

Visual Analytics for Underwater Super Resolution

Giovanni Ficarra

1 Introduction

During my PhD, I am dealing with underwater image super resolution, that is “the process of recovering high-resolution images from low-resolution images” [1]. In this field, the most used metrics to evaluate the performances of the proposed neural networks are SSIM (Structural SIMilarity) and PSNR (Peak Signal to Noise Ratio), but there is not a common agreement about their reliability. In fact, often a visual inspection shows that images with higher scores do not appear better than images with lower scores. Thus, it is hard to establish which are the best models.

It is fundamental to find out which metrics are the most meaningful to proceed in the research of better SR models. To do this, it may be interesting to plot *all* the results of our experiments and visually analyze the differences and similarities among them with respect to the scores they obtain according to the considered metrics. Moreover, it may be useful to check what images with analogous results across different experiments have and do not have in common, for example to build a more significant benchmark dataset.

2 Related Works

In the supplementary material [2] of [3] by Galasso et al. (2013), the authors rely on *scatter plots* to detect strange patterns in the results of neural networks for video segmentation: in each plot, a point is an image described by a pair of metrics (e.g. boundary precision-recall VS volume precision-recall). In this way, all the pairs of metrics are analyzed to discover when they agree and when they disagree, to find out if they are considering different but useful points of view or just making some errors. In particular, if a subset of images is aligned along one direction, but distant along the other, we can inspect those images to analyze the different behaviors of the two metrics and find a reason for their inconsistency, or discard one of the metrics.

Our intent is to extend this approach with an interactive application which allows to select images from the scatter plots to easily compare them and exploit also other visualizations, such as a parallel coordinates plot and a box plot, to combine different insights.

The code of this report and of the application is available on GitHub¹, while the web app is hosted on Heroku². Please note that, since it is hosted on a free server, it may be slow, especially when visualizing all the data.

3 Dataset and Libraries

For this project, we used a dataset composed of frames extracted from videos downloaded from Fish4Knowledge [4], Pexels³, Nautilus Live⁴, and the website of the Ocean Exploration Video Portal of the National Centers for Environmental Information of the USA [5]⁵.

¹<https://github.com/GioFic95/Visual-Analytics-for-Super-Resolution>

²<https://sheltered-taiga-09066.herokuapp.com/>

³<https://www.pexels.com/it-it/>

⁴<https://nautiluslive.org/>

⁵<https://www.ncei.noaa.gov/access/ocean-exploration/video/>

The resulting dataset was splitted in three parts, for training, validation and testing. Here will use the test set, composed of 960 images, together with their super-resolved counterparts, obtained through about fifty experiments, where we tried different compression methods and training datasets, and many model parameters.

The dataset used for this project also includes, for each super-resolved image, its evaluation with SSIM, PSNR_Y, PSNR_RGB and LPIPS [6], already computed and stored in CSV files. Thus, the total amount of data is

$$960 \text{ images} \times 51 \text{ experiments} \times 4 \text{ metrics} = 195'840 \text{ values.} \quad (1)$$

To better visualize such amount of data, our application allows to see only information relative to a subset of the dataset, based on which compression method and training dataset were used.

This preprocessing was performed using Python libraries such as PyTorch, Pandas and Numpy, while the visualizations have been realized using Plotly⁶ and its Dash⁷.

4 Visualizations and Analytics

Since our interest is in evaluating both metrics *and* SR models, other visualizations with respect to the scatter plots proposed in [3] can be useful too:

- **Parallel coordinates** plot with the average scores of each model, to show the performances of the networks according to the various metrics and compare them;
- Another **scatter plot** with two coordinates computed via **PCA**, to explore the possibility of finding new representative metrics, from the combination of the old ones;
- Pairs of original and super-resoluted **images**, shown when a point on a scatter plot is selected, to link the behavior of a model on that image with the score it obtained.
- A **box plot** with statistics of a selected subset of images, to analyze differences among the images across all the models.

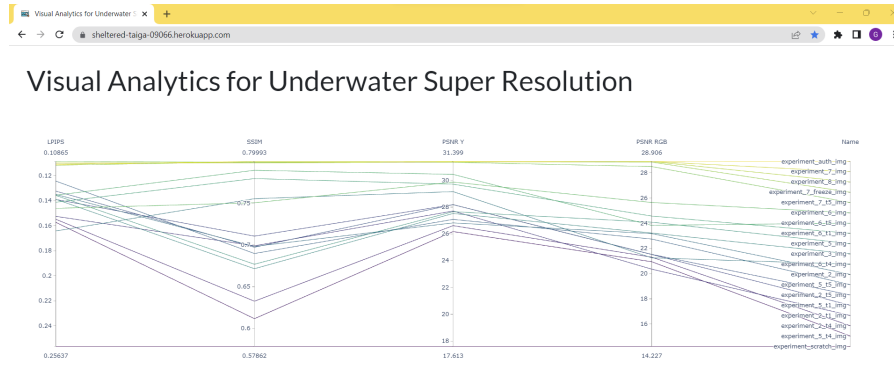


Figure 1: The parallel coordinates plot

4.1 The Parallel Coordinates Plot

The first part of the dashboard, in figure 1, presents a parallel coordinates plot, which shows a set of experiments with their aggregated results, according to all the metrics. The models are dynamically ordered based on a composition of the metrics which takes into account which values are considered better for each metric (the higher the better or vice versa), i.e.:

$$\text{SSIM} + \text{PSNR_Y} + \text{PSNR_RGB} - \text{LPIPS.} \quad (2)$$

⁶<https://plotly.com>, <https://plotly.com/python/>

⁷<https://plotly.com/dash/>, <https://dash.plotly.com/>

In this way, the crossing among lines in the plot is minimized, since we have the best experiments at the top of it, and the worst at its bottom. Moreover, for the same reason, LPIPS, which resulted to be the most discordant metric, is by default at the first column of the plot.

Brushing in the parallel coordinates plot allows to restrict the number of considered models also in the scatter plot.

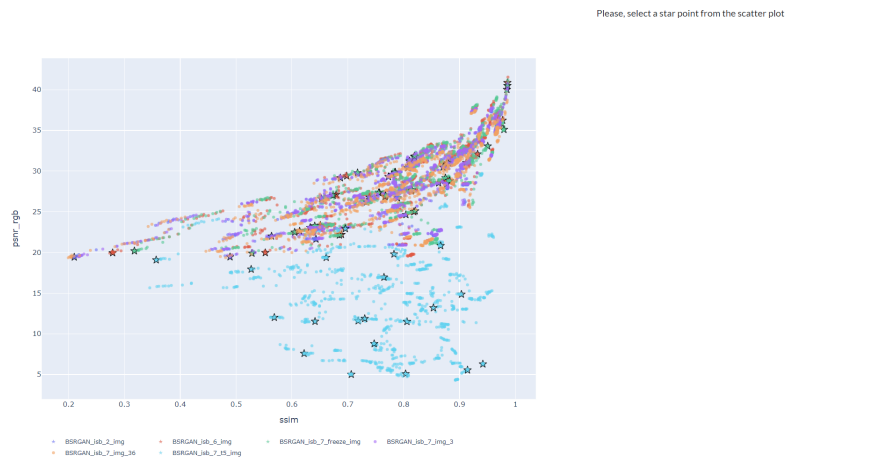


Figure 2: The scatter plot

4.2 The Scatter Plots

A set of scatter plots allows to compare pairs of metrics, to see if and when they agree and disagree. That is, each point in the plot is an image generated by a super-resolution model, with the coordinates given by two metrics selected by the user, and the color depending on the algorithm used to produce the image.

A subset of the points is depicted as a **star**: in this way, we highlight some items to make it easier to select and compare them. In particular, most of these points are chosen such that they have similar value on one axis and different value on the other, that is, they are aligned along one of the axes, so that we can investigate on the images with maximum similarity according to one metric and dissimilarity according to the other.

In the example in figure 3 we selected two images with very low PSNR_RGB (~ 5 db) and quite high SSIM ($\sim 80\%$ on the left, $\sim 91\%$ on the right). We can clearly see that the second one, which should have higher quality, is not better than the first one.

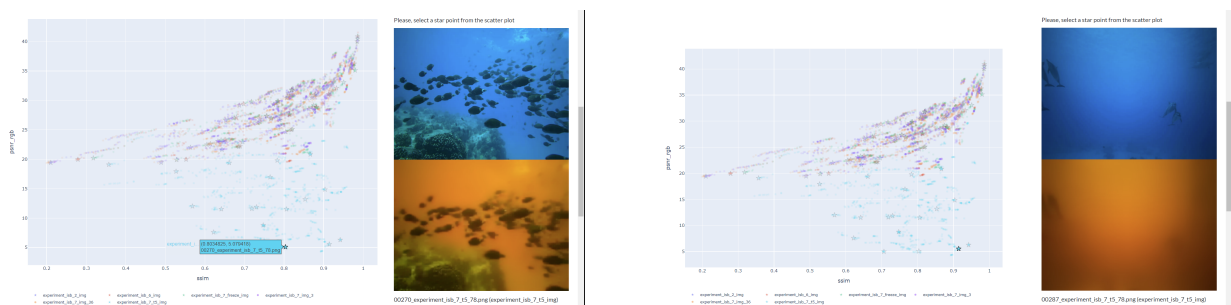


Figure 3: The scatter plot with a selected image

The user can also choose to show the plot of the coordinates obtained with **PCA**. These values are computed when the application is launched, for the whole dataset, since the division into subsets is only intended for ease of visual consultation.

The results seem to show that PCA_X tends to put the best images on the left (negative values), while PCA_Y puts them in the center (close to zero). These new metrics look more coherent with each other, in fact there is a small head on the left center of the plot and a large sparse tail on the right.



Figure 4: Scatter plot comparing PCA results

4.3 The Main Menu

Between the parallel coordinates plot and the scatter plot there is the menu bar, composed of four elements:

- **Training dataset:** allows to select the dataset used to train the super-resolution model, between the training split of the one described in section 3 and the data provided to our group by a client of WSense⁸, the company with which we are collaborating;
- **Compression type:** allows to select the compression type used to pre-process the images used as input for inference and testing;
- **Scatter metrics:** allows to select the pair of metrics to be compared in the scatter plot, including the coordinates produced by PCA;
- **Number of items:** shows the number of images in the dataset currently used to generate the parallel coordinates plot and the scatter plot.



Figure 5: The main menu of the web app

It is important to underline that the radio buttons affects the parallel coordinates plot, the scatter plot, and the box plot; the new selection reflects also on the counting of the items.

The position of this menu makes it comfortable to click on the buttons both when looking at the parallel coordinates plot and at the scatter plot.

4.4 The Box Plot

A box plot is included to allow the user to further explore the dataset with respect to the general behavior of some images through all the considered experiments. In particular, when selecting a new subset through the main menu, the five elements with highest and lowest *variance*, *standard deviation*, *mean*, *median*, and *difference between minimum and maximum* are computed and selected to be shown in the box plot. In this way, the user can click on any of these point and show the relative image, to study what pictures with similar and peculiar behaviors have in common, without overcrowding the plot with too many elements.

⁸<https://wsense.it/>

A drop-down menu allows to select the metric based on which we want to further explore the dataset.

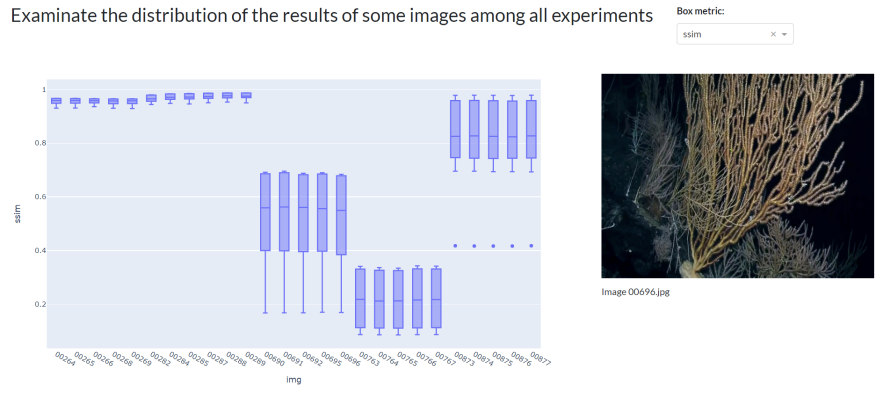


Figure 6: The box plot

To enforce the idea that with the box plot we want to analyze the behavior of some images across all models, this diagram is kept separated from the others, thus also highlighting that filtering via brushing does not affect it.

5 Conclusion

The implemented dashboard allows for an interactive visual inspection of the results of a set of experiments for underwater image super-resolution. The advantages of such analysis are various:

- Compare experiments at a glance;
- Easily compare images with respect to the scores they obtain;
- Find out which are the most reliable metrics;
- Define new metrics using PCA;
- Deeply explore a subset of significant images.

It may be interesting, in the future, to add the possibility to load any dataset, by just uploading pairs of original and super-resolved images, so that the same tool can be used by anyone with any image. Another feature worth including into the app would be the opportunity to add new metrics, from a list or even custom.

References

- [1] Zhihao Wang, Jian Chen, and Steven C. H. Hoi. *Deep Learning for Image Super-resolution: A Survey*. 2019. DOI: 10.48550/ARXIV.1902.06068. URL: <https://arxiv.org/abs/1902.06068>.
- [2] Fabio Galasso et al. “A Unified Video Segmentation Benchmark: Annotation, Metrics and Analysis Supplementary Material”. In: ().
- [3] Fabio Galasso et al. “A unified video segmentation benchmark: Annotation, metrics and analysis”. In: *Proceedings of the IEEE International Conference on Computer Vision*. 2013, pp. 3527–3534.
- [4] Robert B. Fisher et al. *Fish4Knowledge: Collecting and Analyzing Massive Coral Reef Fish Video Data*. 1st. Springer Publishing Company, Incorporated, 2016. ISBN: 331930206X.
- [5] Diva J. Amon et al. “Deep-Sea Debris in the Central and Western Pacific Ocean”. In: *Frontiers in Marine Science* 7 (2020). ISSN: 2296-7745. DOI: 10.3389/fmars.2020.00369. URL: <https://www.frontiersin.org/article/10.3389/fmars.2020.00369>.
- [6] Richard Zhang et al. *The Unreasonable Effectiveness of Deep Features as a Perceptual Metric*. 2018. DOI: 10.48550/ARXIV.1801.03924. URL: <https://arxiv.org/abs/1801.03924>.