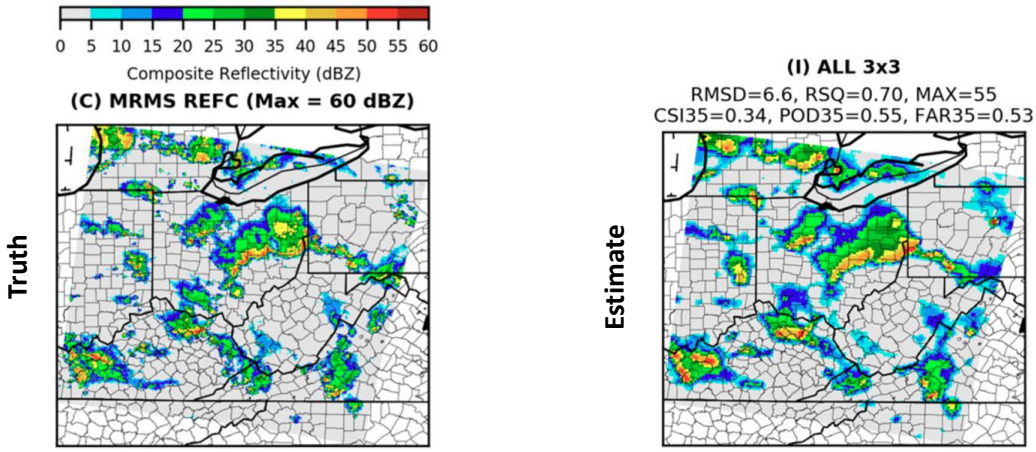
AI-based algorithms are emerging in many meteorological applications that produce imagery as output, including for global weather forecasting models. However, the imagery produced by AI algorithms, especially by convolutional neural networks (CNNs), is often described as too blurry to look realistic, partly because CNNs tend to represent uncertainty as blurriness. This blurriness is undesirable since it might obscure important meteorological features. More complex AI models, such as generative adversarial networks (GANs) and diffusion models, generate images that appear to be sharper, but that sharpness may come at the expense of a decline in other performance criteria, such as accuracy. To choose a good trade-off between sharpness and accuracy for a specific task it is important to quantitatively assess both accuracy and sharpness. While there are numerous well-established measures for accuracy there is a lack of well-established measures for sharpness in meteorological imagery. The purpose of this paper is to fill this gap by 1) exploring a variety of sharpness metrics from other fields, 2) analyzing their suitability for meteorological applications, 3) suggesting protocols for how to use and interpret them, and 4) demonstrating their use for sample meteorological applications using vignettes.

1 Introduction

Convolutional neural networks (CNNs) are known to produce blurry imagery, since CNNs tend to express uncertainty through blurriness. Thus, using CNNs to produce meteorological imagery often results in imagery that is too blurry, as illustrated by the GREMLIN model discussed in Subsection 1.1. Newer models, such as generative adversarial models (GANs) and diffusion models (which belong to the class of Generative AI models) provide much sharper imagery, but there are no established sharpness measures yet in the meteorological literature that would allow the community to evaluate improvements in sharpness. Here we seek to establish a group of metrics that can evaluate sharpness for meteorological imagery, to allow the community to explore trade-offs between accuracy, sharpness, and other performance metrics for emulated imagery, in particular in comparison to observed imagery.

1.1 CNN model "GREMLIN" as a Guiding Example

Throughout this manuscript, we use a specific CNN model, named GREMLIN, as a guiding example to illustrate the sharpness issue and the corresponding evaluation of image sharpness. GREMLIN (Hilburn et al., 2020)) is a CNN model that estimates radar reflectivity from imagery taken by a geostationary satellite. GREMLIN was developed to provide estimates of radar imagery in regions where radar imagery is not available, such as in mountainous and remote terrain and over oceans. Fig. 1 illustrates this process. Figure 1(a) shows a sample observed radar image (ground truth) and Figure 1(b) shows the corresponding estimate from GREMLIN, which is much blurrier. We would like the CNN to provide sharper features. While there are various standard measures for the accuracy of the estimate (see for example the statistics provided on top of the Fig. 1(b), there is no established means to measure and compare the sharpness of both images.



(a) (b)

Figure 1. Example of GREMLIN model illustrating blurriness: (a) Observed composite reflectivity from radar (ground truth); (b) Corresponding estimate of (a) based on satellite imagery generated by the GREMLIN (CNN) model. Error statistics for the estimate (b) are included at the top of the image. It is clear from visual inspection that the estimated image is much blurrier than the original radar observations, but there is no established measure in meteorology to compare the sharpness of the two images.

1.2 Scope and Organization of this paper

The scope of this paper is to provide several metrics to evaluate the sharpness of any type of meteorological image – whether it comes from observations, from a physics-based model, or from an AI-generated model – along with guidelines on how to choose metrics and how to interpret them. Some of the resulting metrics are straightforward to include in loss functions. This paper does *not* explore the consequences of including these metrics in AI models to produce sharper imagery - that is the topic of future research.

The contents and contributions of this article are as follows:

- Section 2 discusses sharpness-related concepts from the areas of photography and computer vision.
- Section 3.2 identifies several interpretable metrics from other fields that may be suitable for meteorological imagery. Ready-to-use implementations will be provided shortly in an accompanying GitHub repository.
- Section 3.3 provides heatmaps of local sharpness to visualize which features in an image are considered to be particular sharp by the varying metrics.
- Section 3.4 develops a calibration scheme that allows scientists to more easily interpret the numbers provided by the various metrics.
- Section 4 identifies key properties of interest for sharpness metrics that are relevant for meteorological applications.
- Section 5 provides vignettes that demonstrate the practical use of these metrics for several different applications, highlighting both their benefits and how to use them.
- Section 6 provides a final discussion and suggests topics for future work.

1.3 Code availability

Python code implementing the metrics discussed here, plus additional metrics, as well as code for plotting heatmaps, etc., will be available shortly on GitHub at `https://github.com/ai2es/sharpness/`.

2 What is sharpness?

The term *image sharpness* is used extensively in literature, but it is difficult to find a consistent definition. In this section we first discuss definitions of sharpness in photography, followed by a classification from the computer vision literature.

2.1 Sharpness measures in photography

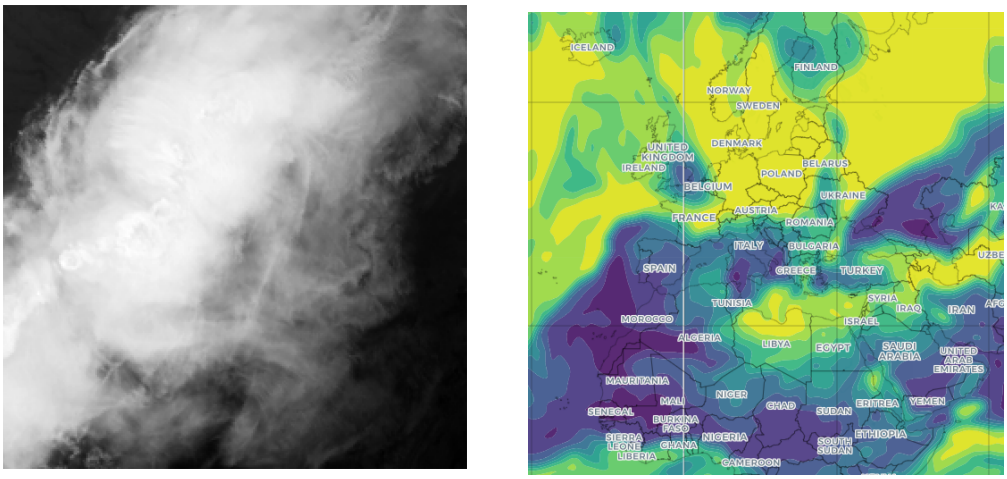
In photography, the following definition is representative (SLR Lounge, 2023):

> Technically speaking, sharpness is defined as the acuity, or contrast, between the edges of an object in an image. A well-defined edge, one that makes an abrupt transition from one color or tone to another, thus defining that object in the photo, is considered to be "sharp."

Consequently, a common industry standard for measuring the sharpness of imaging systems, such as cameras, is the *rise distance* illustrated in Fig. 2. Note that this definition, and many others, assume the presence of cleanly defined edges in the image to define sharpness. However, meteorological imagery, such as the two examples shown in Fig. 3, may not include any such edges. In fact, the perception of sharpness can come from different sources, such as:

(a) (b)

Figure 2. In photography sharpness is often defined by the rise distance of boundary transitions. Panel (a) shows a bar pattern (upper half) and the same image with lens degradation (lower half). Panel (b) illustrates the definition of the rise distance as the width of the red bracket, where the x-axis indicates the distance traveled (orthogonally) across an edge in the image and the y-axis indicates the corresponding pixel values in the image. The rise distance is the spatial distance in an image transition from a 10% intensity to 90% intensity level of the transition between two values. The blurrier the image, the larger the rise distance. Image source: Imatest (2023)



(a) (b)

Figure 3. Two examples of meteorological imagery: (a) a satellite image from GOES-16 showing clouds; (b) output of relative humidity from FourCastNet v2 (Bonev et al., 2023), an AI weather forecasting model. Credit for Panel (b): Jacob Radford (jacob.t.radford@gmail.com) - source `https://aiweather.cira.colostate.edu/`.

1. Edges, i.e., the boundary between different "zones";
2. Textures, i.e., details within a "zone."

For example, when considering a satellite image of clouds such as shown in Fig. 3(a), the texture of the clouds might contribute more or less sharpness than the boundaries of the clouds, depending on which metric is used. Furthermore, consider meteorological images showing atmospheric variables, such as temperature or relative humidity, see Fig. 3(b) for an example. A contour line at any threshold could be considered an edge and it is not obvious which threshold should be chosen for a meaningful analysis of overall sharpness. Thus, we mustn't restrict ourselves to metrics that focus only on sharpness that arises from (sharp) edges, such as the rise distance illustrated in Fig. 2. In photography, there are therefore also sharpness metrics that analyze image frequencies, e.g., by first applying a Fourier transform, and thus do not rely on the existence of clearly defined edges. Nevertheless, metrics from photography are of limited use for our purposes, because they

tend to assume that the properties of a single lens are responsible for the blurriness across an entire image. In contrast, AI-generated imagery exhibits blurriness that differs in two important ways: 1) blurriness is a local phenomenon, i.e., blurriness tends to vary greatly across each image, and 2) even in the local space the mechanism behind the blurriness cannot be represented by properties of an optical lens.

2.2 Sharpness measures in computer vision

The field of computer vision has developed a wider set of sharpness definitions and metrics that are more suitable for our purpose. Vu et al. (2011) introduced the following classification of sharpness metrics:

> Modern methods of sharpness/blurriness estimation can generally be classified into three main trends: 1) edge-based methods, which involve measuring the spread of edges; 2) pixel-based methods, which work in the spatial domain without any assumption regarding edges; and 3) transform-based methods, which work in the spectral domain.

For this study we want to limit the number of metrics, while also providing an intuitive understanding of each, thus selected a set of metrics that have simple mathematical equations, are easy to understand, appear useful for meteorological applications, and cover a wide range of different concepts. The following list discusses our selection using the classification by Vu et al. (2011) above:

1. *Edge-based metrics*, such as the rise distance, first identify edges, then analyze their properties. We do *not* cover these metrics due to their underlying assumption that images must have well-defined edges.
2. *Pixel-based metrics*, aka *spatial metrics*, include gradient-based methods and methods based on eigenvalues using singular value decomposition (SVD) (Wee & Paramesran, 2008) of images. We cover several gradient-based metrics here. We do not include metrics based on eigenvalues/SVD, as they are much more abstract, and less commonly used than gradient-based metrics. Those may be added in future studies.
3. *Transform-based metrics*, aka *spectral metrics*, include methods based on Fourier or wavelet transforms, and we include metrics based on both transforms.
4. *Neural network based metrics* were not yet discussed by Vu et al. (2011), because they did not yet exist. These metrics utilize the latent space of trained neural networks to assess image properties, e.g., see Zhang et al. (2018). Those are *not* covered here, as their functionality is too opaque (i.e., black box character) for this first study.

3 Metrics

In this section, we discuss metrics for both accuracy and sharpness. We emphasize that none of these metrics are new. In fact, most of the accuracy metrics are very standard, and all metrics have previously been used in other applications. The contribution here is that we explore how to use these metrics for meteorological images. As mentioned above, to maximize intuitive understanding and transparency of our metrics, we focus here on metrics defined by mathematical equations, and out of those choose the simplest ones. This excludes perception-based metrics that utilize trained neural networks (Zhang et al., 2018), which while potentially powerful, are fairly opaque. Before we dive further into these metrics we briefly discuss the assumptions for the images to be evaluated.

3.1 Image Assumptions

In this manuscript, we assume images to be two-dimensional. Many of the concepts discussed here also apply to higher dimensional images, but for ease of explanation, we restrict our discussion to 2D images with the two dimensions denoted as x and y.

Furthermore, we only consider single-channel (i.e., gray) images here. For images with multiple channels (or containing multiple colors), we currently suggest applying these measures to each channel separately and combining the results as desired (e.g., taking min, max, or average). However, an exploration of sharpness for images with multiple channels is a topic for future research.

Lastly, we assume images to have no missing values (no NaNs).

3.2 Overview of primary metrics

This section provides a quick overview of the primary metrics we focus on. For simplicity we refer to each of these as being computed across an "image," but they can each also be computed on smaller subsets of an image, as we will see in Section 3.3. We consider three types of metrics below, 1) image intensity and similarity, 2) sharpness metrics based on total variation and image gradients, and 3) sharpness metrics in spectral space. Each group is described in one subsection.

We distinguish between *univariate* metrics, which take a single image as input at a time, and *bivariate* metrics, which require two input images at a time. Accuracy metrics are always bivariate, as one always needs a ground truth for comparison to assess the accuracy of an image, i.e., to determine how similar one image is to another in terms of accuracy. Likewise, bivariate sharpness metrics can only be applied to a pair of images, and the output describes the difference between their sharpness, i.e., how similar they are in terms of sharpness. In contrast, univariate sharpness metrics are applied to a single input image and assess the sharpness of just that image. To compare the sharpness of two images, one calculates the univariate metric for each and then analyzes their difference.

In the following subsections, we provide for each metric a short description, the abbreviated name of its implementation in the GitHub repository (in parenthesis), and whether the metric is univariate or bivariate.

3.2.1 Group 1: Image Intensity and Similarity metrics

While the purpose of this study is to evaluate the sharpness of imagery, it is important to consider sharpness and accuracy in tandem, namely to make sure that increasing sharpness does not come at the expense of drastically reducing accuracy. We selected the following three metrics to measure and compare the intensity and similarity of images.

1. **Image Intensity [univariate]:** We keep track of the min, mean, and max intensity value of each image, because the dynamic range of an image has a significant effect on its apparent sharpness. An easy way to increase many sharpness metrics of an image would be to just increase its dynamic range - which is typically not what we want. This motivates us to keep track of the intensity of images.
2. **Root Mean Squared Error (rmse) [bivariate]:** RMSE is the square root of the mean squared error (MSE) between two images. This makes it more directly comparable to mean absolute error (MAE) while retaining the higher penalty for large deviations. We keep track of RMSE to make sure we do not drastically reduce the accuracy of image estimates while trying to make them sharper.

3. **Structural Similarity Index Measure (ssim) [bivariate]:** SSIM is a similarity measure between two images based on a weighted combination of three simpler comparisons: luminance, contrast, and structure. The product of these measures gives SSIM. An important note is that SSIM acts on a patchwise rather than pixelwise basis, and as such can capture more spatial information than pixelwise methods like MAE, MSE, or RMSE. SSIM values range between 0 and 1, with SSIM = 1 indicating identical images and values approaching 0 indicating increasingly dissimilar images. SSIM is often cited to better represent image similarity - as perceived by humans - than, for example, RMSE. For details, see Wang et al. (2004).

3.2.2 Group 2: Sharpness metrics based on total variation and image gradients

Since sharp boundaries result in sharp gradients, it is intuitive to use properties related to the gradient of an image to assess its sharpness. Total variation is very similar to gradient-based methods and is thus included here. We expect this group of metrics to respond strongly to sharp edges in an image.

1. **Total Variation (tv) [univariate]:** Total variation measures how much an image changes if it is shifted slightly. This can measure the sharpness of edges because when a sharp edge is shifted slightly it will cause a larger difference than if a smoother edge is shifted the same amount. TV values close to 0 indicate very smooth images, while sharper images will have larger TV values. It is important to note that TV is not normalized by image size, so TV values for images (or blocks) of different sizes are not comparable, and it is normal to get TV values that are very large compared to most other metrics described here.
2. **Mean Gradient Magnitude (grad-mag) [univariate]:** At each pixel, we can compute gradients in both the horizontal (x) and vertical (y) directions; the *magnitude* of the gradient at that pixel is then the norm of the vector formed by those directional gradients. The grad-mag is the mean of these gradient magnitudes across the image, and as such gives a summary statistic that reports, on average, how rapidly intensity changes occur within the image. More rapid intensity changes generally correspond with sharper images, so higher grad-mag values indicate a sharper image, with grad-mag = 0 indicating a completely uniform image with no variation.
3. **Gradient Total Variation (grad-tv) [univariate]:** Gradient total variation is the total variation of the gradient magnitude map, where the gradient map is described in grad-mag above, and total variation is as described in TV. Because both TV and gradients measure sharpness, the gradient TV is really giving information about how sharp the sharpness map is - i.e., are areas of rapid change (associated with sharpness) themselves sharp. In practice, this second-order sharpness seems to correspond with sharpness.
4. **Gradient RMSE (grad-rmse) [bivariate]:** In this bivariate metric, we compute the RMSE not between two images directly, but between two gradient magnitude images. We compute the gradient magnitudes as in grad-mag above, but rather than averaging those across a single image to obtain a statistic, we compute the RMSE between the gradient maps for two distinct images. As in general for RMSE, values closer to 0 indicate more similarity, while larger values indicate more dissimilarity. By taking the RMSE of gradient magnitude maps, we are measuring how closely aligned regions of rapid change are between the two images; i.e., measuring how well sharp edges correspond between the two images.
5. **Laplace RMSE (laplace-rmse) [bivariate]:** Laplace RMSE is very similar to gradient RMSE, but instead of taking the magnitude of the gradient vector at each pixel, we compute the divergence of the gradient at each pixel, which is a way of quantifying the local shape of the gradient vector field. By taking the RMSE of

two such divergence maps, we are computing how similar the shapes of edges are between two images. As with any of these RMSE measures, values close to 0 indicate that the two images have very similar Laplacian maps, while larger values indicate larger differences.

3.2.3 Group 3: Sharpness metrics in spectral space

The last set of metrics seeks to analyze the sharpness of images in spectral space. The idea it to first apply a Fourier or wavelet transformation, and then to analyze image properties in the corresponding spectral representation of the image.

1. **Fourier RMSE (fourier-rmse) [bivariate]:** When taking the 2D Fourier transform, the resulting complex-valued phase space can be reduced down to the *power spectrum* by taking the absolute value of the complex values at each frequency, which gives another real-valued 2D array. Fourier RMSE is then the RMSE between the power spectra of the two images being compared. Note that in the power spectrum, spatial coordinates correspond to frequencies, which are all weighted evenly in this RMSE computation.
2. **Fourier Total Variation (fourier-tv) [univariate]:** We once again start with the power spectrum, but instead of comparing two power spectra, we take the Total Variation of the power spectrum for a single image. The power spectrum contains information about sharpness (as high-frequency information can be interpreted as "sharp"), and TV measures how sharp the power spectrum is, so like Grad-TV, we have some degree of second-order sharpness.
3. **Spectral Slope (spec-slope) [univariate]:** As mentioned in FTV, the power spectrum of an image contains information on how sharp an image is – in particular, the distribution of high vs low-frequency information. Spectral slope measures this distribution, and is very sensitive to blurring. It is also entirely invariant to uniform changes in intensity – i.e., rescaling the image will not change the spectral slope value. Values are all negative, with more negative values indicating less sharp images.
4. **Wavelet Total Variation (wavelet-tv) [univariate]:** WTV is based on the wavelet transform, which takes in an image and (for one level) yields a set of four output arrays, the approximation coefficients, and three sets of detail coefficients. The detail coefficients contain information about variation in the image at various scales and orientations, while the approximation coefficient contains information about average intensities, so by summing the absolute value of all of these coefficients, we arrive at a notion of total variation in the image utilizing wavelets. Like Total Variation, we view increasing values of WTV as having higher sharpness and note that WTV is also not normalized by the size of the image, so WTV values for different image (or block) sizes are not comparable.

3.3 Heatmaps and Stats Plots

Heatmaps: Since meteorology is a very visual field, we believe it is essential for all of the metrics to not only provide a single number for quantitative assessment/optimization of sharpness or accuracy, but also a visual representation of which features in an image are perceived to be particularly sharp or blurry. To provide such visual feedback we generate *sharpness heatmaps* by evaluating small patches of each image and displaying the resulting local information as an image, i.e., the heatmap of an image for a specific metric. Throughout the experiments in this paper, we use square blocks with edges that are 1/8th the length of the horizontal edge length of the input image. We use overlapping blocks, to avoid the issue that can arise with disjoint blocks where an edge lying along the border between two blocks is not detected by either. For most experiments, adjacent blocks overlap 75% of their area, but for blocks smaller than 8×8, because of the

discrete nature of pixels, the overlap may be less than 75% as we enforce a minimum block stride of 2 pixels. The output heatmap reports the values for each block on the central pixels of that block, but because of the overlap, each block includes information from a larger region than its value is outputted to. For all metrics that utilize the Fourier transform, we implement Hanning windowing on each block to minimize the edge effects on the Fourier transform. Each heatmap can be shown on its own or used as an overlay over the input image(s) to indicate areas with very high or low values of each metric.

Stats plots: In addition, we keep track of the min, mean, and max of all heatmaps and show those statistics in a separate graph, which we will refer to as *stats plots*. Heatmaps have already been used before, see Vu et al. (2011), while the stats plots are introduced here. They become particularly important (and less trivial) once they are combined with the calibration procedure in Section 3.4.

Figures 4 to 6 illustrate the use of the heatmaps and stats plots for one sample of the GREMLIN model. We use the following colors to indicate the different types of heatmaps:

- **Gray** indicates the original image, i,e., image intensity.
- **Blue** indicates values of univariate metrics, i.e., metrics that are calculated from an individual image (no comparison).
- **Red** indicates values of bivariate metrics, i.e., metrics that compare two images. Throughout this paper, all bivariate metrics indicate the comparison of each image to the original image, which is always shown on the top left. Thus bivariate metrics for the original image itself are always zero.

Dealing with NaNs: Across all heatmpas **yellow** indicates individual pixels with invalid values (NaNs). We have observed NaNs only for two metrics, ssim and spec-slope. Both of them are undefined in areas in the original image that are identically zero. In future versions we plan to test for such cases during calculations and find better ways to deal with these cases, such as outlined, for example, by Vu et al. (2011). Note that the statistics of the heatmaps used in the stats plots, i.e. min/mean/max values of the heatmaps, are taken across all valid pixels, i.e., pixels with NaNs are currently ignored in those statistics.

The heatmaps in Fig. 4 for rmse and ssim for the original image are zero throughout (with some NaNs for ssim), since the image is compared to itself. The corresponding heatmaps for the estimate show the largest errors at the high-intensity regions of the image. That is to be expected, as it is easy for GREMLIN to predict areas with no significant signal, and the majority of GREMLIN errors are always in areas with high intensity.

The heatmaps in Fig. 5 illustrate sharpness according to the gradient-based metrics. The heatmaps with blue color maps (tv, grad-mag, and grad-tv) indicate univariate sharpness metrics, i.e. they show the sharpness of each individual image (no comparison), with higher values (darker blue) indicating higher sharpness of the image. According to there three metrics the original image is much sharper than the estimate, which is most apparent in the high intensity regions. The heatmaps with red color maps (grad-rmse and laplace-rmse) are bivariate sharpness metrics. They show the difference of sharpness between each image and the original image (so, trivially, they are zero for the original image). As expected the differences are largest roughly where the univariate metrics indicate the areas of highest sharpness in the original (observed radar) image.

The heatmaps in Fig. 6 illustrate sharpness according to the sharpness metrics in spectral space. The heatmap in red, fourier-rmse, which is bivariate, indicates very specific areas of sharpness, difference also mostly in the high-density areas of the original image. Note how much more focused the areas of sharpness difference are in contrast to the gradient-based methods. The heatmaps in blue, fourier-tv, spec-slop, and wavelet-

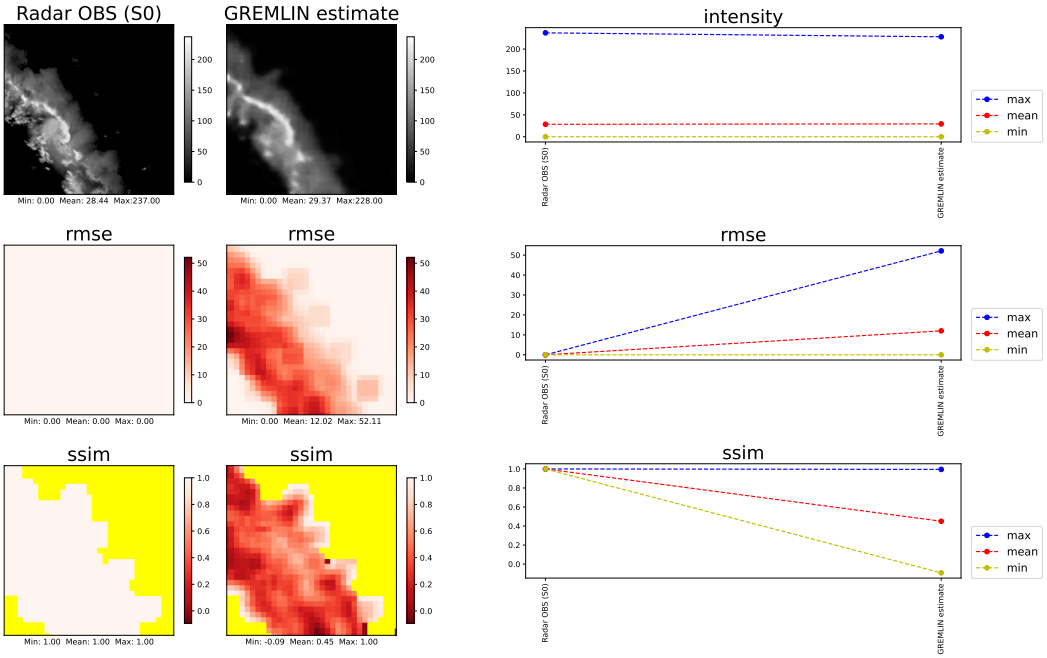


Figure 4. Comparison of observed radar image and its estimate from a neural network model, GREMLIN, using metric group 1: (a) original image and GREMLIN estimate, along with heatmaps for rmse and ssim for both images; (b) stats plots that track the min, mean, and max values of the heatmaps for both images. Original image intensity is shown in gray and bivariate metrics are shown in red.

tv, indicate the sharpness of each individual image (no comparison). fourier-tv indicates a relatively narrow band of sharpness not unlike the gradient-based univariate metrics. wavelet-tv indicates an even narrower band of sharpness in each image. spec-slope stands out indicating a very broad area of sharpness, including areas in which the original image appears to have no signal. The reason is that spec-slope has a very unusual property, namely it is *invariant to the intensity of the signal*, i.e. it responds to sharp features with small intensity difference just as much as to those with high intensity difference.

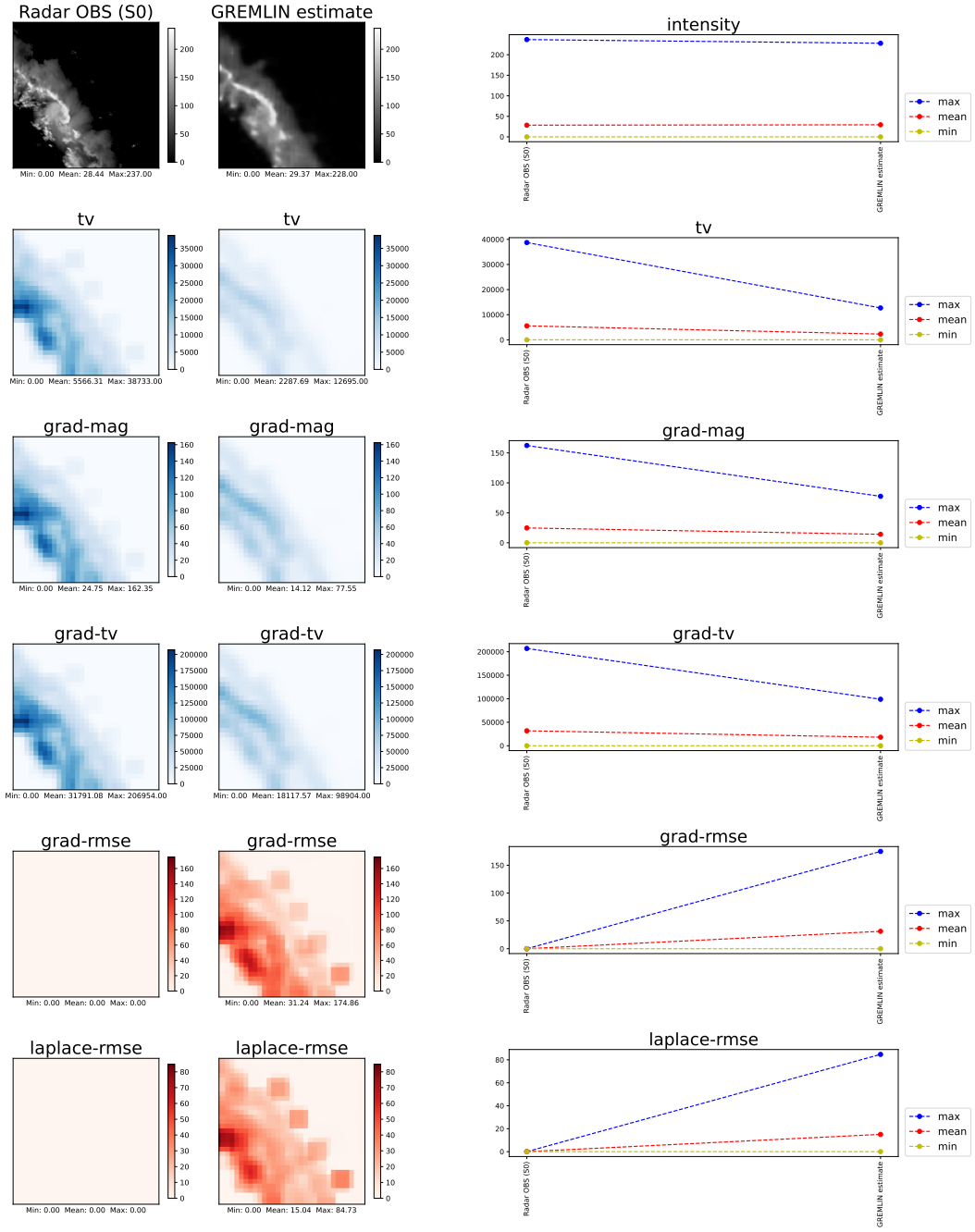


Figure 5. Comparison of observed radar image and image estimate from neural network model, GREMLIN, using metric group 2. Univariate metrics are shown in blue and bivariate metrics in red.

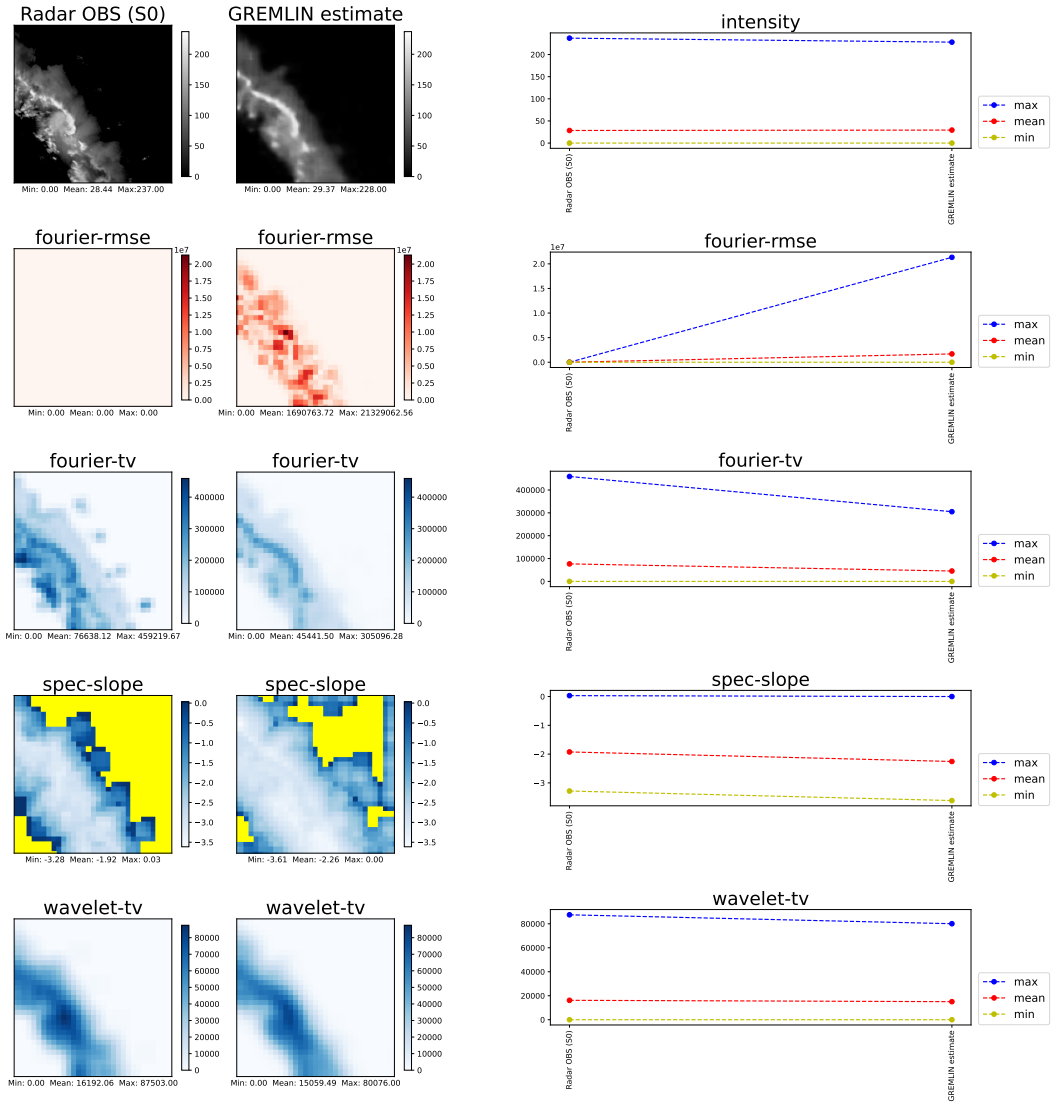


Figure 6. Comparison of observed radar image and image estimate from neural network model, GREMLIN, using metric group 3. Univariate metrics are shown in blue and bivariate metrics in red.

3.4 Protocol to aid interpretation of each metric's values

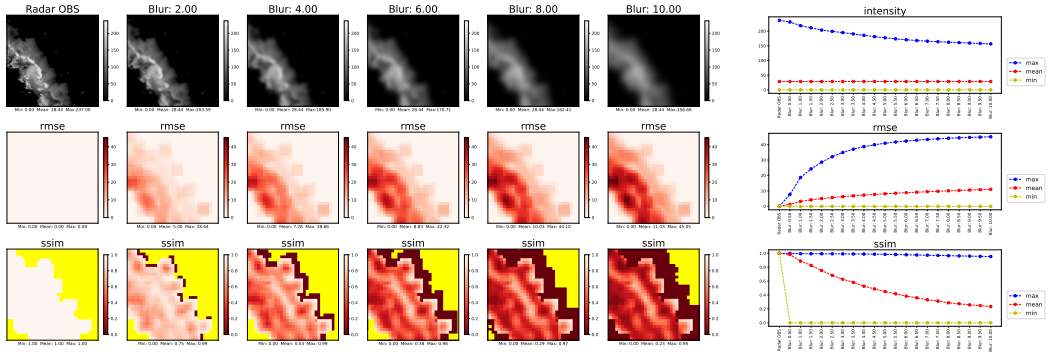
One issue with the results provided in Figures 4 to 6 is that it is hard to interpret the scale of the different metrics. What constitutes strong sharpness? We propose the following procedure to give more meaning to the values of the various metrics:

Proposed algorithm for the comparison of an image pair (original, estimate):

1. Take the original image and create a sequence of increasingly blurred copies, by applying Gaussian blur with σ in increasing from 0 to some chosen value, $\sigma_{\max}$.
2. Calculate all metrics for the blurred versions with respect to the original image.
3. Calculate all metrics for the estimated image with respect to the original image.
4. Generate a stats plot for the blurred images, and indicate the corresponding values for the estimated image by horizontal lines in the stats plots.
5. For each metric, find in the stats plots the x-value where the horizontal line from the estimate (min, mean or max) intersect the curve of values (min, mean or max) from the blurred images. This is called the **Equivalent Gaussian blur value**, $\sigma_{\text{equivalent}}$, for the estimated image for each metric.

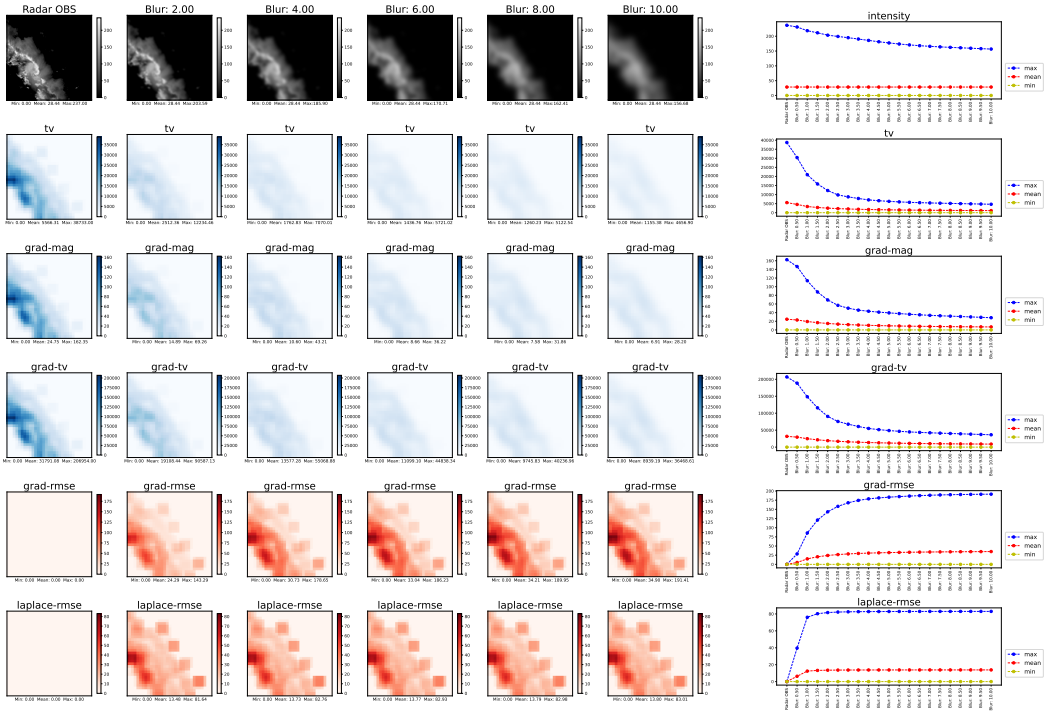
The Equivalent Gaussian blur value, $\sigma_{\text{equivalent}}$, for the estimated image represents for each metric the Gaussian blur operation that - when applied to the original image - would yield an identical metric value as the estimated image.

Figures 7 to 12 illustrate these ideas for a GREMLIN example. The first two steps are illustrated in Figures 7 to 9, each one showing - for one group of metrics - the original image, the blurred versions, their heatmaps and the corresponding stats plots.



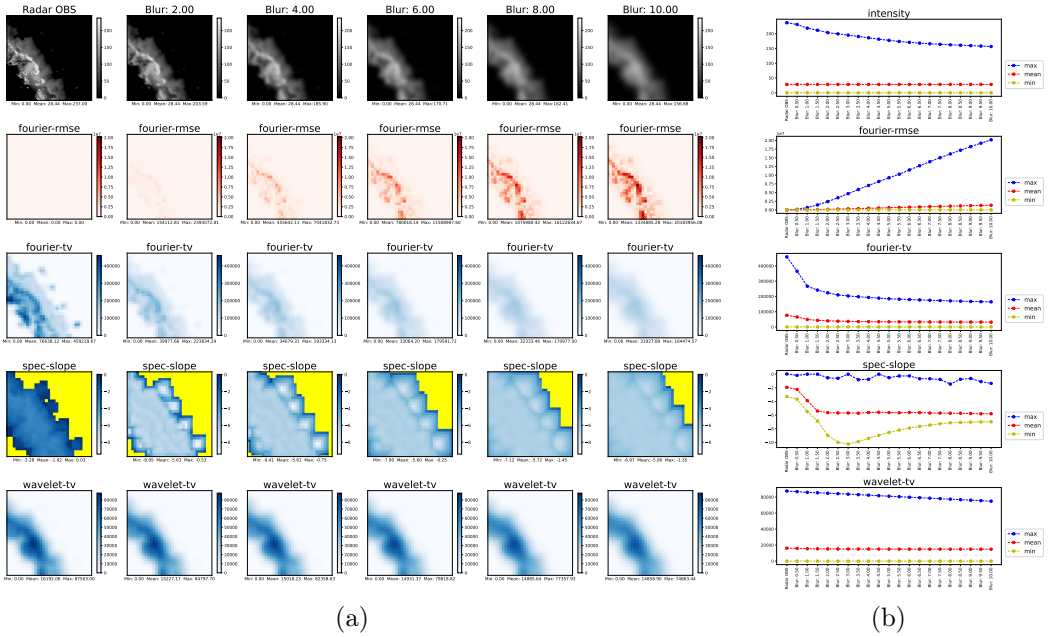
(a) (b)

Figure 7. Metric heatmaps for the original radar image, and for a progression of increasingly blurred versions of that image, using Gaussian blur with σ ranging from 0 to 10, and using metric group 1.



(a) (b)

Figure 8. Metric heatmaps for the original radar image, and for a progression of increasingly blurred versions of that image, using Gaussian blur with σ ranging from 0 to 10, and using metric group 2.



(a) (b)

Figure 9. Metric heatmaps for the original radar image, and for a progression of increasingly blurred versions of that image, using Gaussian blur with σ ranging from 0 to 10, and using metric group 3.

Figures 10 to 12 show Steps 3 to 4, namely tying the metric vales of the blurred images to those of the estimated image.

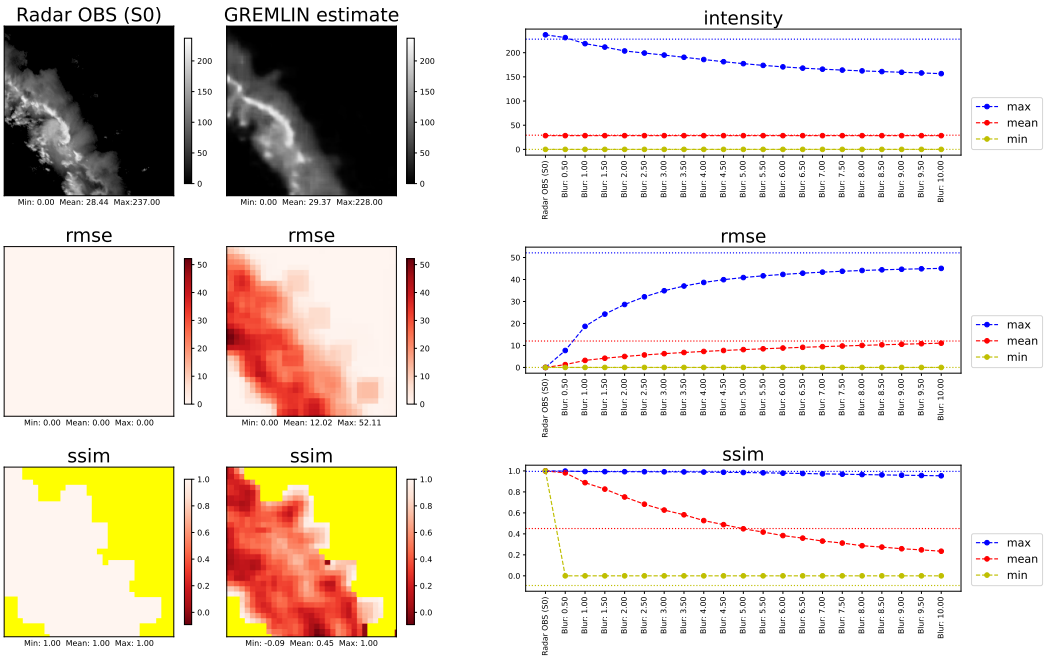
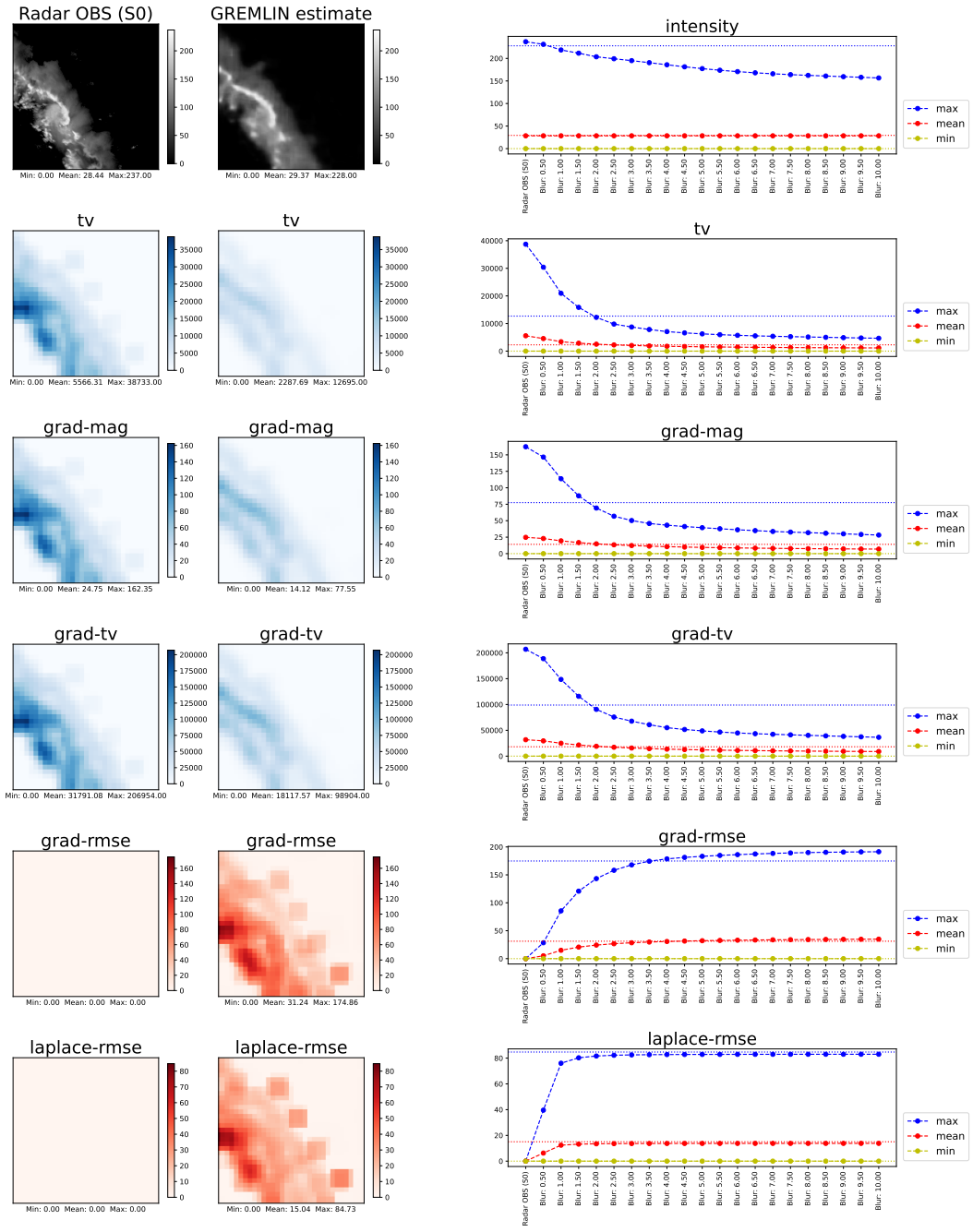
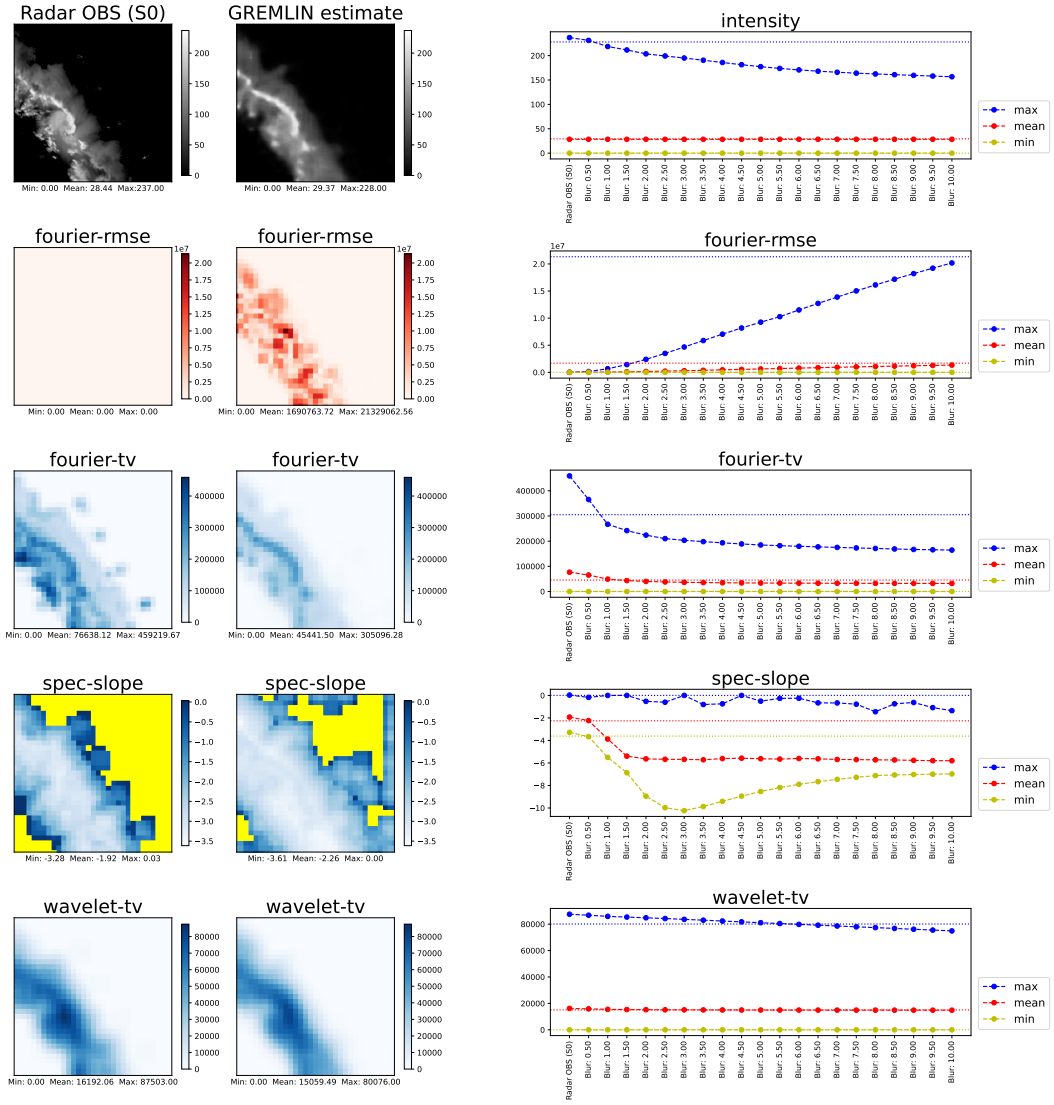


Figure 10. Comparison of observed radar and GREMLIN estimate using metric group 1: (a) heatmap comparison of observed and estimated image (same as Fig. 4(a)); (b) stats plots from a series of blurred images, along with horizontal lines indicating values of estimated image - all with respect to the original image. Thus the blue/red/yellow data points indicate the max/mean/min values of the increasingly blurred image, while the horizontal blue/red/yellow lines indicate the max/mean/min values of the GREMLIN estimate.



(a) (b)

Figure 11. Same as Fig. 10 for metric group 2.



(a) (b)

Figure 12. Same as Fig. 10 for metric group 3.

It remains to perform Step 5, namely to identify and interpret the equivalent Gaussian blur value for the estimated images from Figures 10 to 12. We choose here to use the **mean value** of the metrics for this step, which leads the following results (values are estimates based on visual inspection):

Table 1. Equivalent Gaussian Blur

	Intensity	RMSE	SSIM	TV
$\sigma_{\text{equivalent}}$	$\approx 0.$	> 10	≈ 5	≈ 2
	Grad-Mag	Grad-TV	Grad-RMSE	Laplace-RMSE
$\sigma_{\text{equivalent}}$	≈ 2.5	≈ 2	≈ 3.5	> 10
	Fourier-RMSE	Fourier-TV	Spec-Slope	Wavelet-TV
$\sigma_{\text{equivalent}}$	> 10	≈ 1	≈ 0.5	≈ 1

Observations: Focusing on the univariate sharpness metrics, the gradient-based sharpness metrics place the image estimate at about the blurriness of applying a Gaussian filter with σ around 2 to 2.5, while the spectral metrics consider the estimate to be much sharper, namely corresponding to σ around 0.5 to 1.

4 Important Properties of Sharpness Metrics for Meteorological Imagery

In this section, we discuss properties of sharpness metrics that are important when using them for meteorological imagery. Note that each application requires different properties. For example, in some applications, one may want a sharpness metric that is invariant to the normalization of the intensity values of an input image, while in other applications one may want the metric to be very sensitive to such changes. These properties should therefore not be seen as requirements. Instead, it is a list of properties that we believe users need to know about to select metrics for specific applications and to interpret their outputs of these metrics appropriately. The following are key questions that we think are important for users to ask about the sharpness metrics:

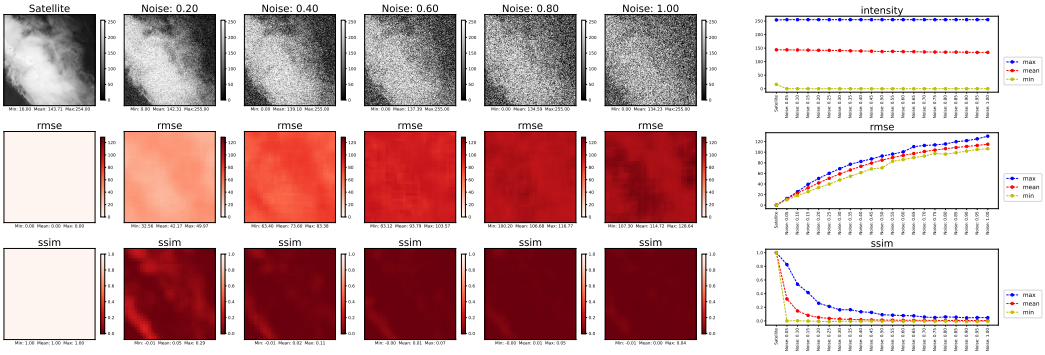
1. Which sharpness metric best measures the type of sharpness features critical for a specific application? For example, does the metric solely focus on edge transitions?
2. What kind of factors, e.g., image size, resolution, normalization of intensity values, impact the values of the sharpness metric?
3. How should the sharpness values be interpreted? Do the absolute values have meaning, or should one only consider the increase/decrease of values?

4.1 Response to adding noise

In this section we consider the question of how the various metrics respond to adding noise to an image, to better understand what changes to an image most strongly impact the sharpness values. In particular, if a sharpness metric is used in a loss function of a neural network to increase sharpness of an image, a metric that increases strongly with noise might push the network to simply add more noise to the image, which then needs to be counteracted by other means, e.g., increased weight on similarity metrics.

As illustrative example we use a satellite image of a cloud, gradually add noise to it, and track the values of the various metrics. The noise is added separately for each

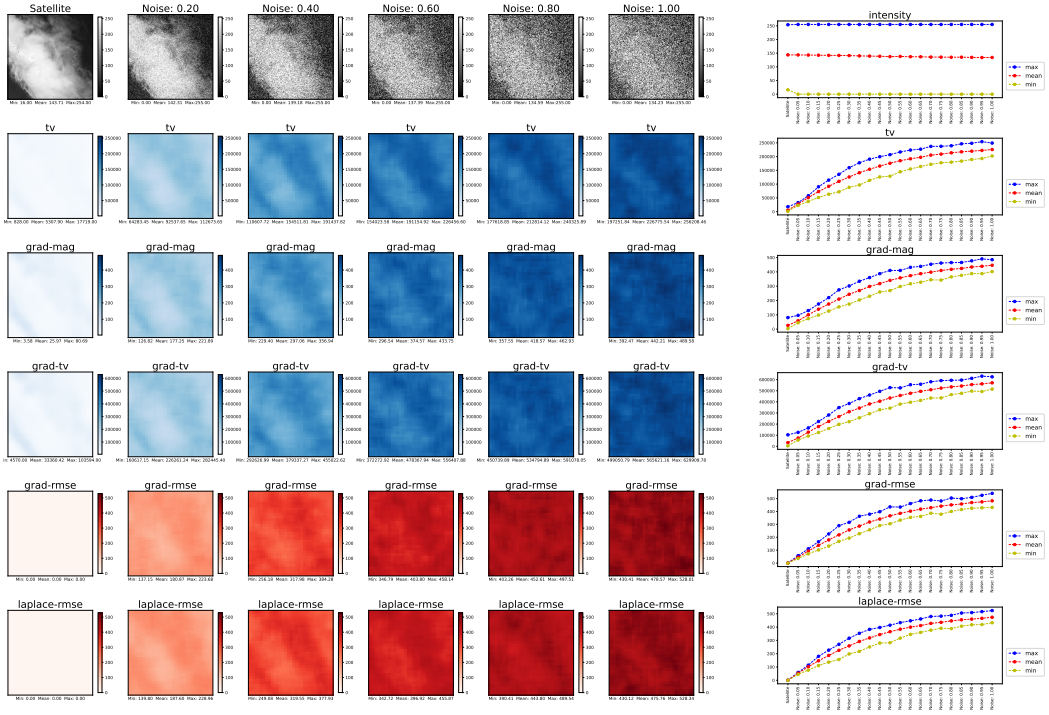
pixel, by drawing from a Gaussian distribution with σ set to be a factor times the maximal intensity of the original image. The factor is provided on top of the first row of images in Figures 13 to 15.



(a) (b)

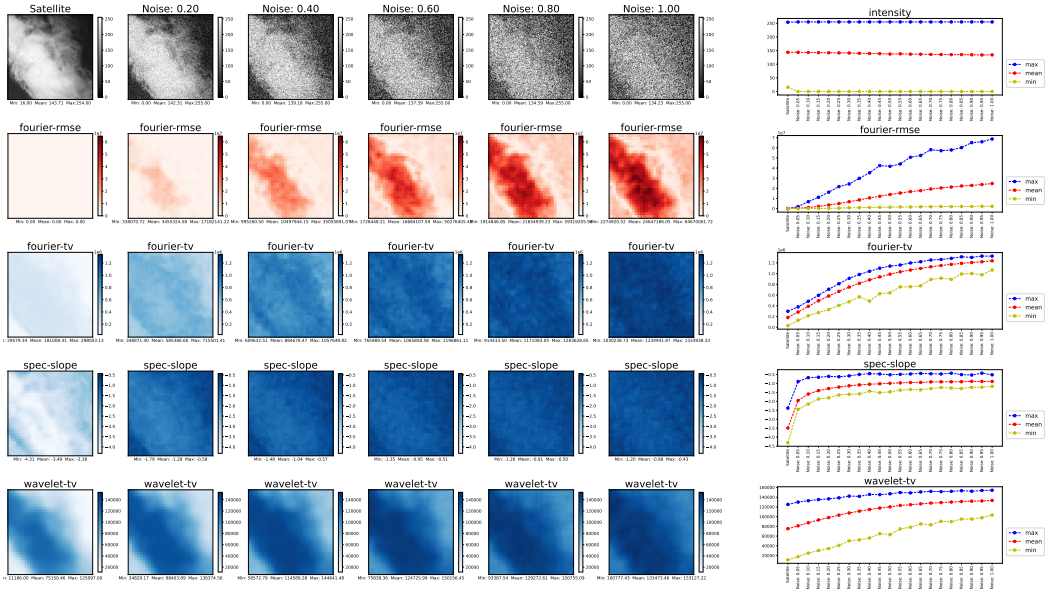
Figure 13. Response of metrics from group 1 to adding noise to a satellite image

Observations: All metrics increase with increasing noise in the image, but to varying degree. wavelet-tv seems to be last affected by the impact of noise, with the mean increasing by less than 100%, while all other sharpness metrics increase dramatically with noise. Thus, wavelet-tv stands out as being the most invariant to the addition of noise in an image.



(a) (b)

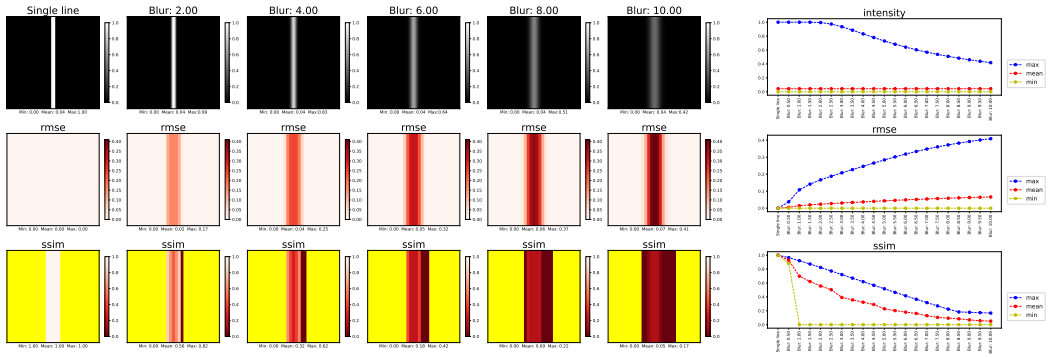
Figure 14. Response of metrics from group 2 to adding noise to a satellite image



(a) (b)

Figure 15. Response of metrics from group 3 to adding noise to a satellite image

4.2 How do the metrics respond to a single edge with varying degree of blurriness?



(a) (b)

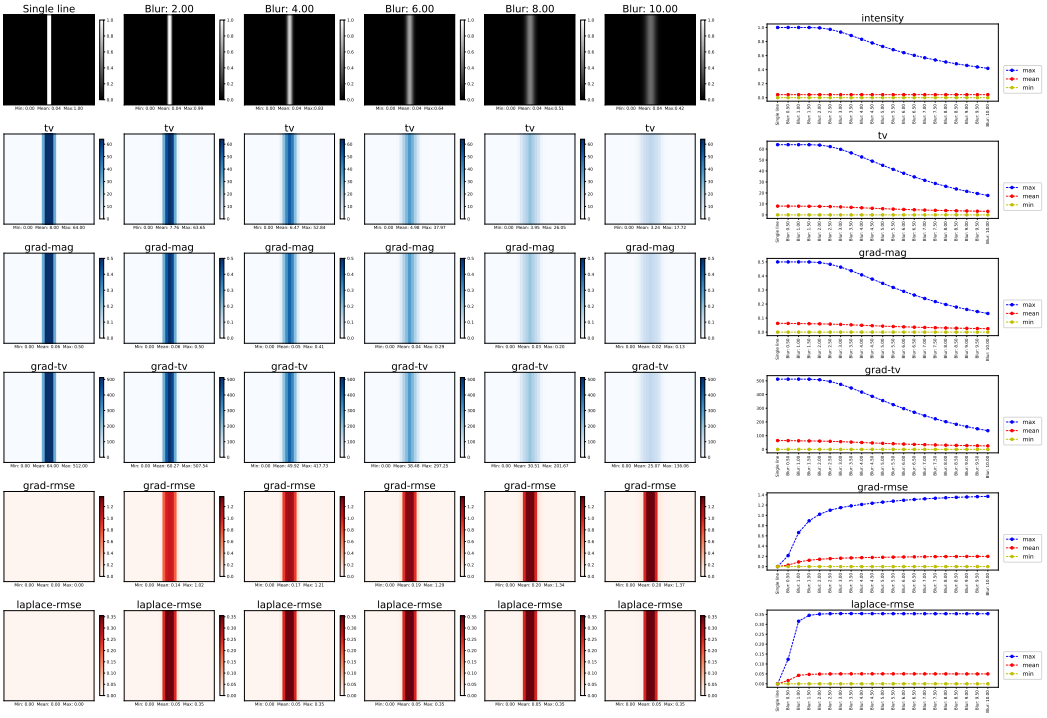
Figure 16. Response of metrics from group 1 to a single line that is gradually blurred

Observations: Figures 16 to 18 show a general trend of decreasing sharpness as the single edge is blurred (which is to be expected) but there are some interesting variations within that trend. We first note that for this example, a large proportion of the blocks in the heatmaps are viewing regions of the image that are uniformly black, and as such have no interesting features. Because of this, the mean and minimum statistics in the plots on the right are heavily weighted by these "minimal information" blocks, so we will primarily be looking at the max value curves for this analysis.

The similarity measures (RMSE and SSIM) exhibit very much the behavior that we would expect – as the line is blurred, the images become less and less similar and the metrics reflect that. It is interesting to note that because SSIM returns NaNs in the uniformly black region, which are then not included into the mean/min/max statistics, the mean statistic in this case is a mean over non-trivial blocks, and as such follows the max curve much more closely.

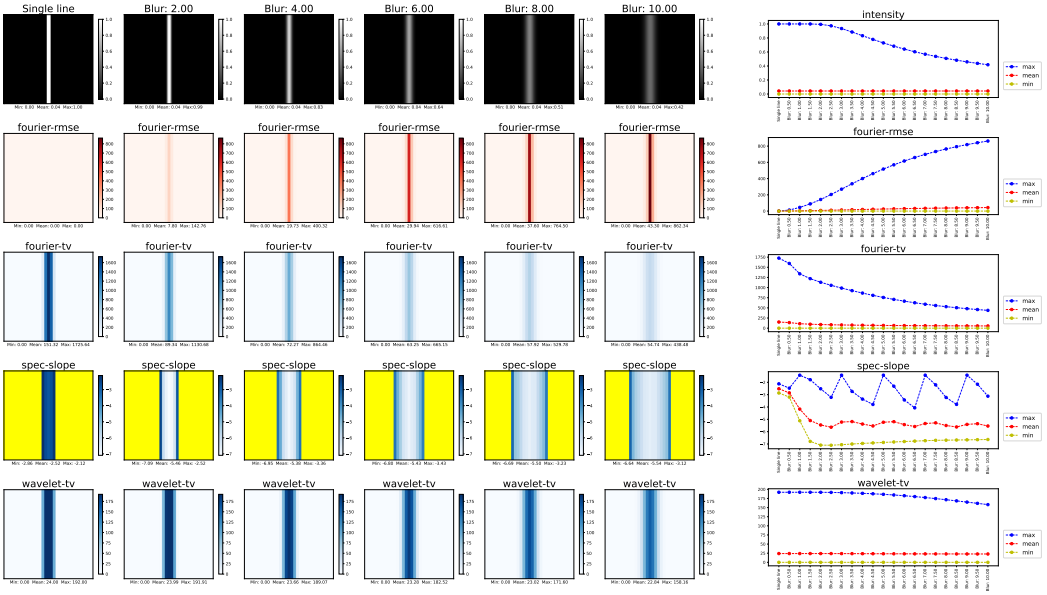
We next look at the gradient-based metrics in group 2. One immediate feature that is interesting to note is that all the univariate metrics in this group (TV, grad-mag, and grad-tv) are initially fairly consistent over relatively low levels of blur, and don't really start falling off until blur level 2.0. After this point, these metrics fall off quite evenly, and the heatmaps show the region around the line reducing in sharpness and diffusing outwards, which is precisely what we would expect. The bivariate metrics (grad-rmse and laplace-rmse) do not exhibit the same initial delay in response. Instead, they show the difference almost immediately (with laplace-rmse being particularly quick) and then the differences taper off. In looking at the heatmaps, we note that the strongest response is in the very center of the line, where we know from grad-mag the gradient magnitudes are becoming smaller, while there is a secondary (weaker) response along the edges of the initial line, where there are now larger gradients than when that region was uniformly dark.

Finally, we examine the spectral methods in group 3, which show the most variation within the same group. Fourier-rmse apperas to have similar trends to the bivariate gradient-based methods, but with a more even response throughout the period of blurring. From the heatmaps, this response appears to be very concentrated in the very center of the line. On the other hand, fourier-tv shows an interesting nonlinearity over the first couple levels of blur, with a larger than expected drop in sharpness between blur 0.5 and 1.0, perhaps indicating an initial stability to small levels of edge blur, and once



(a) (b)

Figure 17. Response of metrics from group 2 to a single line that is gradually blurred



(a) (b)

Figure 18. Response of metrics from group 3 to a single line that is gradually blurred

this initial stability has been bypassed, the fall-off resembles TV and grad-tv. Wavelet-tv displays almost no change in the overall statistics in the sensitivity plot on the right, but we can see in the heatmaps some slow diffusion of sharp regions away from the high central ridge that remains in the increasingly blurry line. This is particularly good information, as it tells us that wavelet-tv does not respond strongly to edge blurriness.

Spectral slope is similar to SSIM in that because the uniformly dark areas return NaNs, all three of mean, min, and max are relevant for this analysis. The most notable feature of the sensitivity plot is the shark-tooth pattern most apparent in the max (and to a lesser extent in the mean). Looking at the heatmaps, we note that the highest response values are right at the farthest extents of where the blurred intensity values are no longer uniformly 0 – this is a result of spectral slope being intensity-invariant, and thus yielding strong results even in very dark regions. This leads to the hypothesis that as these blur values spread through individual blocks, there are particular locations within those blocks that lead to stronger responses. If we look at the mean response (which is more resistant to this shark-tooth pattern), we see that the sharpness falls off initially, then more or less stabilizes at blur 2.0 – this matches with what we have seen elsewhere of spectral slope being quite sensitive to small levels of blur.

4.3 Computational Complexity

The computational complexity of each of these metrics can be expressed in terms of both big O notation, which expresses how well the computational complexity scales with increasing numbers of pixels in each block, and in terms of actual wall clock time to run on a single representative example image. Both expressions are represented in Table 2, and more details on the big O complexity for each metric can be found in Appendix Appendix A. Note that in Table 2, the n references the number of pixels in a given block when computing heatmaps – thus, if the edge length of a heatmap block is doubled, the n in column 2 will increase fourfold.

The wall clock computation times were intended to be representative of how we see these metrics being used in practice. In both the 64×64 and 128×128 case, each metric heatmap is computed 5 times using block sizes of 8×8 and 64×64 respectively. In these computations, there are numerous other steps that are in common between the two image sizes, such as subsetting and collecting results from individual heatmap blocks, which likely explains why we do not see as large an increase as might be predicted when moving from 64×64 to 128×128 images.

Table 2. Computational Complexity of Metrics

Metric	Big O	64×64 wall clock	128×128 wall clock
RMSE	$O(n)$	0.0103 sec	0.0113 sec
SSIM	$O(n)$	0.1379 sec	0.1580 sec
TV	$O(n)$	0.0260 sec	0.0295 sec
Grad-mag	$O(n)$	0.0447 sec	0.0509 sec
Grad-TV	$O(n)$	0.0414 sec	0.0464 sec
Grad-RMSE	$O(n)$	0.0385 sec	0.0446 sec
Laplace-RMSE	$O(n)$	0.0197 sec	0.0223 sec
Fourier-RMSE	$O(n \log n)$	0.0852 sec	0.0946 sec
Fourier-TV	$O(n \log n)$	0.0807 sec	0.0908 sec
Spec-slope	$O(n \log n)$	0.9053 sec	1.2101 sec
Wavelet-TV	$O(n)$	0.1662 sec	0.1740 sec

Note: all wall clock times are an average from five similar computations with the same base image.

4.4 Effect of shifting and scaling intensity range

For almost all of these metrics, the effect of scaling the intensity range will be to proportionally scale the output value, while shifting the intensity range will not change the output value. The exceptions to the rule are that SSIM, Spectral Slope, and Wavelet-TV are affected by shifting the intensity range, and that SSIM and Spectral Slope are invariant to scaling the intensity range.

Let us first address the effect of shifting the intensity range. Almost all of the metrics we utilize are based in some way off of pixel differences, whether between images (as in RMSE) or between pixels (as in TV or gradient-based methods). Whenever we are taking a difference of two quantities, if both of those quantities have had the same constant added or subtracted to them, then those constants cancel and the difference ultimately remains the same. This establishes that RMSE, TV, Grad-mag, Grad-TV, Grad-RMSE, and Laplace-RMSE are all invariant to shifts in intensity. Next, we note that if we take the FFT of a shifted function, the output will shift by some amount that depends

only on the original shift, but not the function – that is,

$$\mathrm{FFT}(f(x) + A) = \mathrm{FFT}(f(x)) + B$$

where B is dependent only on A. This establishes that Fourier-RMSE and Fourier-TV are invariant to shifts.

On the other hand, we observe that as spectral slope is calculated as the slope of a line fit to the log / log plot of the FFT, in the log the shift in Fourier space turns into a nonlinear scaling, which can affect the slope of the fitted line. SSIM is based on the product of three components: luminance, contrast, and structure. While contrast and structure are based on the variances and covariances (respectively) of the inputs, luminance is explicitly based on their means in a way that adding or subtracting a constant will affect the output in nonlinear ways. Similarly, Wavelet-TV is based on the coefficients of the Haar wavelet transform, both the detail and approximation coefficients. The detail coefficients of a Haar wavelet transform are based on pixel differences, which will be invariant to shifts; however, the approximation coefficients capture information about the mean, and as such will be affected by shifting the mean, and thus Wavelet-TV will not be invariant to such shifts.

We can now address the effects of scaling the intensity range by a constant. We will provide here details for the two measures that are invariant to scaling of intensity range, while the details for all other metrics can be found in Appendix Appendix B.

The SSIM of two image patches x and y is given by

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

where μ_x, μ_y are the means of x and y respectively, σ_x, σ_y are the standard deviatiosn of x and y, σ_{xy} is the covariance of x and y, and c_1 and c_2 small constants.

If we scale x and y by some common factor k, this becomes

$$\mathrm{SSIM}(kx, ky) = \frac{(k^2\mu_x\mu_y + k^2c_1)(k^2 2\sigma_{xy} + k^2c_2)}{(k^2\mu_x^2 + k^2\mu_y^2 + k^2c_1)(k^2\sigma_x^2 + k^2\sigma_y^2 + k^2c_2)} = \frac{k^4}{k^4}\,\mathrm{SSIM}(x, y) = \mathrm{SSIM}(x, y),$$

where it is important to note that c_1 and c_2 are proportional to the square of the dynamic range of the input images, so will also scale by k^2 if x and y are scaled by k.

On the other hand, for spectral slope, we note that the FFT is a linear operation, so $\mathrm{FFT}(kX) = k\,\mathrm{FFT}(X)$, as is the polar averaging of the power spectrum to obtain a set of points $\{P_i\}$, or with scaling $\{kP_i\}$. At that point, to compute spectral slope we take the log of these averaged values, and by log rules we see that

$$\log(kP_i) = \log(P_i) + \log(k).$$

That is, the scaling turns into a vertical shift in these points, which does not affect the slope of the line that is ultimately fit to these lines, and thus spectral slope is invariant to intensity scaling. For more details, see Vu et al. (2011).

4.5 Response to change in resolution

The response to changing the resolution (or block size, when computing heatmaps) of an image separates these metrics into two distinct groups: those that involve Total Variation, and those that do not. Most of the metrics other than total variation utilize a spatial mean as a way of collating the statistic down into a single number, and because this mean is normalized by the number of pixels included in it, the metric becomes invariant to the size of block or resolution of image. The metrics for which this holds true are: RMSE, Grad-mag, Grad-RMSE, Laplace-RMSE, and Fourier-RMSE.

Two special cases are SSIM and Spectral Slope. SSIM utilizes even smaller windows within the blocks or images, with the output value being an average across those windows. However, even as the size of those windows increases, the statistic computed on each window is based on the mean, standard deviation, and covariance between corresponding windows, all of which are normalized quantities, so the SSIM value should be (theoretically) invariant to window size. On the other hand, Spectral Slope uses a fundamentally quite different approach to summarizing the data, by first doing polar averaging in spectral space, and then fitting a line to the log of the resulting data points. However, the line fitting operation is invariant to the number of points, so long as the data are consistent, so spectral slope is also ultimately invariant to change in resolution.

While these operations are all theoretically invariant to change in resolution, it is worth noting that by changing resolution, one is also changing the amount of information present in the image. This may end up changing the value of the metric in some way, but overall the results should be similar across resolutions.

On the other hand, all the metrics utilizing total variation (TV itself, as well as Grad-TV, Fourier-TV, and Wavelet-TV), utilize a sum to collate spatial information into a summary statistic. Because these sums are unnormalized, the ultimate value of a TV metric is expected to increase linearly with the number of pixels, and thus as the square of edge length. This is true of TV, Grad-TV, Fourier-TV, and Wavelet-TV, as even though the metrics other than TV itself are acting on derived products (the gradient magnitude map, the power spectrum, or the wavelet coefficients) those derived products all have similar numbers of elements as there are pixels in the original image.

5 Vignettes

This section provides several vignettes that illustrate the use of the sharpness metrics for meteorological applications.

5.1 Bias Correcting Subseasonal Forecasts - A Vignette by Maria Molina

Forecast uncertainties from numerical models at lead times of three to four weeks (i.e., subseasonal) can be reflected as blurriness (Molina et al., 2023), an undesirable property for extreme events particularly along coasts, mountain ranges, and urban-to-rural population transitions where finer-scale details are needed by stakeholders. Two U-nets were trained to bias correct 2-m temperature week 3 forecasts from a hindcast (1999-2020) produced with the Community Earth System Model version 2 (CESM2; Richter et al., 2022). One of the U-nets was trained using one input channel which was the CESM2 week 3 hindcast of 2-m temperature, while the other U-net was trained using four input channels consisting of the same CESM2 field with normalized latitude, longitude, and model terrain height as additional channels. The corresponding ECMWF 5th generation reanalysis (ERA5; Hersbach et al., 2020) was used as the ground truth and MSE was used as the loss function run for 20 epochs.

Here we use gradient magnitude (grad-mag) and Laplace-RMSE to assess sharpness in the CESM2 hindcast and U-net bias corrections as compared to ERA5 for the southern part of South America, a region that contains complex topography, coastlines, and several cities.

Note that we tried several different bivariate sharpness metrics in addition to 'laplace-rmse' from the GitHub repo, including 'grad-ds', 'grad-rmse', 'laplace-rmse', 'hist-int', and 'wavelet-similarity.' All of them distinguished between the sharpness of CESM2 and the ML bias-corrected images, but only 'laplace-rmse' provided a clear signal to distinguish between the two ML bias-corrected images (4-channel vs 1-channel).

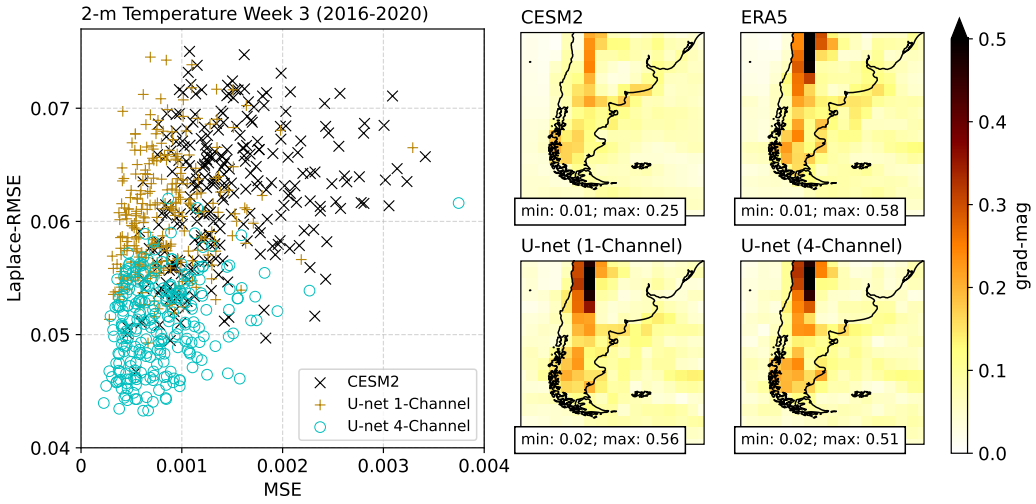


Figure 19. Comparison of 1-channel and 4-channel U-net bias correction of CESM2 2-m temperature week 3 forecasts. Left panel illustrates trade-offs between MSE (x-axis) and Laplace-RMSE (y-axis) for CESM2 and the two U-nets evaluated against ERA5 (ground truth), as indicated in the legend. Panels on the right show grad-mag for the same sample in CESM2, ERA5, and the two U-nets, with the sample's min and max grad-mag indicated on the respective image.

Plotting MSE against Laplace-RMSE shows that including terrain height information and geographic coordinates as input channels for the U-net (4-channel) helps reduce trade-offs between accuracy and sharpness during bias correction (Fig. 19; left panel). Grad-mag for a specific sample of 2-m temperature shows that both U-nets increase sharpness as compared to the original forecast produced by CESM2, and have a much closer resemblance to the ground truth's sharpness (Fig. 19; right panels). Interestingly, grad-mag indicates that both U-nets have comparable sharpness (confirmed with grad-RMSE, not shown), while Laplace-RMSE indicates that sharpness is greater in the 4-channel U-net, likely because Laplace-RMSE incorporates more spatial information than grad-mag by using the divergence of the gradient rather than the magnitude of the gradient.

5.2 Evaluating Sharpness of Multiple Datasets - A Vignette by Jason Stock

In the previous sections we detail how individual metrics can be used to evaluate the sharpness for a given sample. However, when considering if one model's output or dataset is sharper than another, we need a method that allows for a comprehensive comparison. To address this, we augment samples from the entire GREMLIN dataset (1,798 total samples) to emulate different model output, showing a comparison of their distributions from a single sharpness metric and make statistical conclusions.

Similar to the experiments in Subsection 3.4, we augment the samples using Gaussian blur with an increase in standard deviation, specifically $\sigma = 5, 10, 20$. This effectively yields four different datasets that we will compare. Figure 20 illustrates this result for a single sample aligned next to the ground truth MRMS. Thereafter, we compute our sharpness metric for all samples in each of the datasets. For evaluation purposes we choose the Mean Gradient Magnitude ('grad-mag') for our sharpness metric, but any of the other scalar metrics could be subsisted.

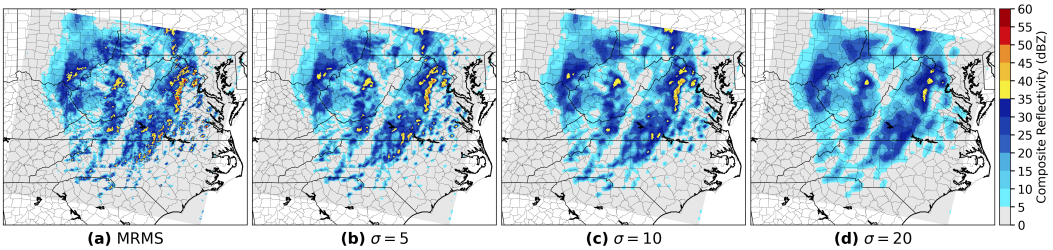


Figure 20. An individual sample with Gaussian blur applied for different σ levels.

We plot the kernel density estimation (KDE) for each of the datasets to visualize a smoothed representation of the underlying distribution. In Figure 21, we see that MRMS is positively skewed with a greater maximum sharpness relative to the other σ levels. Furthermore, the sharpness values decrease along with the spread for higher levels. Note the mean and standard deviations are associated with the sharpness metric for each dataset. With a direct comparison using Welch's independent samples t-test, we conclude that MRMS demonstrates a statistically significant improvement in sharpness over the other datasets (e.g., comparing to $\sigma = 10$, t-statistic $= 31.49$, p-value < 0.001). When making relative comparisons, it is important to keep a consistent scale across datasets. However, the process we define can simply be applied to other datasets, such as output from individual models, along with different sharpness metrics.

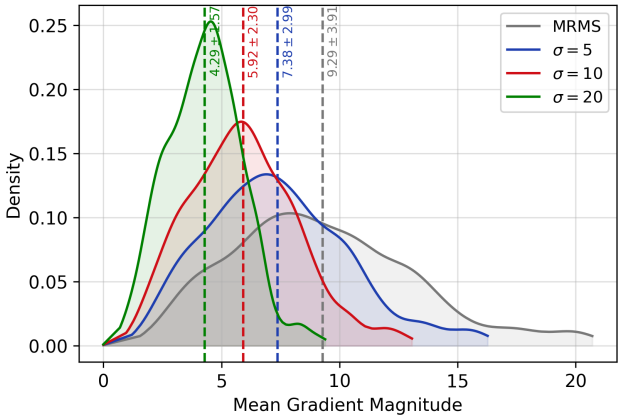


Figure 21. Kernel density estimation of 'grad-mag' computed over all data samples in each dataset defined by σ. The dashed line represents the mean and standard deviation.

5.3 Studying how Sharpness Varies as a function of ML architecture - A Vignette by Michael Yu and Amy McGovern

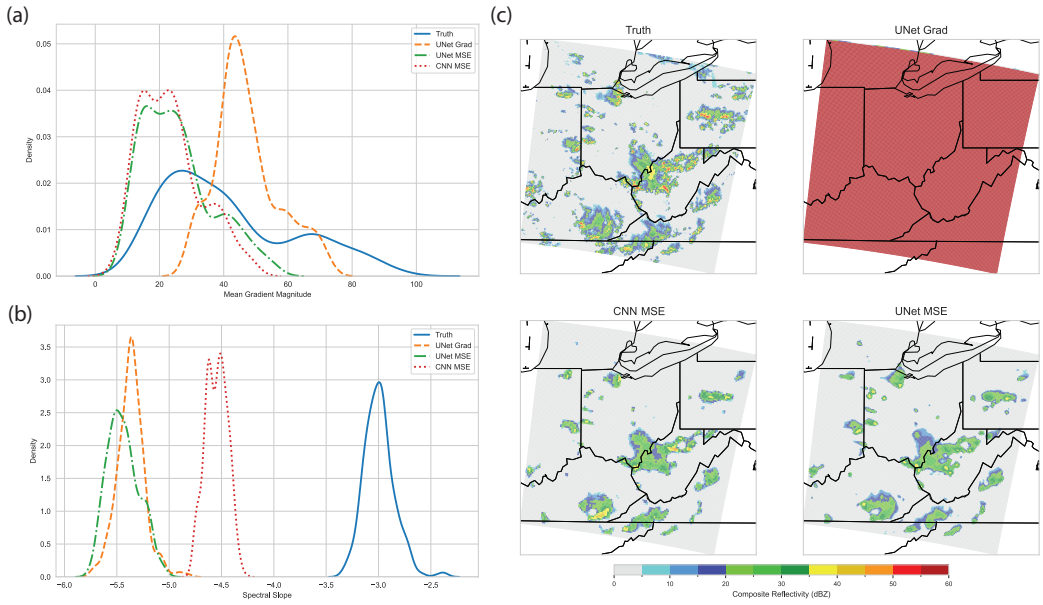


Figure 22. Comparison of the output of UNet MSE, UNet Grad, and CNN MSE for GREMLIN MRMS Reflectivity predictions along with the truth values. (a) Density plot of the mean gradient magnitude (x-axis) shows the magnitude values for all predictions and truth. (b) Density plot of the spectral slope (x-axis) shows the slope values for all predictions and truth. (c) Case study example of MRMS Reflectivity predictions and truth from 2019-04-19 21:59.

With sharp and accurate models as a goal for most machine learning models for weather prediction, we examine the effects of changing both model architecture and loss function. We plan a more in-depth analysis in a follow-up paper. In Figure 22, we examine the effect of varying ML model architectures and loss functions using the sharpness metrics mean gradient magnitude (grad-mag) and spectral slope (spec-slope).

Three models, a UNet with MSE loss (UNet MSE), a UNet with gradient magnitude MAE (UNet Grad), and a CNN—UNet architecture without the skip connections—with MSE loss (CNN MSE), were trained on GREMLIN CONUS2 data (Hilburn et al., 2020) to predict MRMS reflectivity from ABI & GLM data. We then examined the measured sharpness of each model on a specific test day. The ML predictions and the truth for April 19, 2019 are shown in Figure 22c. As the case study shows, UNet Grad was a very poor predictor, since its loss function only penalizes the gradient of the image, not the actual values. UNet Grad is therefore free to add a large constant to the image values without any penalty. However, since this model is trained to *only* focus on modeling the gradient correctly, it tends to add a constant similar to the mean grad-mag of the ground truth (Truth) distribution to all outputs. This results in predictions with higher sharpness values and lower accuracy because the shape of the UNet Grad distribution differs from the Truth, which is why we included it in this study.

Plotting the grad-mag (Figure 22a) shows that the UNet MSE and CNN MSE have similar distributions whereas the UNet Grad has a higher mean gradient magnitude with values closer to the center of the Truth distribution. Overall, the UNet Grad performs

poorly in terms of capturing the shape of grad-mag (Panel a), in addition to performing poorly in actual values (Panel c), which was unexpected.

Plotting the spectral slope (Figure 22b) shows that the UNets have similar spec-slope distributions whereas the CNN MSE has noticeably higher spec-slope values than both UNets, with all three prediction distributions being lower than the truth distribution. Here, the CNN MSE's higher sharpness (spec-slope) values correspond to closer predictions to the truth in the case study (Figure 22c), so it performs best in terms of spectral slope.

In this preliminary study, the two sharpness metrics behave differently as a function of model architecture and loss function. Grad-mag was more affected by the loss function whereas spec-slope was more affected by the model architecture. In one case (grad-mag) higher sharpness values led to less accurate predictions, pointing to a need to not solely evaluate models by sharpness.

6 Contributions and Future Work

Obviously, we have only scratched the surface here of how to best measure sharpness for meteorological imagery. Nevertheless, we hope that this study serves as the starting point for a much larger discussion on this topic, realizing that different metrics are most suitable for different applications and purposes.

The key contributions of this paper include:

1. Starting the important discussion of how to best measure sharpness for meteorologicaly imagery.
2. Identifying three groups of interpretable metrics that may be a good first basis to evaluate similarity and sharpness of meteorological imagery.
3. Providing a code base for the community to quickly adopt these metrics for their own research, including visualizations of local sharpness (heatmaps).
4. Developing a protocol to use to interpret the metric values for a given pair of imagery, consisting of our proposed stats plots in combination with a calibration method that uses gradual application of blur to the original image (equivalent blur value).
5. Analysis of some of the properties of the different metrics - although, as pointed out in future work below, that study is far from finished.
6. Providing vignettes that demonstrate how these metrics can be used for practical applications.

We suggest to expand this study by exploring the following topics:

1. A more in-depth study of which kinds of sharpness the various metrics primarily focus on, e.g. from edges vs. texture.
2. Adopting a better way to avoid NaNs as output of metrics, such as discussed in Vu et al. (2011) for spec-slope.
3. Adding some of the metrics discussed in Section 2.2 that we dropped for this study.

Another important topic - one that we did not even touch on here - is the question of how to effectively use these metrics to *train* neural networks, rather than just to *evaluate* them, as indicated in the Vignette in Section 5.3. We plan to explore this topic in a follow-up paper.

7 Data Availability Statement

The code for metrics and heatmaps will soon be available at `https://github.com/ai2es/sharpness/`. It also includes the imagery used for testing here. The only data not provided are those related to the vignettes (Section 5).

Acknowledgments
This material is based upon work supported by the National Science Foundation under Grant No. ICER-2019758.

References

Bonev, B., Kurth, T., Hundt, C., Pathak, J., Baust, M., Kashinath, K., & Anandkumar, A. (2023). Spherical fourier neural operators: Learning stable dynamics on the sphere..

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., ... others (2020). The era5 global reanalysis. *Quarterly Journal of the Royal Meteorological Society*, *146*(730), 1999–2049.

Hilburn, K. A., Ebert-Uphoff, I., & Miller, S. D. (2020). Development and interpretation of a neural-network-based synthetic radar reflectivity estimator using goes-r satellite observations. *Journal of Applied Meteorology and Climatology*, *60*(1), 3–21.

Imatest. (2023). *Sharpness: What is it and how it is measured. documentation – v23.1.* Retrieved from `https://www.imatest.com/support/docs/23-1/sharpness/` (Accessed on 10/26/2023)

Molina, M. J., Richter, J. H., Glanville, A. A., Dagon, K., Berner, J., Hu, A., & Meehl, G. A. (2023). Subseasonal representation and predictability of north american weather regimes using cluster analysis. *Artificial Intelligence for the Earth Systems*, *2*(2), e220051.

Richter, J. H., Glanville, A. A., Edwards, J., Kauffman, B., Davis, N. A., Jaye, A., ... others (2022). Subseasonal earth system prediction with cesm2. *Weather and Forecasting*, *37*(6), 797–815.

SLR Lounge. (2023). *Slr lounge / sharpness.* Retrieved from `https://www.slrlounge.com/glossary/sharpness-photography-definition/` (Accessed on 05/05/2023)

Vu, C. T., Phan, T. D., & Chandler, D. M. (2011). s_3: a spectral and spatial measure of local perceived sharpness in natural images. *IEEE transactions on image processing*, *21*(3), 934–945.

Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, *13*(4), 600–612.

Wee, C.-Y., & Paramesran, R. (2008). Image sharpness measure using eigenvalues. In *2008 9th international conference on signal processing* (pp. 840–843).

Zhang, R., Isola, P., Efros, A. A., Shechtman, E., & Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 586–595).

Appendix A Computational Complexity of Metrics

Throughout this section, we will refer to computational complexity in terms of the number of pixels in an image (or block) as n. It is important to note that this n grows as the square of the side length, so to translate these to computational complexity in terms of a side length ℓ, one need only make the substitution $n = \ell^2$.

A1 RMSE

The computational complexity of RMSE is $O(n)$, as the computation

$$\mathrm{RMSE}(X, T) = \sqrt{\frac{1}{n}\sum_{i=0}^{n-1}(x_i - t_i)^2}$$

involves only n operations.

A2 SSIM

The computational complexity for SSIM for a given window with p points is $O(p^2)$, but because window sizes are typically fixed, and the number of windows grows linearly with the number of points in an image, the practical complexity is $O(n)$.

The SSIM for a particular pair of windows x and y is given by

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

where μ_x and μ_y are the input window means, σ_x and σ_y are the input window variances, σ_{xy} is the covariance of the input windows, and c_1 and c_2 are constants. In this computation, computing the means and variances are each $O(n)$ operations, but computing the covariance of the two windows is a $O(n^2)$ operation, as each pair of pixels in the windows must be compared.

A3 Total Variation

Total variation involves taking two differences per pixel, and as such is an $O(n)$ operation.

A4 Mean gradient magnitude

Of note for this and following sections, the convolution operation for an image against a small filter is $O(n)$ complexity. Therefore, the mean gradient magnitude involves two Sobel filter convolutions (each $O(n)$), a magnitude computation for each pixel which is $O(n)$, and computing a mean across the image, which is $O(n)$. Thus, computing mean gradient magnitude is $O(n)$.

A5 Gradient RMSE

The first step of computing gradient RMSE is computing the gradient magnitude map for each image, which as discussed in mean gradient magnitude above, is $O(n)$, and the second step is computing RMSE, which is also $O(n)$. Thus, gradient RMSE is an $O(n)$ computation.

A6 Gradient Total Variation

Like gradient RMSE, this is the composition of computing a gradient magnitude map and computing total variation, both of which are $O(n)$ operations, so computing gradient total variation is also an $O(n)$ operation.

A7 Laplace RMSE

The Laplacian map is computed by OpenCV via two consecutive applications of the Sobel filter in both the horizontal and vertical directions (for four total filter convolutions), which is still an $O(n)$ operation. As discussed, computing RMSE is an $O(n)$ operation as well, so computing Laplace RMSE is also an $O(n)$ operation.

A8 Fourier RMSE

Computing the Fourier RMSE involves as a first step computing the 2-dimensional fast Fourier transform (FFT) of each of the input images. This is a $O(n \log n)$ operation, and the successive steps (taking the absolute value of the resulting complex-valued spectra, then taking the RMSE) are $O(n)$, so in total, computing Fourier RMSE is an $O(n \log n)$ operation.

A9 Fourier TV

Much like Fourier RMSE, the bottleneck in computing Fourier TV is computing the FFT, so Fourier total variation is also $O(n \log n)$.

A10 Spectral Slope

Computing spectral slope involves a number of steps. First, we must compute the power spectrum (absolute value of the 2D FFT) of the input, which is an $O(n \log n)$ computation. Then, we compute a polar average via coordinate interpolation, which is an $O(n)$ operation, and find the line of best fit on the log log transform of the resulting $\sqrt{n}/2$ points, which is $O(\sqrt{n})$, so the total time complexity is $O(n \log n)$.

A11 Wavelet Total Variation

The wavelet decomposition process for finite-dimensional filters (like the Haar wavelets used in our implementation) is computed in $O(n)$ time, and total variation is (as we know) $O(n)$, so wavelet total variation can be computed in $O(n)$ time.

Appendix B Details on Effect of Scaling Intensity

B1 RMSE

The root mean square error between images X and T on a common domain D is given by

$$\mathrm{RMSE}(X,T) = \sqrt{\sum_{i\in D}(X_i - T_i)^2}.$$

If we scale X and T by some common factor k, we see that

$$\begin{aligned}\mathrm{RMSE}(X,T) &= \sqrt{\sum_{i\in D}(kX_i - kT_i)^2} \\ &= \sqrt{\sum_{i\in D} k^2(X_i - T_i)^2} \\ &= k\sqrt{\sum_{i\in D}(X_i - T_i)^2} \\ &= k\,\mathrm{RMSE}(X,T).\end{aligned}$$

B2 Total Variation

The TV of an image X is given by

$$\mathrm{TV}(X) = \sum_{i,j} |X_{i,j} - X_{i+1,j}| + |X_{i,j} - X_{i,j+1}|.$$

If we scale X by k, we get that

$$\begin{aligned}\mathrm{TV}(kX) &= \sum_{i,j} |kX_{i,j} - kX_{i+1,j}| + |kX_{i,j} - kX_{i,j+1}| \\ &= \sum_{i,j} k|X_{i,j} - X_{i+1,j}| + k|X_{i,j} - X_{i,j+1}| \\ &= k\sum_{i,j} |X_{i,j} - X_{i+1,j}| + |X_{i,j} - X_{i,j+1}| \\ &= k\,\mathrm{TV}(X).\end{aligned}$$

B3 Mean Gradient Magnitude

The gradient magnitude at a pixel P is given by

$$I(P) = \sqrt{I_x(P)^2 + I_y(P)^2},$$

where I_x and I_y are directional derivatives. By the linearity of the derivative operator, $I_x(kP) = kI_x(P)$ and similarly for I_y, so

$$I(kP) = \sqrt{I_x(kP)^2 + I_y(kP)^2} = \sqrt{k^2 I_x(P)^2 + k^2 I_y(kP)^2} = k\sqrt{I_x(P)^2 + I_y(P)^2} = kI(P).$$

The mean gradient magnitude (grad-mag) is simply the mean of this quantity, and the mean is a linear operation, so

$$\text{grad-mag}(kX) = k\,\text{grad-mag}(X).$$

B4 Gradient TV

By what we described in the first part of the above section, the gradient magnitude map scales proportional to the scaling of X, and we already know that TV scales proportionally, so gradient TV also scales proportionally.

B5 Gradient RMSE

Similar to the argument for gradient TV, because RMSE scales proportionally, we know that gradient RMSE scales proportionally.

B6 Laplace RMSE

The Laplacian at a point P is defined as

$$L(P) = I_{xx}(P) + I_{yy}(P),$$

where I_{xx} is two applications of the horizontal gradient operator, and because the gradient operator is linear, $I_{xx}(kP) = kI_{xx}(P)$. Thus, the Laplacian operator is also linear, so Laplace RMSE will also scale proportionally to the input image scaling.

B7 Fourier RMSE

We note that because the Fourier transform is based on an integral of the input against a kernel, and we can pull scalar multiplication out of integrals, we have that $\mathrm{FFT}(kX) = k\,\mathrm{FFT}(X)$. Thus, because RMSE scales proportionally, Fourier RMSE also scales proportionally.

B8 Fourier TV

Similarly to Fourier RMSE, because the FFT and TV both scale proportionally to their input, Fourier TV scales proportionally.

B9 Wavelet TV

The Haar wavelet transform is based on taking directional differences (for the detail coefficients) and averages (for the approximation coefficients) between pixels in an image, and both differences and averages respect scalar multiplication. Therefore, the coefficients of the Haar wavelet transform (both detail and approximation) will be scaled proportionally to the input, and as wavelet TV simply adds all of those coefficients, it will also be scaled proportionally.