Hyperspectral Image Fusion

A Comprehensive Review

Miguel Jorge da Lomba Magalhães

Master's thesis
Master's Programme in Imaging and Light in Extended Reality (IMLEX)
School of Computing
University of Eastern Finland
August 2022

UNIVERSITY OF EASTERN FINLAND
Faculty of Science and Forestry, Joensuu
School of Computing
Master's Programme in Imaging and Light in Extended Reality (IMLEX)

da Lomba Magalhães, Miguel Jorge: Hyperspectral Image Fusion – A Comprehensive Review
Master's thesis, 52 p.
Supervisors: Prof. dr. Kevin Smet and Prof. dr. Yasushi Kanazawa
August 2022

Abstract: Existing hyperspectral imaging (HSI) systems produce images that lack spatial resolution due to hardware limitations. Even with the proven efficacy of this technology in several computer vision tasks, the aforementioned limitation obstructs its applications. Contrarily, conventional RGB images have a much larger resolution with just three spectra.

In this thesis, we present the state-of-the-art of hyperspectral image fusion (HIF) which merges a low-spatial resolution HS image with a high-spatial low-spectral resolution image of the same scene, such as an RGB image. This work explores super-resolution from a practical point of view, with all the methods sharing the experimental conditions and without fine parameter tuning. Moreover, we describe the limitations and issues of existing methods, the shortcomings of the existing comparison techniques, and how these should be addressed in a practical comparison framework.

This work took place in the Light & Lighting Laboratory, a research group embedded in the Department of Electrical Engineering of KU Leuven, located on the Technology Campus Ghent in Belgium.

Keywords: hyperspectral; image fusion; super-resolution; data fusion; review; comparison; upsampling; multispectral; RGB;

Contents

List of Figures

List of Tables

1. Introduction

Hyperspectral imaging (HSI) is a spectroscopy-based analytical technique that collects several images (bands) over a wide and continuous wavelength range with a large number of discrete wavelength bands of the same spatial area [95], forming an hyperspectral (HS) cube - presented in fig. 1.1. This cube contains the spectrum of light for each pixel of the scene with a fine wavelength resolution. It is formed by two dimensions that represent the spatial position (x, y), and a third that is the spectral coordinate (λ). For reference, a typical colored image only contains three distinct values for each of the three primary colors Red Green Blue - see table 1.1.

A spectral cube can be viewed as several slices at consecutive wavelengths that display the spatial data for each wavelength (a *slice* of the cube, on the right of fig. 1.1) [82]; or can be viewed as several vectors arranged together along the spatial dimensions where each vector corresponds to the radiance for each specific location for all spectral bands [13, 82], on the left side of the same figure.

Taking into account such a detailed representation of scenes, which can have hundreds of spectral bands of information, allows HSI to be employed in several fields. Its applications range from remote sensing [4, 14] through medical imaging [79] to

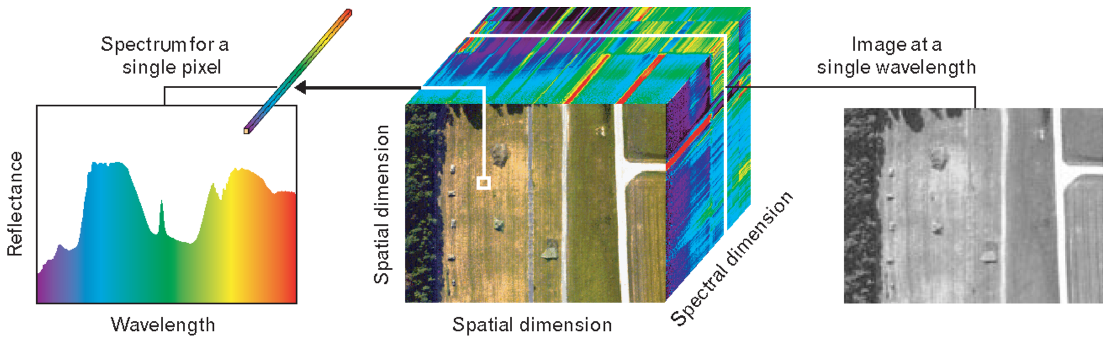


Figure 1.1: Representation of a hyperspectral (HS) cube (center) with a simultaneous illustration of a sample of spatial and spectral data. The spectrum of a pixel (left) has the spectral radiance values of that pixel along the spectral dimension. The slice of a single wavelength (right) represents the different intensity of the spectral radiance values across both spatial dimensions [82].

Image Type	Number of bands	Example application
Pan-chromatic	1	Satellite imagery
RGB	3	Social media
Multispectral (MS)	order of tens	Food research
Hyperspectral (HS)	several hundred	Remote sensing

Table 1.1: Comparison image types per number of bands together with an example of a practical application [25, 95].

food quality and safety control [43]. Moreover, it has also been used to improve the performance of computer vision tasks, in particular segmentation and classification [107], face recognition [111], tracking [113], and document analysis [63].

Although HS cameras can be built to function in many regions of the electromagnetic spectrum, the work presented in this thesis is focused on the visible spectral bands ($\lambda \in [400, 700]$ nm) and near-infrared band ($\lambda \in [700, 1000]$ nm) [1, 13]. Moreover, since HS images contain hundreds of narrow consecutive bands, per definition, each band consists of contiguous intervals with a bandwidth of 2 nm to 20 nm - for accurate colorimetric measurements, a maximum of 5 nm is recommended. Therefore, a HS image has finer spectral resolution (higher number of bands) when compared to multispectral (MS) image. For instance, multispectral imagery usually refers to separated, not contiguous bands in the order of tens with broader bandwidths [25] - see table 1.1. There is some ambiguity in these two terms, however, as a distinguishable feature, Cucci and Casini [25] mentions that "only HSI offers the possibility of reconstructing highly resolved spectra for each pixel of the imaged area, thus enabling spectroscopic analysis of the constituent materials".

1.1 Motivation

Although HSI has proven its usefulness with its rich spectral information, it lacks acutely in terms of spatial resolution. This problem originates from hardware limitations - since the system needs to capture small spectral windows, it requires long exposures to collect enough photons to ensure a good signal-to-noise ratio, which leads to low spatial resolution in hyperspectral images. For reference, a typical colored image (RGB) requires a smaller exposure time resulting in a high spatial resolution image, however it lacks spectral information as it contains just three different bands (channels).

One could argue that we could use high-resolution sensors for hyperspectral imaging, however, this is not very effective since it decreases the density of the photons that reach the sensor due to the very narrow spectral window [4, 12].

Another possible argument is that we could do measurements of smaller regions and then merge them. However, to begin with, that would lead to a lengthier process than the current solution which is already a lengthy undertaking. Therefore, that is also not a humanely viable solution when collecting a large HS dataset.

Therefore, this lack of spatial resolution hinders the development of further applications and obstructs the accuracy of the already existing ones.

1.2 Scope and Objectives

Since the issue of low-resolution images arises from hardware limitations, there have been several developments in software-based approaches to improve the spatial resolution of hyperspectral images.

Therefore, in this thesis we research, test, and compare existing methods of spatial resolution enhancement in hyperspectral images in a practical environment, simulating real-world conditions.

Finally, the increased information in the form of the improved spatial resolution of the HS image leads to better results when analysing the data. As mentioned previously, the measurement of accurate super-resolution hyperspectral reflectance images improves the applicability of this technology across all the pre-existing applications and also allows for other novel usages that would otherwise not have been possible with the available low-resolution HS images.

1.3 Document Structure

This thesis is organized as follows. Firstly, chapter 2 introduces the problem together with the existing types of techniques for spatial resolution enhancement in hyperspectral images, the most commonly used datasets, and metrics. This section presents the context of our research and the current state-of-the-art, and it also serves as an introduction to the next section.

Afterward, in chapter 3, the concrete benchmarking conditions are introduced together with its particularities, which issues it addresses, and how it differs from existing papers.

This is followed by a discussion of our findings and a comparison of results in chapter 4. It presents the selected metrics used for comparison, and their values regarding the output of the selected techniques. Additionally, a visual analysis of their outputs is also included to further inspect the obtained results.

Finally, we conclude this thesis in chapter 5, where a summary of the entire thesis is presented together with its conclusions and a review of viable future work. For a detailed Gantt chart of the work plan of this thesis, please refer to page 53.

2. Related Work

As previously mentioned in chapter 1, due to hardware limitations, the latest research on super-resolution hyperspectral imaging has been on software-based solutions. This chapter introduces several of those techniques of spatial resolution enhancement in hyperspectral images.

These software-based approaches can be divided into two main categories: (1) single hyperspectral image super-resolution (SHSR), and (2) hyperspectral image fusion (HIF). SHSR has received less attention from researchers due to the fact that it is a more sophisticated task since it only has as input an HS image. Hyperspectral image fusion (HIF) on the other side is a simpler task that is able to produce superior results with the disadvantage that it also requires an additional higher spatial resolution image which can be either MS, RGB, or pan-chromatic[1] - see table 1.1. However, since RGB cameras are so common and affordable when compared to HS cameras, it is acceptable to assume that both images can be easily acquired at the same time.

Since SHSR has considerably less spatial information available than HIF methods, its results are mathematically limited in terms of the final information that they can produce. For demonstration purpose, we use an HS image of an Extended eSFR ISO 12233:2017 chart - pictured in fig. 2.1. These charts have resolution wedges which allow the measurement of camera spatial resolutions - this is measured at the region where the different lines stop being distinguishable and are blurred as if they were a single region.

The HIF methods take as input both a high-spatial low-spectral resolution image (pictured in fig. 2.2) and a low-spatial high-spectral resolution image (resolution pictured in fig. 2.3), while the SHSR methods only take the latter. From these images, it is intuitive that no matter how good the SHSR method is, it is impossible to distinguish between a single homogeneous block and lines - for example, in the fig. 2.3 this region starts on the number "2" mark. On the other side, the HIF with the information from the high spatial-resolution image will be able to accurately reconstruct

[1]It is a single-band gray-scale image with a high spatial resolution.

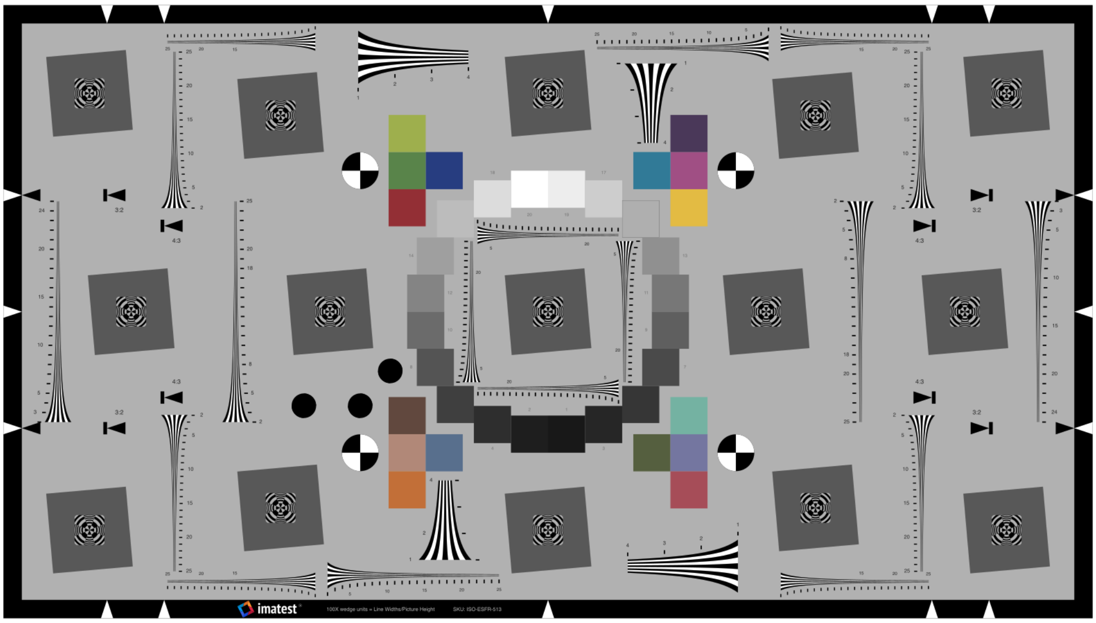


Figure 2.1: Representation of an Extended eSFR ISO 12233:2017 chart.

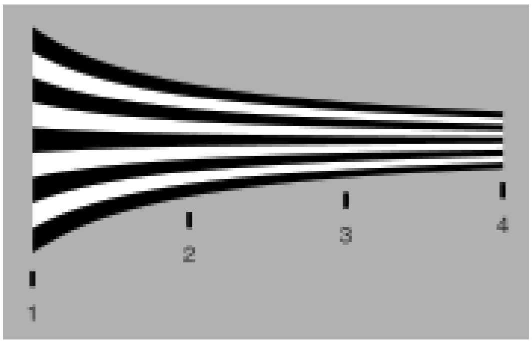


Figure 2.2: Higher resolution image of a wedge.

Figure 2.3: Lower resolution image of a wedge.

the spatial resolution of the image. For this reason, HIF methods can produce better results than SHSR methods. This work is focused only on the HIF super-resolution methods.

HIF can be summarized as the practical method of merging images from two or more sensors to create a composite image [124]. It has been defined in the remote sensing field as "a formal framework in which are expressed means and tools for the alliance of data originating from different sources. It aims at obtaining information of greater quality; the exact definition of *greater quality* will depend upon the application" [65].

These techniques have their genesis in the remote sensing field due to the early use of HSI in airborne systems. The first forms of HIF were under the term pan-sharpening, which can be considered a special case of the broader problem [55]. In

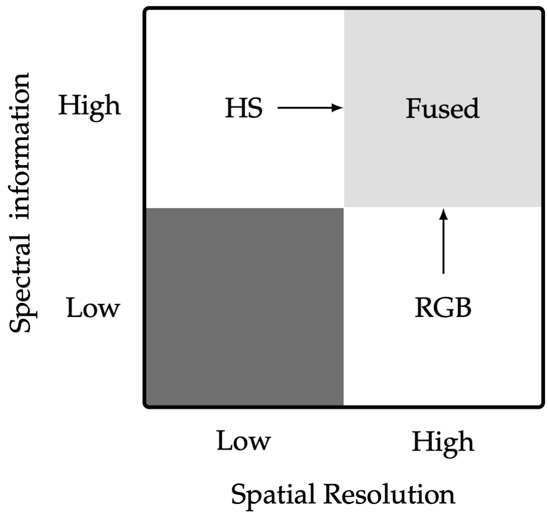


Figure 2.4: Spatial vs spectral representation of the data in HIF [36].

pan-sharpening, the inputs are an HS and a pan-chromatic image, which contains information of a single wavelength [7].

The main idea is to extract detailed spatial information from the panchromatic image, and then transfer this detailed spatial information to the HS image to reach the desired output [105].

With the previous line of thought in mind, it becomes clear that an HS and RGB image fusion is also just another case of the broader category. Since RGB images are ubiquitous and more easily accessible than MS images, a common testing ground for HIF is with these types of images as input, which we will follow suit.

In short, the combination of the information from the two images in HIF provides more comprehensive information in the form of a high-spatial and high-spectral resolution image [124] - see fig. 2.4.

2.1 Problem Formulation

The goal of HIF is to obtain an accurate super-resolution hyperspectral image from two input images: a low-spatial high-spectral resolution image and a high-spatial low-spectral resolution image, as previously described. For simplicity, the input images will be addressed as hyperspectral (HS) image and as Red Green Blue (RGB) image, respectively. The output super-resolution image will be addressed as super-resolution (SR) image. The input-output diagram of HIF is presented in fig. 2.5.

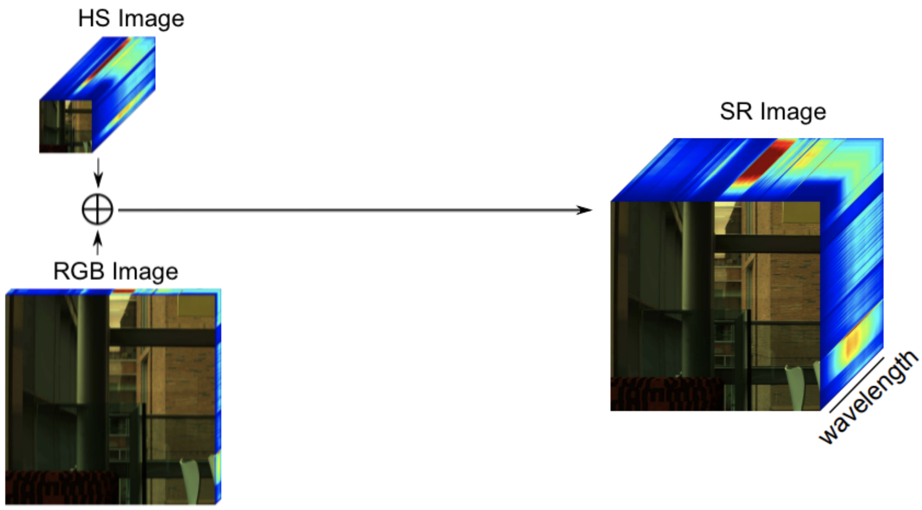


Figure 2.5: Input and output diagram of HIF.

This section starts by introducing a mathematical formalization of this problem. This has the purpose of clarifying to the reader how the inputs and the corresponding output are represented. Afterward, we discuss the existing challenges within the presented problem.

2.1.1 Formalization

As mentioned before in chapter 1, a HS cube (fig. 1.1) contains the spectrum of light for each pixel and is formed by two dimensions that represent the spatial position (x, y), and a third that is the spectral coordinate (λ). Therefore, a cube $\mathbf{C}$ can be mathematically described as $\mathbf{C} \in \mathbb{R}^{x \times y \times \lambda}$.

Based on this convention, the inputs and corresponding output of the system can be formally summarized as follows. Let $\mathbf{HS} \in \mathbb{R}^{w \times h \times \Lambda}$ and $\mathbf{RGB} \in \mathbb{R}^{W \times H \times \lambda}$ be the two input images. The variables w, h and λ denote the width, the height and the spectral dimension respectively; with the same capital letters corresponding to the same variable but with high value, such that $W \gg w$, $H \gg h$ and $\Lambda \gg \lambda$. Additionally, $\lambda = 3$ since the RGB image has three color channels (Red Green Blue) - table 1.1. From these inputs, we obtain the super-resolution hyperspectral image $\mathbf{SR} \in \mathbb{R}^{W \times H \times \Lambda}$ through $\mathbf{\Psi} : \mathbb{R}^{w \times h \times \Lambda} \times \mathbb{R}^{W \times H \times 3} \to \mathbb{R}^{W \times H \times \Lambda}$, with $\mathbf{SR} = \Psi(\mathbf{HS}, \mathbf{RGB})$.

2.1.2 Challenges

The scope of this work was already introduced in chapter 1 together with its objectives. However, several challenges arise from the task at hand.

Since we capture two separate images (the HS and the RGB) we need to deal with two separate cameras. These cameras are pointing to the exact same location but positioned next to each other with a slight position shift. Moreover, each camera will have its own lens which will further impact the difference between images. Therefore, the input images of the system will be unaligned and will contain distinct distortion patterns. To overcome this issue, one must first calibrate both cameras to remove distortions, and then co-register both images correctly making sure they are aligned and overlap accordingly. Consequently, high accuracy of geometric co-registration of the data is of the uttermost importance, since even a minor error will result in an inaccurate spatial up-sampling process [85].

Additionally, at least one of the input images will have a padding around the area of interest. This occurs because images need to be aligned, therefore a padding must be present so that we don't lose valuable information when applying the geometric transformation that registers the two input images. So, before starting the super-resolution processing, the images must be cropped to only include the part of the images with the area of interest.

2.2 Wald's Protocol

In 1997, Wald et al. [116] proposed what would be named Wald's Protocol, a paradigm for quality assessment of fused images.

In order to be able to evaluate an image fusion process, one needs to start with a high-spatial resolution hyperspectral reference image, also known as the ground truth (GT). The hyperspectral ground truth reference image is the starting point for the protocol, which is composed of the following steps [78, 147]:

1. From the HS reference image, we produce two synthetic images that are going to be the input to the HIF method: (1) a low spatial resolution HS image, and (2) a high spatial resolution RGB image. To synthesize the low-spatial resolution hyperspectral image, the high-spatial resolution hyperspectral GT image is blurred and downsampled by a pre-defined scaling factor to a smaller spatial resolution; and to synthesize the RGB image we typically simulate a spectral response of an RGB camera over the GT image.

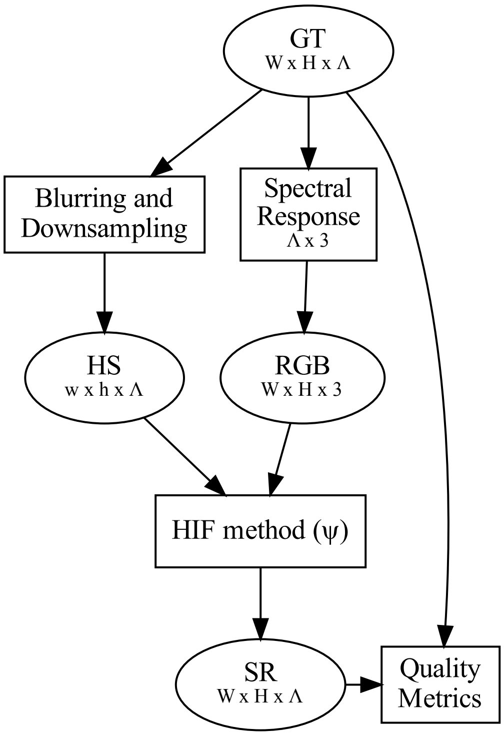


Figure 2.6: Flow diagram of the evaluation methodology for hyperspectral image fusion techniques derived from the Wald's protocol.

2. Those two images serve as input to the HIF method that we are testing, which in turn produces a super-resolution (SR) HS image.
3. The output SR HS image is then compared against the hyperspectral ground truth (GT) reference image. This is used to compute quality metrics and perform a visual analysis of the results.

For clarity, the flow diagram of this experimental evaluation methodology is presented in fig. 2.6. The nodes shaped as ellipses represent images and follow the notation presented in the section 2.1.1, while the rectangular shapes represent computations.

Additionally, the **scaling factor** that is used to downsample the HS GT image to generate the downsampled HS image, is the same scaling factor that is used to upsample the method's input HS image to generate the SR HS image. The scaling factor (sf) can be calculated as follows:

$$\mathrm{sf} = W/w = H/h \tag{2.1}$$

From this protocol it is clear that there are three essential aspects of a simulation environment: (1) **HIF methods**, (2) **datasets**, and (3) **quality metrics** - the next sections present those.

2.3 Types of HIF Methods

This section presents different types of techniques for HIF. Each type includes references to sample methods within that category (not an exhaustive list) and mentions its advantages and limitations. Nevertheless, these are not strict categories with some methods being hybrid in their genesis - having more than a single type of technique involved [78].

2.3.1 Spatial Unmixing

Techniques of this type exploit spatial unmixing (SaU) for the resolution enhancement of HS images [53, 84, 85, 112, 165, 166]. These techniques assume that the spectrum for a single pixel is composed of a linear combination of a finite number of pure spectra of known materials (end-members) [4]. These methods use the reference spectra of pure spectral classes to derive their proportions in mixed pixels spectra [112, 165]. Therefore, an *a priori* knowledge of end-members and their spectral profiles is required [85]. These techniques can make use of spectral libraries of pure spectra [4].

Within SaU, constraints can be included to dictate that a mixed pixel needs to be calculated from all end-members according to a percentage between 0% and 100%, and that these must sum to 100% - this is labeled as constrained unmixing. These constrained unmixing techniques tend to better preserve the available information of the HS image, but the unconstrained unmixing techniques are preferable when dealing with noisy data [165].

The unmixing-based algorithms perform well when dealing with images with a large spatial homogeneity (with simple shapes and few colors), but it may lead to a loss of local spectra variability for the same region due to an implementation across the whole image at once [54]. Zurita-Milla et al. [166] addresses this drawback with the use of a sliding window that makes use of the neighboring pixels to take into account the local variability. However, this limits the spectral signatures (end-members) to just the ones present in that sliding window.

This type of technique might work well when dealing with homogeneous regions, but it leads to spectral distortions when dealing with fine details. Furthermore, these methods require high accuracy of geometric co-registration of the data [85].

2.3.2 Linear Transformation of Color Coordinates

This type of techniques uses the linear transformation of color coordinates L*a*b*, computed from the input low-resolution HS image, and L*, computed from the RGB image [17, 57, 64]. This approach exploits the fact that human vision is more sensitive to lightness and fuses this component from the high-spatial resolution image with the HS image.

This is possible since the light in the eye becomes progressively smaller as the spatial frequency increases as a consequence of the optical limitations of the retinal mosaic [60]. Therefore, the chromatic channels have a much lower spatial resolution than the luminance channel. This has been exploited by image compression algorithms such as the JPEG format [117]. On the downside, it leads to a distortion of the spectral information [18].

In short, this approach is suitable to produce colorimetric images that are *"good enough"* for the human eye [57], but it is not suitable for scientific analysis due to its lack of spectral accuracy [18].

2.3.3 Pan-Sharpening Based

HIF has its genesis in the remote sensing field, where pan-chromatic images (single band images - see table 1.1) were ubiquitous. Therefore, those forms of HIF only addressed the pan-sharpening problem which fuses an HS image with a pan-chromatic image. Consequently, pan-sharpening can be considered a special case of the broader HIF problem [55].

Based on the pan-sharpening methods, several techniques were introduced to generalize those methods to the broader HIF problem, allowing the use of a multiple bands (MS) image as the high-spatial resolution input image instead of panchromatic image [41].

For example, Chen et al. [23] pair each band of the MS image with the corresponding group of the spectral images (bands) from the HS image. Grohnfeldt et al. [44]

calculate a weighted pan-chromatic image from the MS image[2] for each band of the HS image - this pan-chromatic image is then used to guide the spatial resolution enhancement of each HS band.

Selva et al. [97, 98] introduced the term hyper-sharpening to denominate this paradigm which allows traditional pan-sharpening methods to be effective in the fusion of HS and MS images.

However, because the pan-sharpening techniques were adapted to the general HIF and not designed with that in consideration, its results are lacking when compared with the state-of-the-art HIF methods that were specifically designed for the general problem.

2.3.4 Matrix Factorization

In the last decade, matrix factorization for the enhancement of the spatial resolution of HSI systems was introduced [54, 62, 146, 135]. Although a sub-type spatial unmixing, matrix factorization is presented separately in a different category due to its distinct characteristics.

These methods can be divided into two stages [62]: the first is the application of the unmixing algorithm to the HS input to estimate a spectral dictionary containing pure end-members; and in the second, the learned spectral dictionary in conjunction with the RGB image are used to produce the desired output, as described in section 2.3.1.

This technique was explored by Yokoya et al. [146] who used a coupled nonnegative matrix factorization (CNMF) algorithm for the unmixing stage, the end-members and abundance matrices (values of the linear combination that generates each mixed pixel) via alternating NMF under the constraints of the observation model, composed by both the SRF[3] and PSF[4]. Wycoff et al. [135] made use of an algorithm based on alternating direction method of multipliers (ADMM) for the factorization of the matrices. Akhtar et al. [4] proposed a sparse spatiospectral representation of HS images that also incorporates the nonnegativity of spectral signals and exploits the spatial structure of the images.

This type of method has the advantage of not requiring a high accuracy of geometric registration between the two input images [54].

---

[2]It assumes that the pan-chromatic image is the result of a linear combination of the bands from the MS image.

[3]Spectral response function (SRF) describes the relative sensitivity of an imaging system to energy of different wavelengths.

[4]Point spread function (PSF) describes the response of an imaging system to a point object.

2.3.5 Bayesian-Based

This type of technique performs Bayesian sparse learning to conduct probabilistic reconstruction of the SR HS image based on the two input images. The base idea of this approach is to maximize the likelihood of the spectra in the output SR HS image.

Bayesian-based methods use a non-parametric[5] Bayesian sparse image representation to perform the HIF. These approaches extrapolate the probability distributions for the material spectra and their respective proportions in the image. Using that information it uses it to sparse code the RGB image, which allows it to learn a dictionary (of spectra) to then construct the SR image [6, 49, 59, 104, 128, 129, 130, 158].

Although they are able to perform well, they require an high accuracy of geometric co-registration of the input images.

2.3.6 Tensor-Based

Tensor factorization has been applied to multi-frame data[6] in several areas, ranging from denoising, completion, classification, among others [27]. Following these works, tensor factorization was also introduced to the HIF field.

A tensor (a high-order extension of a matrix) can capture the correlation between the spatial and spectral domains at the same time [152]. Although similar to matrix factorization, when using a tensor it is possible to directly decompose a SR image as a tensor and three dictionaries (one per dimension): one for the spectral domain and two for the spatial domains (height and width) [70].

This type of method has been extensively used by several authors [27, 29, 50, 70, 152, 138, 139]. It has the ability of producing finer spatial details whilst preserving spectral structures [152].

[5]This allows very flexible models that are capable of encompassing natural information easily. Its parameter space has infinite dimensions, contrarily to parametric scenario where it has a finite dimension [89].

[6]Compact representation of a data series where multiple data instances are merged together in a single file.

2.3.7 Deep Learning

Deep learning is a category of machine learning methods which imitates the way humans gain certain types of knowledge through multi-level artificial neural networks. This type of algorithm has been extensively used for several tasks which require image processing such as image restoration and resolution enhancement of RGB images [31]. Inspired by these developments, deep learning has recently also been applied to the spatial enhancement of HS images with several distinct techniques which include non-local low-rank tensor approximation [123], transfer learning [149], 3D full convolutional neural network (CNN) [83], among others [48, 90, 93, 103, 136].

Due to disk size constraints and taking into account that each HS cube can have several gigabytes of data, most HS datasets consist of just a few images. These datasets lack enough HS images for a good training of the models. Moreover, their size per item means that it is not practical, or even viable, to train the CNNs with these datasets [103]. To overcome this issue, Hang et al. [48] exploits two intrinsic properties of HSI: spectral correlation and projection property - *i.e.*, the RGB image can be regarded as a three-dimensional projection of the HS image.

These methods tend to obtain good results and in a computationally efficient manner. Although they have have a drawback, which is the fact that they might need to be trained before being used which is a lengthy task, but a task that only needs to be performed once.

2.3.8 Registration Simulation

Recent methods [94, 120, 163] have developed registration simulation methods that take into account the real-world scenario where images are not exactly aligned, and can also handle scaling differences and spatial distortions. Other implementations take the alignment for granted and ignore the real applicability of their method.

The main advantage when compared to other types of methods is that these techniques estimate and optimize the SR image iteratively, therefore, they are resilient to the nonexistence of *multi-modality registration*[7].

[7] This refers to the lack of co-registration between the two input images of an HIF method, since these images are acquired using different cameras.

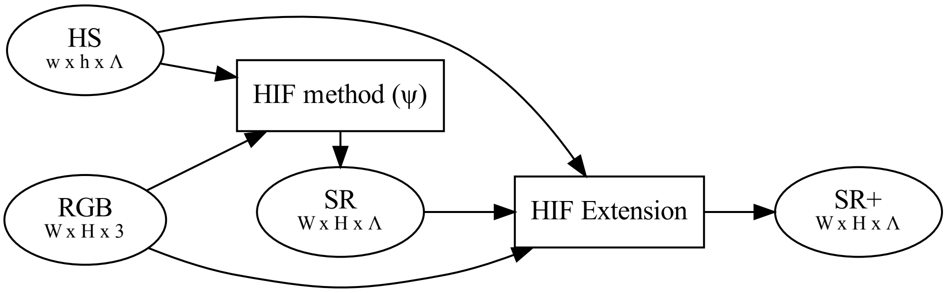


Figure 2.7: Diagram of the extended HIF pipeline. The node "SR+" denotes the improved super-resolution image, which is the output of the HIF extension.

2.3.9 Extensions

Although not a type of method *per-se*, there have been recent developments in developing HIF extensions - listed in section B.4. These extensions take the inputs and the output of an HIF method and improve the output HS SR image [114, 122].

These extensions can be added to any of the aforementioned type of techniques, however their improvement varies according to the method being extended and its characteristics. Refer to fig. 2.7 for a diagram of the extended HIF pipeline.

2.4 Datasets

In order to test and compare the effectiveness of the different existing methods, experiments are usually conducted on widely available public HSI datasets.

CAVE dataset It consists of 32 HS images which have a spatial dimension of 512 pixels x 512 pixels with 31 spectral bands taken within the wavelength range between 400 *nm* and 700 *nm* [142]. The dataset[8] includes images with a wide range of natural and artificial materials, objects, shapes, and colors - see fig. 2.8.

[8]https://www.cs.columbia.edu/CAVE/databases/multispectral/

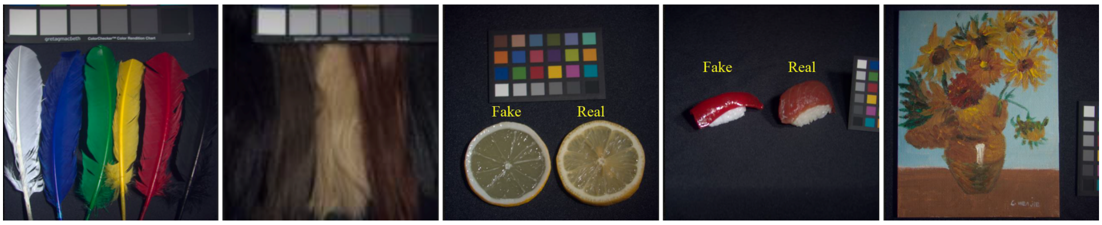


Figure 2.8: CAVE dataset sample images [142]. From the left to the right: feathers, hairs, lemon slices, sushi, and oil painting.

Harvard dataset It consists of 77 HS images which have a spatial dimension of 1040 pixels x 1392 pixels with 31 spectral bands taken within the wavelength range between 420 *nm* and 720 *nm* [19]. The dataset[9] includes images of both indoor and outdoor scenes - see fig. 2.9.

Figure 2.9: Harvard dataset sample images [19]. From the left to the right: imge7, imgb0, imgf7, imgh3, and imgd7.

NUS dataset It consists of 88 HS images which have a spatial dimension of 512 pixels x 512 pixels with 31 spectral bands taken within the wavelength range between 400 *nm* and 700 *nm* [87]. The dataset[10] includes images of both indoor and outdoor scenes, fruits, and color charts - see fig. 2.10.

Figure 2.10: NUS dataset sample images [87]. From the left to the right: Scene83, Orange Veg (CC), Scene38, Scene08, and Scene01.

ICVL dataset It consists of 200 HS images which have a spatial dimension of 1392 pixels $\times$ 1300 pixels with 519 spectral bands taken within the wavelength range between 400 *nm* and 1000 *nm* [8, 9]. The dataset[11] includes images of both indoor and outdoor scenes, and also some objects together with a color chart - see fig. 2.11.

[9] http://vision.seas.harvard.edu/hyperspec/index.html
[10] https://sites.google.com/site/hyperspectralcolorimaging/dataset/general-scenes
[11] http://icvl.cs.bgu.ac.il/hyperspectral/

Figure 2.11: ICVL dataset sample images [9].

EHU (remote sensing) dataset Although not an official designation, the name comes from the Euskal Herriko Unibertsitatea which compiled this dataset. It aggregates 7 remote sensing hyperspectral scenes captured between 1992 and 2008 each with distinct spatial resolution and a number of spectral bands between 103 and 224. The following scenes[12] are included: Indian Pines [10], Salinas, Pavia Centre, Pavia University, Cuprite, Kennedy Space Center (KSC) and Botswana - see fig. 2.12. Other commonly used remote sensing HS scenes (not included in the previous dataset) are named: Chikusei [145], WHU-Hi [161], Washington DC Mall, China farmland, USA farmland, Jasper Ridge, Urban, Samson and Cooke City.[13]

Figure 2.12: EHU dataset sample images. From the left to the right: Kennedy Space Center, Cuprite, Salinas, Pavia Centre, Pavia University, and Cuprite.

2.5 Quality Metrics

To fully evaluate the various methods it is necessary to have **full-reference quality assessment metrics** which ensure an objective comparison of the resolution enhancement process [58, 78, 102, 147]. To this end, we present in this section metrics which compare the output of the super-resolution (SR) methods ($\hat{x}$) with the ground truth (GT) of the datasets (x) - see section 2.2. The presented metrics can assess quality in the **spectral domain** (SAM and SID) and the **spatial domain** (SCC), or assess the **global image quality** (*Total Error*, RMSE, RASE, ERGAS, PSNR, SSIM, MS-SSIM, PSNR-B, UQI, VIF and Q2^n) - see table 2.1. Some of the formulas were omitted for brevity.

[12] https://www.ehu.eus/ccwintco/index.php/Hyperspectral_Remote_Sensing_Scenes
[13] https://rslab.ut.ac.ir/data

Metric	**Domain**	**Best Value**
Total Error	Global	$\downarrow 0$
RMSE	Global	$\downarrow 0$
RASE	Global	$\downarrow 0$
ERGAS	Global	$\downarrow 0$
SCC	Spatial	$\uparrow 1$
PSNR	Global	$\uparrow$
SSIM	Global	$\uparrow 1$
MS-SSIM	Global	$\uparrow 1$
PSNR-B	Global	$\uparrow$
UQI	Global	$\uparrow 1$
SAM	Spectral	$\downarrow 0$
SID	Spectral	$\downarrow$
VIF	Global	$\uparrow$
$Q2^n$	Global	$\uparrow$

Table 2.1: Comparison of quality metrics.The domain column represents the axis where the metric assesses quality, which can be either spectral, spatial or global domains. The best value column represents which value is the best either the smaller ($\downarrow$) or the larger ($\uparrow$), together with the best value when it exists.

Total Error This metric, proposed by Munechika [86], sums the RMSEB of each of the bands in an HS cube:

$$\mathrm{RMSEB}(x_j, \hat{x}_j) = \sqrt{\frac{\sum_{i=1}^{N} \left(x_{j_i} - \hat{x}_{j_i}\right)^2}{N}} \tag{2.2}$$

where x_{j_i} and $\hat{x}_{j_i}$ are the values of pixel i in band j of images x and $\hat{x}$; and N is the number of pixels in a band, equal to the number of rows times the number of columns.

The smaller the value of the *Total Error*, the better the fusion results, with the best value being 0. It is defined as follows:

$$\mathrm{TotalError}(x, \hat{x}) = \sum_{j=1}^{B} \mathrm{RMSEB}(x_j, \hat{x}_j) \tag{2.3}$$

where B is the total number of bands. For clarity, the total number of values in a cube is defined as B times N, the spectral times the spatial dimension. However, this metric is influenced by the number of spectral bands in an HS image. Therefore, it is not an adequate quality metric to be used to evaluate the results of HIF methods [115].

Root Mean Squared Error (RMSE) This metric is the standard deviation of the prediction errors. The smaller its value, the better the fusion results, with the best value being 0. It is defined as follows:

$$\mathrm{RMSE}(x, \hat{x}) = \sqrt{\frac{\sum_{j=1}^{B} \sum_{i=1}^{N} \left(x_{j_i} - \hat{x}_{j_i}\right)^2}{B * N}} \tag{2.4}$$

where all variables were already defined. This metric is not influenced by the number of bands like the the *Total Error*. However, it still poses the following problem: it is not unit-independent - an `uint8` and a `float` have as maximum values 255 and 1, respectively, resulting in different RMSE values for the same data in different formats [115]. This issue can be addressed by normalizing the values to a common unit.

Relative Average Spectral Error (RASE) It is represented in percent and describes the average performance of a method in the considered spectral bands [42, 96]. The smaller its value, the better the fusion results, with the best value being 0. It is defined as follows:

$$\mathrm{RASE}(x,\hat{x}) = \frac{100}{\mu_x}\mathrm{RMSE}(x,\hat{x}) \tag{2.5}$$

where μ_x is the mean value calculated over all bands and pixels for the ground truth spectral image - see eq. (2.6). Ranchin and Wald [96] introduced this metric to address the shortcomings of RMSE. However, it is not independent of the ratio between the higher and the lower spatial resolutions of the input images [115].

$$\mu_x = \frac{\sum_{j=1}^{B}\sum_{i=1}^{N} x_{j_i}}{B * N} \tag{2.6}$$

Erreur Relative Globale Adimensionnelle de Synthèse (ERGAS) It provides a global statistical measure of the quality of the fused data [115]. The smaller its value, the better the fusion results, with the best value being 0. It is defined as follows:

$$\mathrm{ERGAS}(x,\hat{x}) = 100\frac{h}{l}\sqrt{\frac{1}{B}\sum_{j=1}^{B}\frac{\mathrm{RMSEB}(x_j,\hat{x}_j)^2}{\mu_{x_j}^2}} \tag{2.7}$$

where $\frac{h}{l}$ is the ratio between the higher and the lower spatial resolutions of the input images; x_j and $\hat{x}_j$ are the bands j of images x and $\hat{x}$, respectively; $\mathrm{RMSEB}(x_j,\hat{x}_j)$ computes the Root Mean Squared Error between the two bands x_j and $\hat{x}_j$ (see eq. (2.2)); μ_{x_j} is the mean value for band j of the ground truth spectral image (x) - see eq. (2.8); and the other variables were already defined. Contrarily to RASE, this metric is independent of the ratio of the spatial resolutions $(\frac{h}{l})$, besides also being unit-independent and not influenced by the number of bands, the other conditions that the RMSE and the *Total Error* didn't fulfill.

$$\mu_{x_j} = \frac{\sum_{i=1}^{N} x_{j_i}}{N} \tag{2.8}$$

Spatial Correlation Coefficient (SCC) This metric reflects the indirect correlation of the spatial contiguity between images [162]. The larger the value of SCC, the better the fusion results, with the best value being +1. It can be computed using the following formula:

$$\mathrm{SCC}(x,\hat{x}) = \frac{\sum_{j=1}^{B}\sum_{i=1}^{N}\left(x_{j_i}-\mu_x\right)\left(\hat{x}_{j_i}-\mu_{\hat{x}}\right)}{\sqrt{\sum_{j=1}^{B}\sum_{i=1}^{N}\left(x_{j_i}-\mu_x\right)^2}\sqrt{\sum_{j=1}^{B}\sum_{i=1}^{N}\left(\hat{x}_{j_i}-\mu_{\hat{x}}\right)^2}} \tag{2.9}$$

where μ_x and $\mu_{\hat{x}}$ are the mean values for images x and $\hat{x}$, respectively - see eq. (2.6); and the other variables were already defined.

Peak Signal-to-Noise Ratio (PSNR) It is defined by the ratio between the maximum possible value according to the numerical data format (as an example, for an `uint8` and a `float` the maximum values are 255 and 1, respectively), and the MSE (formulated above in eq. (2.4) but removing the root). The larger the value of PSNR, the better the fusion results. It is usually expressed as a logarithmic quantity using the decibel (dB) scale, as follows:

$$\mathrm{PSNR}(x,\hat{x}) = 10\log_{10}\left(\frac{\mathrm{MAX}^2}{\mathrm{MSE}(x,\hat{x})}\right) \tag{2.10}$$

where MAX is the maximum possible pixel value of the image according to the data format (as described above), and MSE is the mean squared error, formulated in eq. (2.4) but removing the root.

Structural Similarity Index (SSIM) Both PSNR and RMSE are commonly used due to their ease-of-use, and the fact that they have clear physical meanings. However, they do not provide a good value for the perceived visual quality [127]. To address these issues, Wang et al. [127] proposed SSIM to measure the similarities between the estimated image and the reference image. This metric takes into account the biological factors of the human visual system. The similarity between images is a product of luminance, contrast, and structure [127]. The larger the value of SSIM, the better the fusion results, with the best value being +1. Combining the three functions and assuming they all have the same relative importance, it can be defined as follows [144]:

$$\mathrm{SSIM}(x,\hat{x}) = \frac{\left(2\mu_x\mu_{\hat{x}}+C_1\right)\left(2\sigma_{x\hat{x}}+C_2\right)}{\left(\mu_x^2+\mu_y^2+C_1\right)\left(\sigma_x^2+\sigma_{\hat{x}}^2+C_2\right)} \tag{2.11}$$

where $\sigma_{x\hat{x}}$ is the correlation between x and $\hat{x}$; μ_x and $\mu_{\hat{x}}$ are the mean values for x and $\hat{x}$, respectively); σ_x and $\sigma_{\hat{x}}$ are the standard deviation for x and $\hat{x}$, respectively; and C_1 and C_2 are small stabilizing (data type dependent) constants which are required for when $\mu_x^2 + \mu_y^2$ and $\sigma_x^2 + \sigma_{\hat{x}}^2$ are zero, respectively, to avoid unstable results.

Multi-scale Structural Similarity Index (MS-SSIM) Proposed by Wang et al. [126], this metric based on SSIM provides more flexibility than the single-scale approach through the incorporation of image resolution variations and viewing conditions changes. The larger the value of MS-SSIM, the better the fusion results. The system applies a low-pass filter and a 2-factor downsample to the image in an iterative way. The initial image is denoted as scale M, and the highest scale is M, obtained after $M-1$ iterations. Within this metric, the overall evaluation is computed by combining the measurement at different scales using the following formula:

$$\mathrm{MSSIM}(x,\hat{x}) = [l_M(x,\hat{x})]^{\alpha_M} \cdot \prod_{k=1}^{M} [c_k(x,\hat{x})]^{\beta_k} \, [s_k(x,\hat{x})]^{\gamma_k} \tag{2.12}$$

where M represents the highest scale being used to test; at the k-th scale, the contrast comparison and the structure comparison are calculated by $c_k(x,\hat{x})$ and $s_k(x,\hat{x})$, respectively, using the both input HS images; the luminance is only computed at scale M and is denoted by l_M; and, the exponents α_M, β_k and γ_k are used to adjust the relative importance of different components - these are commonly fixed and equal to $+1$.

Block Sensitive - Peak Signal-to-Noise Ratio (PSNR-B) Yim and Bovik [144] proposes this metric based on PSNR but modifies it by including a blocking effect[14] factor. Contrarily to SSIM, this metric is designed specifically to assess blocky and deblocked images but has no proven perceptual significance. The larger the value of PSNR-B, the better the fusion results.

Universal Image Quality Index (UQI) Also referred to as UIQI or simply as Q, is named "universal" since it does not depend on the images being tested, the viewing conditions, or the individual observers. It combines the loss of correlation, contrast distortion and luminance distortion [125]. The larger the value of UQI, the better the fusion results, with the best value being $+1$. It can be mathematically described as follows:

[14] Effect caused by image compression techniques which segment the images into small blocks that are processed independently, with this effect occurring along block boundaries.

$$Q(x,\hat{x}) = \frac{1}{B}\sum_{j=1}^{B}\frac{4\mu_{x_j}\mu_{\hat{x}_j}}{\mu_{x_j}^2+\mu_{\hat{x}_j}^2}\frac{\sigma_{x_j\hat{x}_j}^2}{\sigma_{x_j}^2+\sigma_{\hat{x}_j}^2} \tag{2.13}$$

where $\sigma_{x_j\hat{x}_j}$ is the correlation between the bands x_j and $\hat{x}_j$; σ and μ represent the variance and mean, respectively, over all pixels of an image at a single band; and the other variables were already defined. Q is computed for each band, which is then averaged over all bands of the HS cube.

Spectral Angle Mapper (SAM) This technique by Yuhas et al. [150] determines the spectral similarity between two HS cubes. The smaller the value of SAM, the better the fusion results, with the best value being 0. The result of the comparison is presented as the angular difference (in radians) according to the following equation:

$$\mathrm{SAM}(x,\hat{x}) = \frac{1}{N}\sum_{i=1}^{N}\arccos\frac{x_i\cdot\hat{x}_i}{\|x_i\|\,\|\hat{x}_i\|} \tag{2.14}$$

where x_i and $\hat{x}_i$ are two spectra at pixel index i; $\|x_i\|$ and $\|\hat{x}_i\|$ represent their norm; and $\cdot$ represents the dot product operation. Additionally, each spectrum is treated as a B-dimensional vector. The similarity between each pair of spectra is determined without taking into account their relative brightness values.

Spectral Information Divergence (SID) Introduced by Chang [20], it compares the similarity between two pixels by measuring the probabilistic discrepancy between their corresponding spectral signatures. Similarly to SAM, it analyzes the data on a per-pixel basis and not per band, so instead of comparing matrices, it compares vectors. Since it measures the divergence, the smaller the value of SID, the better the fusion results.

Visual Information Fidelity (VIF) This metric proposed by Sheikh and Bovik [99] computes the relationship between image information and visual quality. It uses the information theoretic criterion for image fidelity measurement using natural scene statistics (NSS), human visual system (HVS) and an image distortion model. It performs well under both single-distortion and cross-distortion scenarios. The larger the value of VIF, the better the fusion results

$Q2^n$ This metric is an extension of the Universal Image Quality Index (UQI) to MS and HS images through hypercomplex numbers [40]. It is based on the computation of the hypercomplex correlation coefficient between the two images. This metric is a generalization of UQI which is able to measure at the same time both spectral and spatial distortions, contrary to the base metric which just averaged the spatial distortions of the bands.

3. Practical Application

The recent advancements in the field have continuously improved and have outperformed the pre-existing super-resolution state-of-the-art methods by ensuring higher quality, both in spatial reconstruction and spectral fidelity as stated by Mei et al. [83]. However, the existing comparisons between HIF methods which analyze their results have several discrepancies between each other, and also assume testing conditions that are not viable in a real-world scenario. This will be the focus of the discussion in this chapter.

In the previous section, we have introduced Wald's protocol [116] which is widely used and accepted as being the *de facto* standard for the benchmarking and analysis of HIF methods. Moreover, in appendix B we list a compilation of existing HIF methods available. Despite almost all those methods using this protocol, they differ in terms of testing conditions and cannot be directly compared across different papers. The next sections in this chapter: (1) introduce the issues of existing benchmarking conditions used across different papers; (2) propose clear methodologies to directly tackle those issues; and (3) demonstrate an implementation of said methodologies in a practical pipeline for HIF benchmarking.

3.1 Issues in HIF Testing

Although methods are tested and compared using Wald's protocol as their basis, there are several issues with existing implementations that hinder the accuracy and applicability of the outcome of the simulation. In this section, we describe those issues and their consequences that impede a fair, comparable, and real-world testing environment.

A major factor that makes the testing of said techniques difficult is the lack of **publicly available code** in numerous methods - see section B.3. Although metrics and visual results are presented, the lack of public code makes it impossible to test different datasets and/or different testing conditions. Furthermore, this also makes it difficult to ascertain what the actual algorithm is, since it is common to have certain subtleties which are usually only implemented in real code and not described in papers.

In turn, when other methods need to be compared with the ones that do not have public code, they re-use the metric values published in literature [6]. However, that does not provide a fair comparison since the testing conditions will not necessarily be exactly the same.

Moreover, there are several factors that influence the results, and justify the need to run the methods under the **same conditions** to ensure an objective and fair comparison between them. From the testing conditions that influence the results, the following can be highlighted:

- images can be stored in different formats (`uint8`, `float`, among others) with different scales (range in the data), which cause the unit-dependant metrics to be different even with equal results [115];
- when simulating input images there are several techniques that can be used, leading to the same dataset having different simulated images according to the chosen combination of techniques [5, 104] - for example, the low-resolution HS image is influenced by the blurring and downsampling techniques; the RGB images depends on the spectral response chosen; and both depend on the extra stages that might be added to the pipeline, such as denoising [147];
- the addition of result stabilizers either in the form of denoisers [147], or as extensions (see section B.4) [114, 122] at the end of the pipeline to produce improved results; when comparing two methods where one has a stabilizer, the other method should also be tested with that module to guarantee fairness when comparing results[1], since the results difference might be on the stabilizer and not in the actual method that is being compared.

On another note, since the available datasets include **noisy bands**, such as CAVE, the fusion methods are not able (purposefully) to reconstruct those noisy bands accurately (with respect to the reference image which does include noise). Therefore, their end result ends up being an image higher in visual quality and with less noise, but one which will perform poorly in the aforementioned metrics which compare it with the GT image - see fig. 3.1. For this reason, those bands are usually ignored when computing the quality metrics [5, 104, 154, 157]. Additionally, the metric(s) to select the noisy bands is/are not clearly described, and appear to be observer-dependent.

[1]If we have two methods and we want to test a stabilizer, both should be tested with that extension but just the best result of each method (either with or without extension) should be considered since an extension might have nefarious implications to a method.

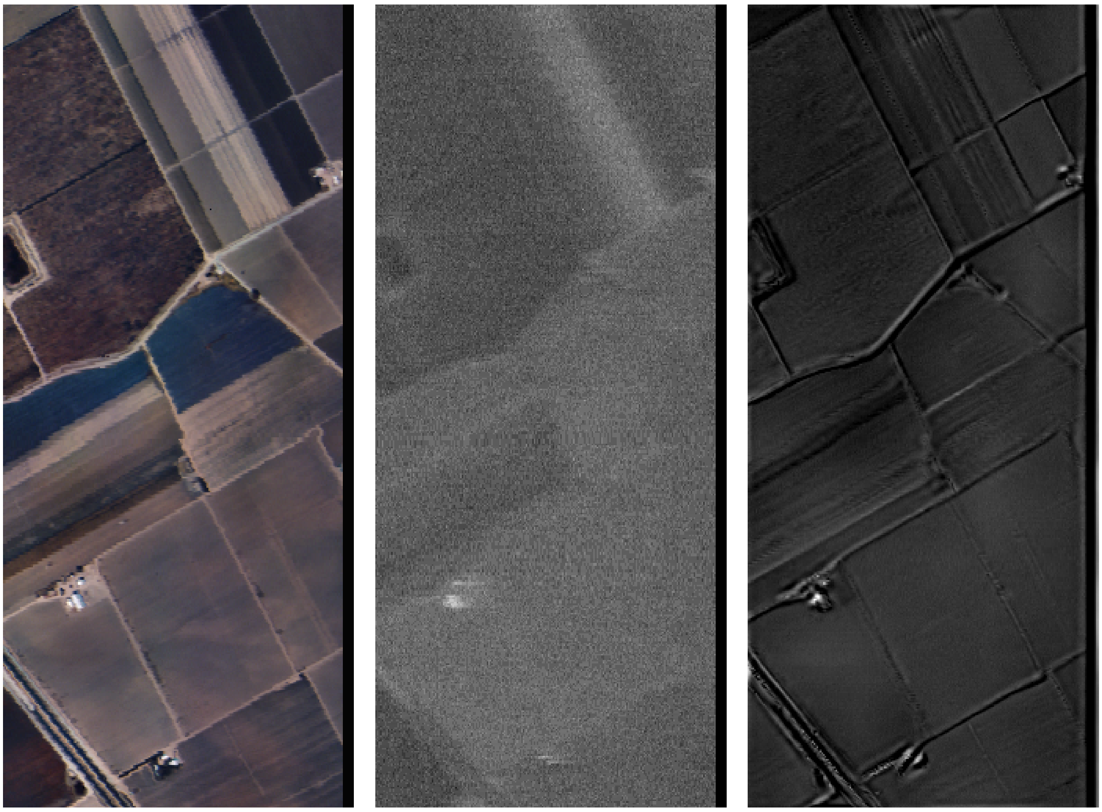


Figure 3.1: Application of the HIF method CNN-FUS [30] to the remote sensing scene Salinas from the EHU dataset, where the first and last bands are noisy. The image on the left is a color composite simulated from the HS cube; the image on the middle is the first band of the GT image; and the image on the right is the first band of the result of the image fusion. The contrast of the reconstructed image (right) is increased to aid the visual analysis; if the same operation was performed to the GT image (center) it would have increased the noise, however, this was not applied for an impartial analysis.

Another issue arises from the lack of a diverse set of spatial **scaling factors** used to downsample the GT image to generate the low-spatial resolution HS image. If a method only uses a single scaling factor [5, 6], it is not possible to ensure that it performs equally well across other scaling factors, or if the method is only suitable for that selected scaling factor - since it could have only published under the scaling factors that it performs the best. Therefore, multiple scaling factors must be used when comparing and evaluating different HIF methods, to enable a fair comparison.

Additionally, different authors use **different metrics** to compare their methods with others - for example, one method might be evaluated using SAM and the other using RMSE, both might be performing as the best method when evaluated with that specific metric, but for a fair comparison we need to compute several common metrics across both methods. This further hinders the direct comparison of results and is another argument for running the code locally. On top of that, some metrics cannot be directly compared across images or scaling factors, for example, RASE and RMSE are not independent of the ratio between the higher and the lower spatial resolutions of the input images; and the latter is also not unit-independent [96, 115].

Moreover, several methods are dependent on the **initial parameters**, and the selection of these parameters is of the utmost importance to obtain the best results. Therefore, these values should be predefined by default or learned according to the input. For example, in NSSR [32] these parameters are dependent on the point spread function (PSF) (a uniform blur with a kernel size of 8, 16, or 32; or a Gaussian blur all have different parameters accordingly). Another example is in BSR [129], where the parameter `maxAtoms` is defined according to the input image. This is only possible because, in a simulation environment, one can iterate these values to obtain the best results possible since one has access to the ground truth image. However, it is not possible to obtain these cherry-picked values in a practical environment, and the published results are not representative of a real-world application.

A common application for HIF is in the remote sensing field, therefore it is common to **only use remote sensing scenes** to test and compare methods [147]. However, HSI is broader than that particular field, and although a method might be great for that particular type of data, it might lack quality for indoor/outdoor scenes, artificial/natural objects, shapes, and colors that are not available in satellite imagery, and so forth [66]. Therefore, to fully compare the fusion methods, different datasets with different specifications and characteristics should be used.

Datasets of hyperspectral images are not common due to their high disk space usage and the need for specialized equipment to capture said images. However, when dealing with deep-learning techniques, the models are trained on the very **few available datasets** [103] - listed on section 2.4.

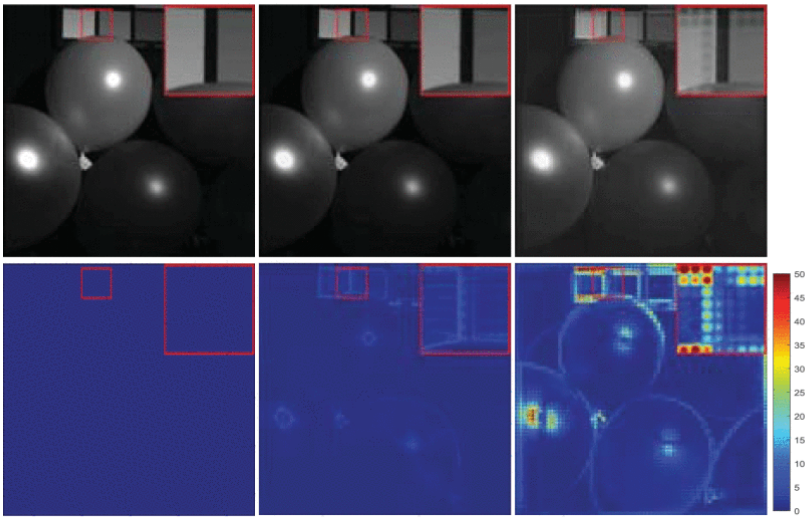
When simulating the two input images (HS and RGB) for Wald's protocol, they both have a common starting point - the GT reference image - which leads to accurately aligned input images. However, that is not the case for a real-world application, where the images are sourced from two distinct cameras: an HS camera and an RGB camera. Therefore, although most HIF algorithms assume there is an **accurate registration**, this is not the case for a practical application and just works for simulated test scenarios [88, 163].

Moreover, some methods require **knowledge of the PSF and/or spectral response function (SRF)** of the cameras to be able to run. If we are simulating the images we will have this information easily accessible and without any associated error, since these are the parameters used to generate the simulated inputs. However, that is not the case in a real-world scenario, since in practical applications it is unknown and it has to be estimated [32, 147]. In practice, the HIF methods can be divided into: (a) blind methods where the estimation of the camera models are optimized; and (b) non-blind methods which receive said parameters as input [120]. To compare both types of methods fairly, the non-blind methods should be required to also estimate those parameters and should not use the exact values that were employed to generate the simulated images - this has the goal of replicating uncertainty of the real-world.

Since most of the existing fusion methods have as a premise that the HS and RGB input images are exactly registered, they tend to perform poorly when that is not the case [88] - see fig. 3.2. Nevertheless, some of the recent methods already take that into consideration [16, 94, 160]. For example, Bungert et al. [16] propose to estimation of the characteristics the kernel during the reconstruction process instead of using a fixed kernel, which makes possible the estimation of the projective transformation between the input images in a dynamic manner.

Another issue comes from the different **distortions** that the two camera lenses have - see figs. 3.3 and 3.4. This is commonly unaddressed and it is not present in the simulated environment from Wald's protocol. If those distortions are not corrected beforehand, the images will not properly overlap and that will interfere with the accurate images co-registration, and consequently with the accurate super-resolution process.

Moreover, even if the images are undistorted beforehand, that process has an associated error that is not taken into account in the existing fusion methods. In a practical pipeline for HIF, the input images should be completely undistorted to ensure the best possible result.



(a) GT (b) Registered (c) Unregistered

Figure 3.2: Reconstructed result of the image "balloons" from the CAVE dataset using the NSSR [32] method with (b) registered (ideal situation) and (c) unregistered (1° rotation) input images. The top row provides the reconstructed 30$^{\text{th}}$ band, and the bottom row the reconstruction error map [88].

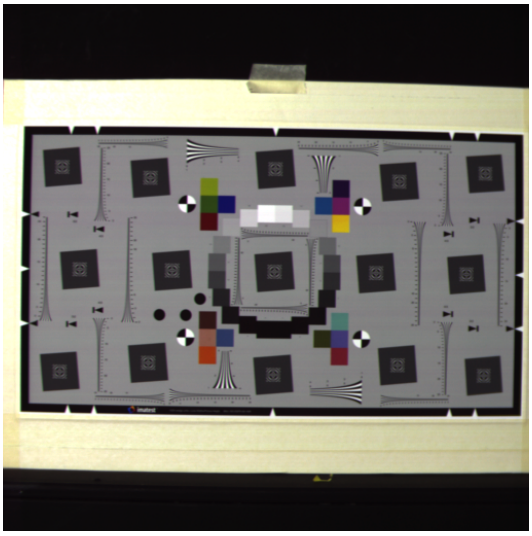


Figure 3.3: Color composite photo of an Extended eSFR ISO 12233:2017 chart from an HS image taken with a Specim IQ camera.

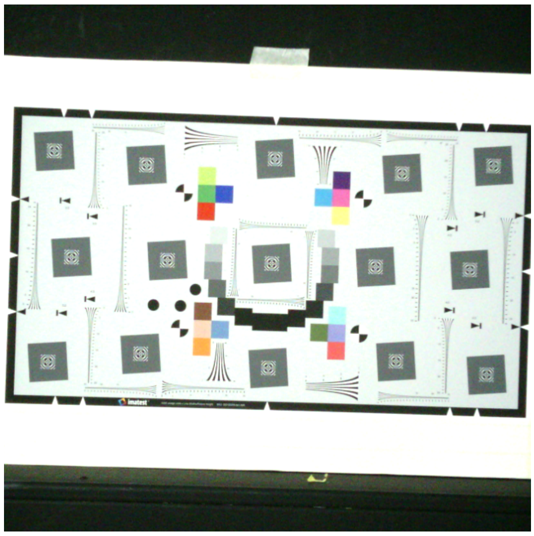


Figure 3.4: Photo of an Extended eSFR ISO 12233:2017 chart taken with the RGB camera of the Specim IQ camera.

This section can be summarized as follows: (1) different authors employ different testing conditions that lead to different results which are unfeasible to be directly compared with each other; and (2) the lack of real-world simulation conditions further hinder the ecological validity of the results which are being presented.

3.2 Proposed Testing Protocol

To address the previously described inconsistencies in HIF methods testing, we propose a fair and extensible testing protocol, based on Wald's protocol [116].

For a method to be fairly compared, its code must be publicly available and run under the **same conditions** as its peers, which means that:

- images are all in the same format (inc. data type);
- input images are simulated under the exact same conditions;
- if results stabilizers are added, they should be tested for all methods (with the best alternative for each the method being selected, either with or without said stabilizer);
- if noisy bands are removed for the computation of metrics, those exact bands are to be removed across all the methods in the same manner;
- SRF and PSF parameters are estimated, emulating a real-world scenario;
- the initial parameters are to be the ones proposed in the original paper, and should not be cherry-picked for each separate image.

Moreover, to guarantee the broad reach of the proposed protocol, it should be tested using multiple spatial scaling factors and distinct datasets with different characteristics - the images should include natural/artificial shapes, objects, colors; indoors/outdoors scenes; and remote sensing scenes from different continents in both urban and natural areas. Additionally, the results should be compared using different quality metrics which measure different quantities.

Finally, the testing pipeline can be extended by emulating real-world interferences to the simulated inputs, which can be in the form of noise, minor image distortions, and small geometric transformations to simulate errors in the co-registration.

3.3 Implementation of the Testing Protocol

Following the previously described proposal, we developed a testing protocol for hyperspectral image fusion (HIF) methods. It can be accessed using the following URL: https://github.com/magamig/hif-benchmarking/.

Within this repository, we developed wrappers for several methods - listed in section B.1 - which allowed us to run them automatically over a combinatory of scaling factors (downsampling factor), datasets, and their images - see section C.1. This implies that for each image of a dataset, the script will produce a number SR images equal to the number of scaling factors multiplied by the number of methods being tested.

```
SCALINGS = [4,8,16]
DATASETS = ["CAVE","EHU","Harvard"]
METHODS = ["CNMF","FUSE","SFIM","GSA","GLP","GSOMP",\
    "NSSR","SupResPALM","CNNFUS","HySure","MAPSMM",\
    "LTTR","LTMR","CSTF","BayesianSparse"]
```

This repository includes a wide variety of **HIF methods** from 2000 to 2020, with different types of approaches being tested: pan-sharpening based, matrix factorization, bayesian-based, tensor-based and deep-learning.

Regarding the **scaling factors**, we test different factors to ensure that the results are not biased to a method that might perform good for one scaling factor, but poorly on another. Moreover, for ease of use and compatibility with all the methods, powers of two were selected. For reference, when testing remote sensing scenes, scaling factors are referred to as ground sampling distance (GSD) ratios, which indicate the ratio between the length in meters of a side of a terrain square represented by a pixel in the HS and RGB image.

In terms of **datasets**, the goal is to be as broad as possible and include exemplars from distinct images. For this reason, the following datasets were selected: CAVE, Harvard, and EHU. These datasets, described in section 2.4, include images of natural/artificial shapes, objects, and colors; indoors/outdoors scenes; and remote sensing scenes from different continents in both urban and natural areas. This variety of images is a requirement to adequately compare methods.

However, since images are stored under different formats and have their own particularities, they had to be pre-processed to standardize the input data - see code in sections C.2 to C.4. Moreover, although the CAVE dataset comes with the color composites (simulated RGB images) of the scenes, the ones from the Harvard and the EHU datasets had to be simulated - for reference see fig. 2.6.

For the Harvard dataset, the RGB image is simulated by integrating the ground truth over the spectral dimension, using spectral response function (SRF) of the camera Nikon D700[2] [6], as follows:

```
srf = np.array([
    [0.005,0.007,0.012,0.015,0.023,0.025,0.030,0.026,0.024,...],
    [0.000,0.000,0.000,0.000,0.000,0.001,0.002,0.003,0.005,...],
    [0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,...]
]).T
gt = scipy.io.loadmat(mat_path)['ref']
rgb = numpy.dot(gt,srf)
```

$$\mathbf{RGB} = \mathbf{GT} @ \mathbf{SRF} \tag{3.1}$$

where $\mathbf{GT} \in \mathbb{R}^{W \times H \times \Lambda}$, $\mathbf{SRF} \in \mathbb{R}^{\Lambda \times 3}$, $\mathbf{RGB} \in \mathbb{R}^{W \times H \times 3}$, and @ represents the matrix multiplication operation. For this particular dataset $W = 1040$, $H = 1392$ and $\Lambda = 31$.

For the EHU dataset, the data was more complex in terms of bands since (1) each image of the dataset had a different number of bands (Λ), and (2) some bands had been previously removed creating images with non-contiguous wavelength intervals. For these reasons, the previously mentioned approach was more difficult to implement for this dataset. Therefore, we used the `colorize` function from MATLAB, as follows:

```
load("data/GT/EHU/Indian_pines.mat");
hcube = hypercube(indian_pines,
   [1:103,109:149,164:219,221:240]*10+400);
msi = colorize(hcube, "Method", "rgb", "ContrastStretching", true);
save("data/MS/EHU/Indian_pines.mat", "msi")
```

Due to the non-contiguous wavelength intervals of these HS images, these color composite simulated (RGB) images are not an accurate representation as the previous ones where the wavelength intervals were contiguous.

Regarding the downsampling process to generate the smaller spatial resolution HS image, we used the Lanczos resampling [34, 109] to resize the reference GT image. This filter was selected since it achieves the best downscaling quality, although at the cost of performance which is not relevant for this application [24].

This implementation can be described as the *bare-bones* implementation of the proposed testing protocol which addresses the consistency of evaluation conditions for the direct comparison of results from HIF methods. Furthermore, although not

[2] http://www.maxmax.com/spectralresponse.htm

present in this implementation, it can also be easily extended to deal with the lack of real-world simulation conditions - this is further described in section 5.1.

In the next chapter, we analyze the results of this testing protocol and compare the results of the implemented methods across the different scaling factors and the selected datasets.

4. Results & Discussion

As described in section 3.3, several hyperspectral image fusion methods were implemented in the testing protocol through method-specific wrappers. To analyse these methods, they were first applied to a HS image of an eSFR ISO 12233:2017 chart. Then, the resulting super-resolution hyperspectral images were analyzed along both the spatial and the spectral dimensions from which we did a preliminary selection of methods - this limits the number of methods to be tested later on, since those tests that follow are time-consuming.

Afterwards, the selected methods were applied to a comprehensive set of datasets that include indoors/outdoors, natural/artificial images, and remote sensing scenes: CAVE [142], Harvard , and EHU - previously described in section 2.4. These results were then analysed in both a numerical as well as a visual manner.

4.1 Preliminary Selection of Methods

In this section, we do a preliminary selection of the implemented HIF methods. This method selection is performed according to their performance with an image of an eSFR ISO 12233:2017 chart, along both the spatial and the spectral dimensions.

To test the spatial dimension, we analyse the results of each method when applied to an image of a resolution wedge. Figure 4.1 presents the results of these method, with each row representing a distinct method, and each column represents a scaling factor (4x, 8x and 16x) used within Wald's protocol. The first row contains the input downsampled HS image output for the different scaling factors. Both the HS GT and RGB images are omitted for brevity.

Upon visual inspection of these results, we can observe structural artifacts in the output of certain methods: LTTR [29], GSOMP [5], MAPSMM [35], CNN-FUS [30], SFIM [72], NSSR [32], and LTMR [26]. Therefore, these methods were not further considered in our analysis.

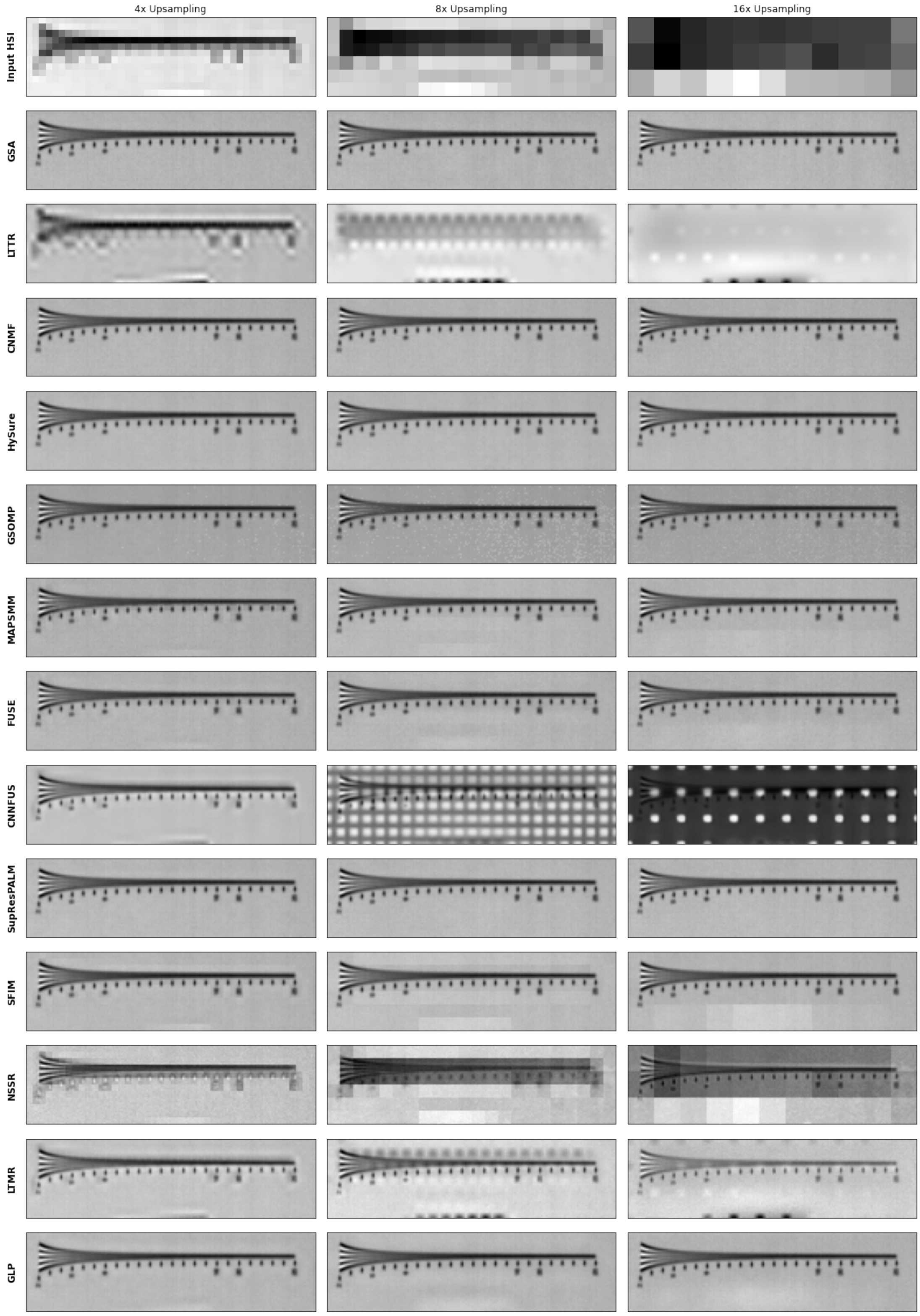


Figure 4.1: Comparison of HIF methods applied to a resolution wedge across different scaling factors to the band at 490 nm. The first row represents the input downsampled image, and thee following rows represent the different methods output SR image; the columns represent the scaling factors 4x, 8x and 16x, respectively.

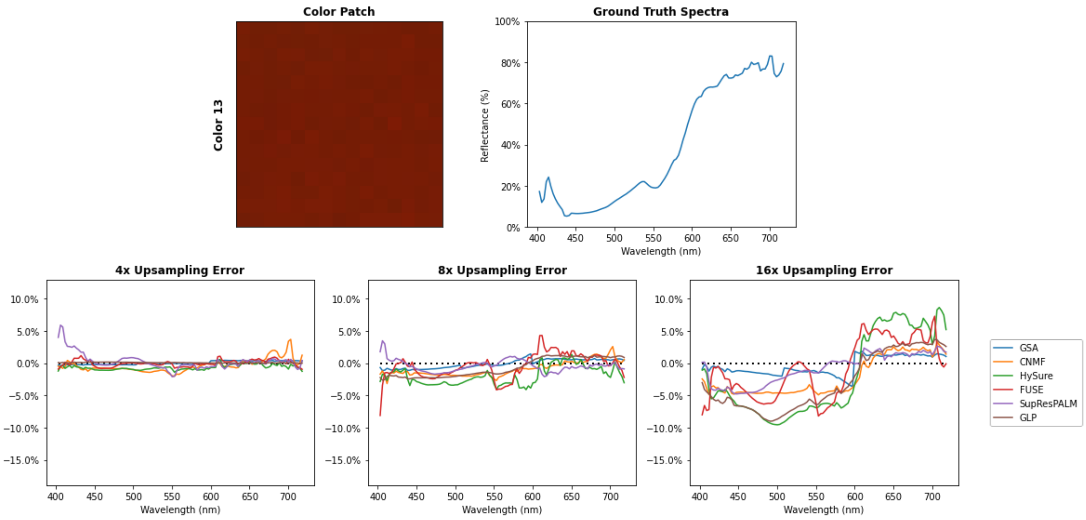


Figure 4.2: Color Patch 13 of the eSFR ISO 12233:2017 chart, with its ground truth spectrum, and the errors between the ground truth spectrum and the result of HIF methods at scaling factors of 4x, 8x and 16x.

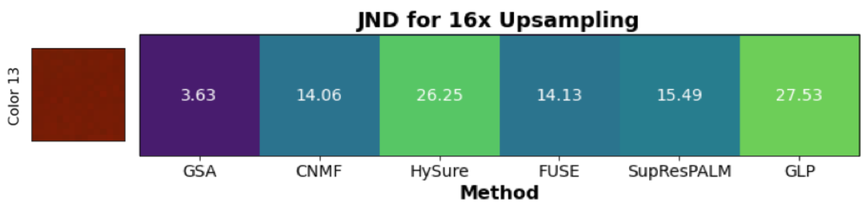


Figure 4.3: Color Patch 13 of the eSFR ISO 12233:2017 chart, with its JND for different HIF methods at a spatial scaling factor of 16x.

Following this structural and resolution analysis, a spectral fidelity analysis was performed to the other methods: GSA [3], CNMF [146], HySure [104], FUSE [130], SupResPALM [67], and GLP [2]. For reference, we present an analysis of the spectral dimension of a sample color patch in fig. 4.2, with the full analysis being presented in appendix D. This analysis includes the color composite of each patch simulated from the HS cube; the mean ground truth of the spectra for that color patch; and, the error for scaling factors of 4x, 8x and 16x for the different HIF methods listed above.

As expected, an image fusion process with higher scaling factors is more complex. Therefore, the error for a 16x scaling factor is generally higher than the error for a 8x scaling factor, which in turn is also generally higher than the error for a 4x scaling factor. This is clearly visible in fig. 4.2, where the errors in terms of JND are distant from the ideal super-resolution result represented by the dashed line, where the error is nonexistent. The JND was computed using CIELAB with CIE Illuminant E.

Moreover, figs. D.6 to D.8 present the JND of the reconstructed color patch according to difference between the reconstructed spectrum (from the SR HS image) and the GT reference spectrum, for each upsampling factor, HIF method, and color patch. A sample color patch is presented in fig. 4.3.

From an analysis of these graphs, it is clear that some methods perform better than others. For example, GSA [3] is able to provide good results across different scaling factors, whilst GLP [2] provides good results at a scaling factor of 4x but performs worse at higher scaling factors - its results are not stable across scaling factors.

Taking into account the results from the previously described analysis, no technique was disregarded since the results were relatively close to each other, and their performance for these specific colors might not be representative of real datasets. Therefore, **the final methods selected for further analysis were the following:**

GLP Introduced by Aiazzi et al. [2], this technique obtains the spatial details for each band through the difference between the high-spatial resolution image and its low-pass version multiplied by a gain factor. It was adapted from **pan-sharpening** (see section 2.3.3) to the generalized HIF through hypersharpening [98].

GSA Proposed by Aiazzi et al. [3], this method improves previously existing methods by taking into account the impact of the SRF on the fusion process. It adds the spatial details to the low-spatial resolution HS image by multiplying the difference between the high-spatial resolution image and a synthetic intensity component, by a band-wise modulation coefficient. This synthetic intensity component is computed through a linear regression between the two input images, mitigating spectral distortions. Adapted from **pan-sharpening** (see section 2.3.3), this method provides good spectral results which were corroborated by our findings present in appendix D.

CNMF This method is based on **nonnegative matrix factorization (NMF)** (see section 2.3.4) where the low-resolution HS image is unmixed by nonnegative matrix factorization (NMF), and from the high-spatial resolution image we obtain the high-resolution abundance maps through a least squares regression [11]. In CNMF, end-members and abundance maps are estimated via spectral unmixing based on NMF, whilst taking into account the camera model which incorporates both SRF and PSF. The ouput high-resolution HS image is computed through the product of the spectral signatures and the high-resolution abundance maps.

HySure This technique, proposed by Simoes et al. [104], introduces the total variation regularization, which affects the structural and resolution fidelity of the results since it preserves edges whilst smoothing out noise in homogeneous areas. This **bayesian-based** (see section 2.3.5) method fuses the input image through a minimization of a convex objective function [147].

SupResPALM This method, introduced by Lanaras et al. [67], SupResPALM unmixes the two input images into the end-members spectral signatures (pure spectra) and their corresponding mixing coefficients (fractional abundances) - see section 2.3.4. This **matrix factorization** method encompasses constrains based on elementary physical properties of spectral mixing.

FUSE Introduced by Wei et al. [130], FUSE uses a Sylvester equation to solve the maximization problem of a forward model which represents the likelihoods of the observations. It can be generalized to incorporate prior information for the fusion problem, allowing a **Bayesian** estimator - see section 2.3.5. Moreover, its computational complexity is significantly lower than its peers.

For a more detailed description of each method please refer to the cited papers. These methods were then run across three scaling factors - 4x, 8x, and 16x - and three distinct datasets - CAVE, Harvard, and EHU (described in section 2.4).

4.2 Analysis on Datasets

In this section, the results of previously selected methods are analyzed in both a numerical as well as a visual manner, according to their result when applied a diverse set of datasets - CAVE [142], Harvard [19], and EHU - following Wald's protocol. These datasets were selected due to their diversity of data from indoors/outdoors and natural/artificial images with a wide range of materials, objects, shapes, and colors; but also including remote sensing scenes.

4.2.1 Numerical Analysis

Several full-reference quality assessment metrics were presented in section 2.5. These metrics determine the similarity between the HS GT reference image and the estimated high-spatial resolution HS image from Wald's protocol. For this review, we select some of the most widely used metrics across for the three dimensions: spatial (SCC), spectral (SAM) and global (SSIM).

These metrics were computed across different scaling factors (4x, 8x, and 16x) to the output of the previously selected HIF methods to all the images of the 3 datasets. The results of this extensive analysis are present in appendix E. For brevity, in this section, we present the average of the quality metrics for the CAVE dataset in table 4.1, for the Harvard dataset in table 4.2, and for the EHU dataset in table 4.3.

Upon close inspection of the results, it is clear that the higher the scaling factor applied to the image, the worse the output of the HIF methods. This is true for all methods. However the methods GSA and SupResPALM are more stable across different scaling factors than the others. This is an intuitive result since the lower the scaling factor, the more HS information we have available, and the easier it is to reconstruct the SR HS images.

Moreover, from the selected methods using the selected quality metrics, the top-performer with the CAVE dataset is GLP, and with the Harvard dataset it is GSA. This fact raises the question that when choosing a method, one must take into account the type of HS image we are dealing with some methods perform better with remote sensing scenes, others with indoors scenes, etc.

Additionally, GSA might perform better than CNMF in the spectral dimension, according to SAM, it performs worse in the spatial dimension, according to SCC (when applied to the CAVE dataset). Therefore, when choosing a HIF method, one must

Scaling Factors	HIF Methods	Quality Metrics		
		SSIM ↑	SCC ↑	SAM ↓
4x	GLP	0.967	0.574	0.078
	GSA	0.870	0.518	0.173
	CNMF	0.918	0.487	0.143
	HySure	0.892	0.490	0.167
	SupResPALM	0.867	0.504	0.282
	FUSE	0.910	0.467	0.146
8x	GLP	0.945	0.569	0.105
	GSA	0.870	0.528	0.178
	CNMF	0.914	0.490	0.141
	HySure	0.863	0.494	0.209
	SupResPALM	0.867	0.506	0.282
	FUSE	0.879	0.458	0.181
16x	GLP	0.919	0.558	0.133
	GSA	0.867	0.529	0.186
	CNMF	0.903	0.479	0.148
	HySure	0.827	0.510	0.241
	SupResPALM	0.866	0.497	0.281
	FUSE	0.858	0.469	0.197

Table 4.1: Average of the quality metrics measured across different HIF methods and three scaling factors (4x, 8x and 16x) applied to the **CAVE** dataset dataset as per Wald's protocol.

Scaling Factors	HIF Methods	Quality Metrics		
		SSIM ↑	SCC ↑	SAM ↓
4x	GLP	0.954	0.543	0.058
	GSA	0.955	0.547	0.048
	CNMF	0.953	0.539	0.056
	HySure	0.969	0.542	0.055
	SupResPALM	0.972	0.534	0.057
	FUSE	0.957	0.539	0.052
8x	GLP	0.942	0.540	0.064
	GSA	0.946	0.545	0.053
	CNMF	0.948	0.530	0.071
	HySure	0.960	0.542	0.061
	SupResPALM	0.962	0.530	0.063
	FUSE	0.941	0.537	0.056
16x	GLP	0.933	0.537	0.068
	GSA	0.937	0.545	0.057
	CNMF	0.933	0.529	0.073
	HySure	0.945	0.538	0.068
	SupResPALM	0.924	0.521	0.073
	FUSE	0.930	0.534	0.061

Table 4.2: Average of the quality metrics measured across different HIF methods and three scaling factors (4x, 8x and 16x) applied to the **Harvard** dataset dataset as per Wald's protocol.

Scaling Factors	HIF Methods	Quality Metrics		
		SSIM ↑	SCC ↑	SAM ↓
4x	GLP	0.878	0.401	0.239
	GSA	0.885	0.508	0.243
	CNMF	0.867	0.450	0.354
	HySure	0.783	0.425	0.345
	SupResPALM	0.802	0.471	0.334
	FUSE	0.727	0.346	0.384
8x	GLP	0.843	0.395	0.276
	GSA	0.845	0.501	0.267
	CNMF	0.840	0.459	0.351
	HySure	0.727	0.444	0.416
	SupResPALM	0.809	0.479	0.340
	FUSE	0.684	0.331	0.416
16x	GLP	0.821	0.398	0.308
	GSA	0.802	0.500	0.293
	CNMF	0.842	0.432	0.316
	HySure	0.717	0.459	0.440
	SupResPALM	0.786	0.481	0.346
	FUSE	0.683	0.354	0.441

Table 4.3: Average of the quality metrics measured across different HIF methods and three scaling factors (4x, 8x and 16x) applied to the **EHU** dataset as per Wald's protocol.

consider what is the target goal of said process (spectral accuracy, accurate colorimetric results, visual structural fidelity, among others) and then compare the HIF methods accordingly.

4.2.2 Visual Analysis

Even though the quality metrics might be numerically similar, a visual analysis might display distinct images with different characteristics and/or reconstruction errors. Therefore, for a complete evaluation of the output of HIF methods, a visual analysis of the results must be performed to better understand the structure and visual fidelity of the SR HS images. In this subsection, we present a visual analysis for of the output of the selected HIF methods with a scaling factor of 16x for one sample image per dataset: CAVE in fig. 4.4, Harvard in fig. 4.5, and EHU in fig. 4.6.

In figs. 4.4 to 4.6, each row represents a different HIF method. In the columns, the first one is a gray image of the band number 15 of SR HS image from that method, the second and thirds column are the RMSE and SSIM maps, respectively.

In the aforementioned figures, the darker the RMSE and SSIM maps the better, with the highlighted regions being the problematic areas of the SR HS image that differ the most from the GT HS image.

In these three exemplars, the one which achieves the best results is in the Harvard dataset - fig. 4.5 - where the images have much darker maps even with a scale from 0 to 0.1. This cannot be directly compared with the figs. 4.4 and 4.6, since these use a scale from 0 to 0.5. However, this would further exacerbate the difference between them.

Moreover, homogeneous regions tend to perform better than structurally complex regions such as the hair of the stuffed toy in fig. 4.4. Additionally, borders between distinct regions also tend to be areas of inferior results. These two statements are true across different methods, datasets, and scaling factors.

For a complete evaluation of a method, both a numerical and a visual analysis must be taken into account to find the most suitable HIF method for our needs. Although some methods might be generally superior than others, there is not a one fits all solution. Therefore, a proper evaluation and comparison must be performed to obtain the best results for a specific use-case.

Figure 4.4: Visual analysis of RMSE and SSIM maps of the image "chart_and_stuffed_toy" from the CAVE dataset across different HIF methods with a scaling factor of 16x.

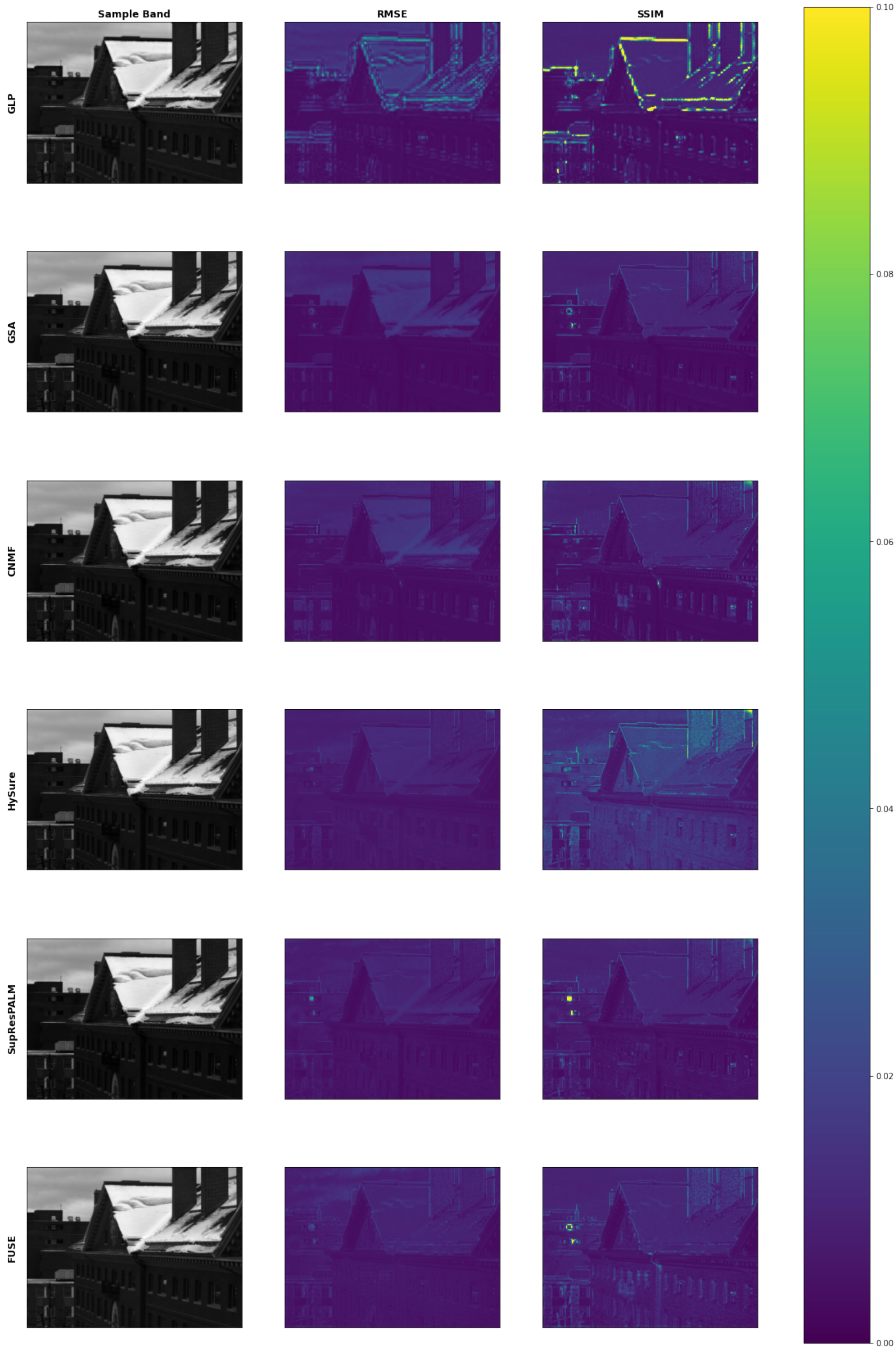


Figure 4.5: Visual analysis of RMSE and SSIM maps of the image "img1" from the Harvard dataset across different HIF methods with a scaling factor of 16x.

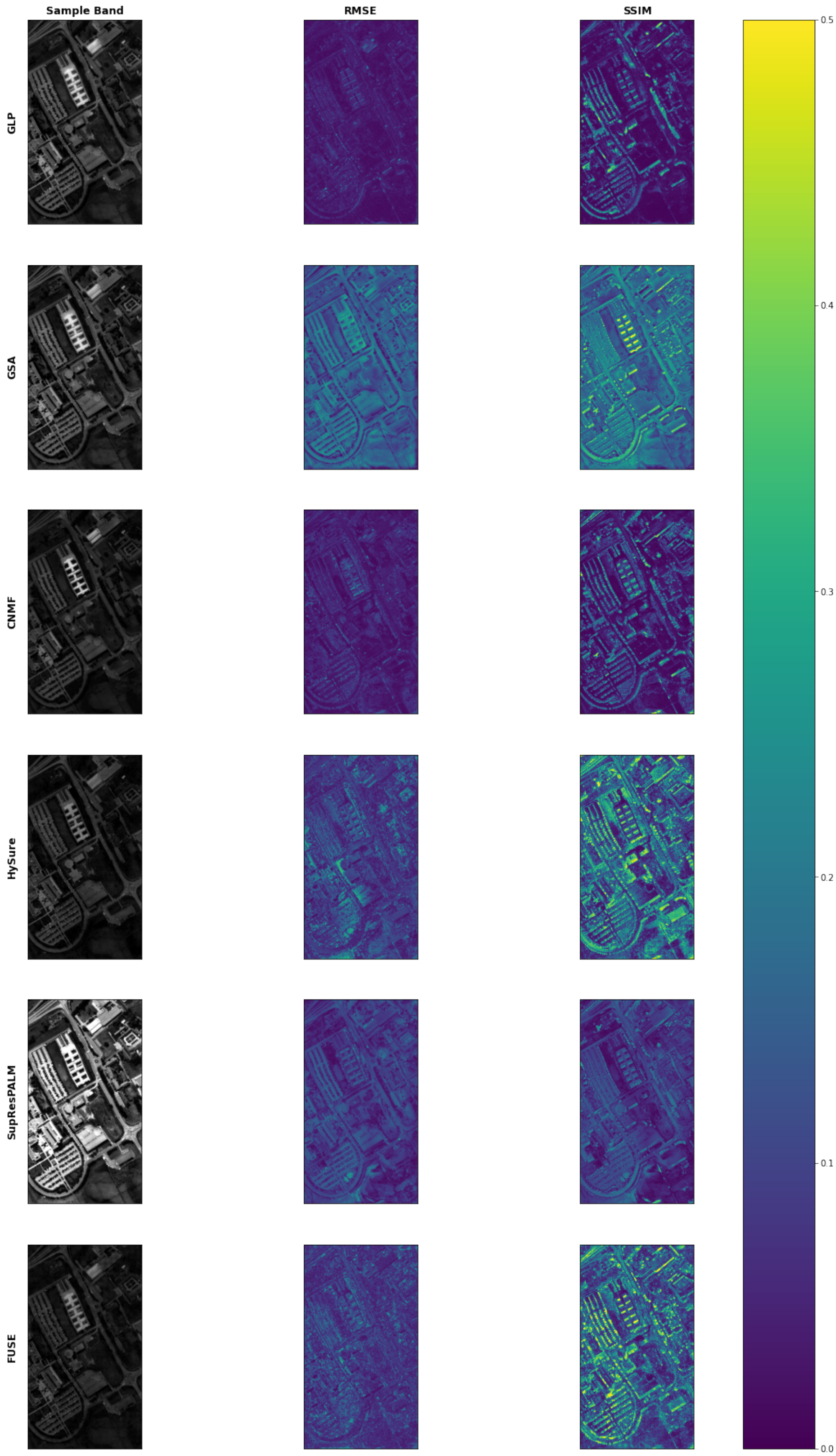


Figure 4.6: Visual analysis of RMSE and SSIM maps of the image "PaviaU" from the EHU dataset across different HIF methods with a scaling factor of 16x.

5. Conclusion

In this thesis, we have provided a comprehensive review of the state-of-the-art of hyperspectral image fusion. Furthermore, we analyzed how HIF methods are compared and some of the shortcomings of existing testing protocols.

To address these issues, we present **a generalized, fair, and extendable testing protocol** which demonstrates its applicability with several methods, datasets, and scaling factors being tested.

Afterward, we compared the obtained results across all methods, datasets, and spatial scaling factors through a numerical and also a visual analysis of the output super-resolution hyperspectral images, comparing it with the ground truth images as per the Wald's protocol.

Upon close inspection of the results, the selection of a HIF method is data-dependant. One must take into account the type of HS image that is being dealt with - some methods perform better with remote sensing scenes, others with indoors scenes, etc.

Moreover, the best method for a task might not be the best for a different task, and the selection of a method is dependent on the end goal. Even though the quality metrics might be numerically similar, a visual analysis might display distinct images with different characteristics and/or reconstruction errors. Therefore, the selection of a method needs to take all this into account and will need to compromise on either the spectral accuracy, colorimetric results, or even the visual structural fidelity.

Although this work provides a base testing protocol for the comparison of hyperspectral image fusion techniques, it is extendable and has room to encompass more real-world variables as described in section 5.1.

In summary, future developments in the field should take into account that the testing conditions should be equal across all methods and should emulate as good as possible the constraints encountered in a real-world scenario. Finally, the selection of a HIF method should be both data- and task-dependant.

5.1 Future Work

The protocol presented in this thesis addresses most of the shortcomings of other testing frameworks presented in section 3.1. However, due to its modularity, it can be easily extended to include other real-world interferences to the simulated inputs that are not present in the current implementation; which can be in the form of noise, minor image distortions, and small geometric transformations to simulate errors in the co-registration. This would lead to a multitude of new results, where one could evaluate the impact of said conditions on the result of the fusion process.

Another topic which requires further analysis is the lack of contiguous wavelength intervals in the images of the EHU dataset. Although commonly used to test HIF methods in the remote sensing field, the lack of some wavelength intervals interferes with the real-world simulation of RGB images. Moreover, this also leads to abrupt changes in the spectra when those removed intervals occur, which do not occur frequently in nature - this might impact the performance of some techniques that expect contiguous intervals (without "holes") which tend to act in a more "well-behaved" manner.

Moreover, the recent developments in HIF methods' extensions were not included in the present version - described in section 2.3.9 and listed in listed in section B.4. These extensions can be added to the end of the pipeline to improve the results of the fusion task. However their improvement varies according to the method being extended and its characteristics, therefore this should also be tested with the entire list of methods available.

Section B.1 lists the methods that are implemented in the repository (at the time of publishing); section B.2 lists other methods with their code publicly available but without an implemented wrapper. These methods were not implemented due to the limited time at our disposal. However, they can and should be added to our implementation through the development of custom wrappers for each method.

In short, this work serves as a basis for future developments in testing and comparison of fusion techniques, with the capability of being easily extended with the integration of other testing variables/parameters.

Acronyms

KSC Kennedy Space Center. 18

M

MS multispectral. 2, 5, 7, 12, 13, 25
MS-SSIM Multi-scale Structural Similarity Index. 18, 19, 23
MSE Mean Squared Error. 22

N

NMF nonnegative matrix factorization. 13, 40
NSS natural scene statistics. 24

P

PSF point spread function. 13, 29, 30, 32, 40
PSNR Peak Signal-to-Noise Ratio. 18, 19, 22, 23
PSNR-B Block Sensitive - Peak Signal-to-Noise Ratio. 18, 19, 23

R

RASE Relative Average Spectral Error. 18, 19, 21, 29
RGB Red Green Blue. v, 1, 2, 5, 7–9, 12–15, 27, 30, 31, 33, 34, 36, 50
RMSE Root Mean Squared Error. 18–22, 29, 45–48
RMSEB Root Mean Squared Error Band. 20

S

SAM Spectral Angle Mapper. 18, 19, 24, 29, 41–44, 73–135
SaU spatial unmixing. 11, 13
SCC Spatial Correlation Coefficient. 18, 19, 22, 41–44, 73–135
SHSR single hyperspectral image super-resolution. 5, 6
SID Spectral Information Divergence. 18, 19, 24
SR super-resolution. 7, 10, 14–16, 18, 33, 36, 37, 39, 41, 45, 49
SRF spectral response function. 13, 30, 32, 39, 40
SSIM Structural Similarity Index. 18, 19, 22, 23, 41–48, 73–135

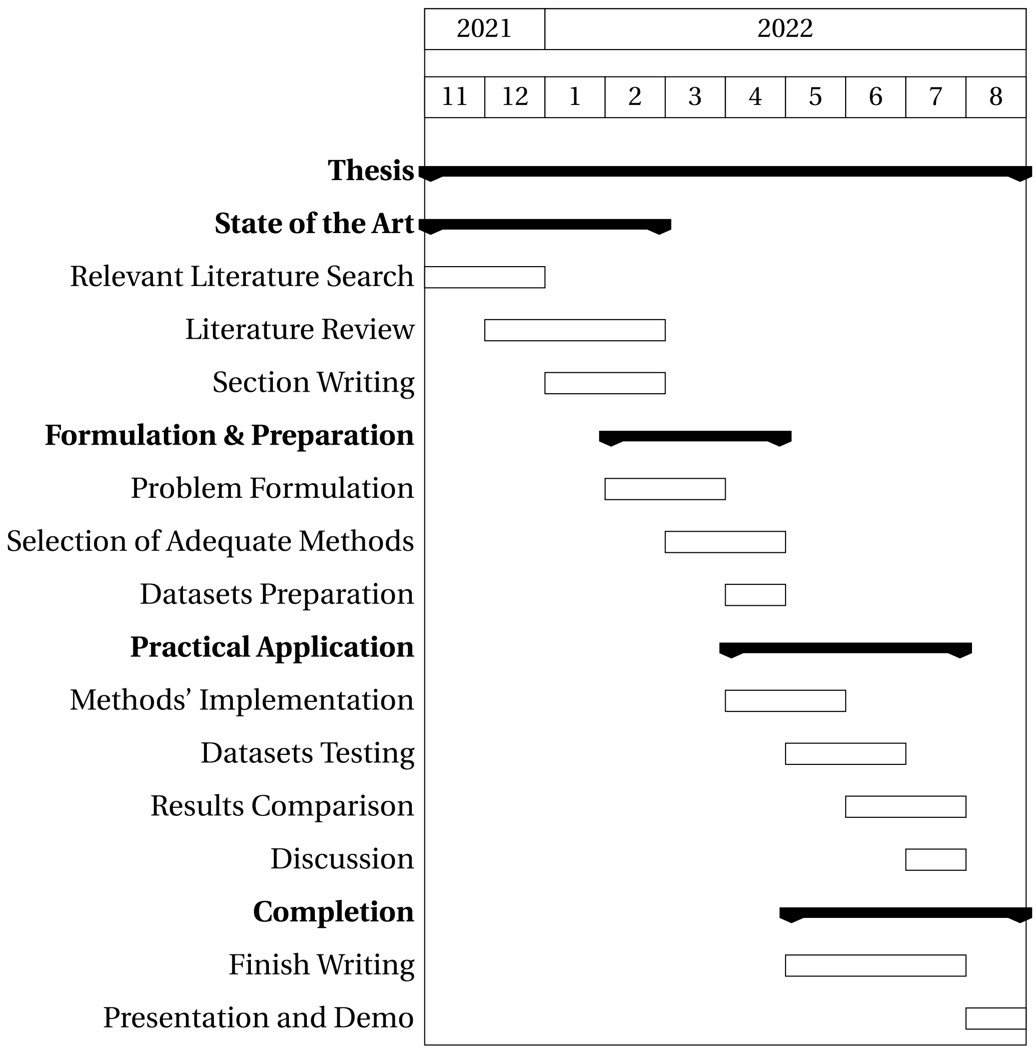
U

UIQI Universal Image Quality Index. 23
UQI Universal Image Quality Index. 18, 19, 23, 25

V

VIF Visual Information Fidelity. 18, 19, 24

A. Work Plan Schedule



B. List of HIF Methods

In this appendix, we list several hyperspectral image fusion (HIF) methods: section B.1 lists the methods that are implemented in the repository (at the time of publishing); section B.2 lists other methods with their code publicly available without an implemented wrapper; section B.3 lists other methods that do not have their code publicly available; and section B.4 lists extensions that can be added to any of the HIF methods and are able to improve the results of the image fusion process. Since some of the methods were only presented as "ours", and did not have a proper name, we have adopted a name or acronym used by other papers which referenced those. The printed URLs pointing to the code are present in the bibliography.

B.1 Implemented Methods in the Repository

Name	**Year**	**Author(s)**	**Code**
SFIM	2000	Liu [72]	Matlab
MAPSMM	2004	Eismann [35]	Matlab
GLP	2006	Aiazzi et al. [2]	Matlab
GSA	2007	Aiazzi et al. [3]	Matlab
CNMF	2011	Yokoya et al. [146]	Python / Matlab
GSOMP	2014	Akhtar et al. [5]	Matlab
HySure	2014	Simoes et al. [104]	Matlab
BayesianSparse	2015	Akhtar et al. [6]	Matlab
FUSE	2015	Wei et al. [130]	Matlab
SupResPALM	2015	Lanaras et al. [67]	Matlab

Name	Year	Author(s)	Code
NSSR	2016	Dong et al. [32]	Matlab
CSTF	2018	Li et al. [70]	Matlab
LTMR	2019	Dian and Li [26]	Matlab
LTTR	2019	Dian et al. [29]	Matlab
CNN-FUS	2020	Dian et al. [30]	Matlab

B.2 Other Methods with Code Available

Name	Year	Author(s)	Code
MF	2011	Kawakami et al. [62]	Matlab
SNMF	2013	Wycoff et al. [135]	Matlab
BSR	2015	Wei et al. [129]	Matlab
RGB-HIU	2015	Kwon and Tai [66]	Matlab
BlindFuse	2016	Wei et al. [132]	Matlab
FUMI	2016	Wei et al. [131]	Matlab
CMS	2018	Zhang et al. [153]	Matlab
BRS	2018	Bungert et al. [16]	Matlab
DHSIS	2018	Dian et al. [28]	Matlab
MSDCNN	2018	Yuan et al. [148]	Python
SSF-CNN	2018	Han et al. [46]	Python
STEREO	2018	Kanatsoulis et al. [61]	Matlab
uSDN	2018	Qu et al. [93]	Python
Two-CNN	2018	Yang et al. [140]	Matlab
SSU	2019	Zhou et al. [163]	Matlab
MHF-net	2019	Xie et al. [136]	Python
DBIN	2019	Wang et al. [120]	Python

Name	Year	Author(s)	Code
DHIP	2019	Sidorov and Hardeberg [103]	Python
HSI-CSR	2019	Fu et al. [38]	Caffe
CUCaNet	2020	Yao et al. [141]	Python
GDD	2020	Uezato et al. [110]	Python
TFNet	2020	Liu et al. [76]	Python
PZRes-Net	2020	Zhu et al. [164]	Python
Rec_HSISR_PixAwaRefin	2020	Wei et al. [134]	Python
SSRNET	2020	Zhang et al. [157]	Python
TONWMD	2020	Shen et al. [100]	Python
RAFnet	2020	Lu et al. [80]	Python
UAL	2020	Zhang et al. [155]	Python
DBSR	2020	Zhang et al. [154]	Python
NonRegSRNet	2021	Zheng et al. [160]	Python
TSFN	2021	Wang et al. [121]	Python
ADMM-HFNET	2021	Shen et al. [101]	Python
Fusformer	2021	Hu et al. [51]	Python
HSRnet	2021	Hu et al. [52]	HSRnet
MoG-DCN	2021	Dong et al. [33]	Python
HyperFusion	2021	Tian et al. [108]	Python
u^2-MDN	2021	Qu et al. [94]	Python
RGBaux	2022	Li et al. [69]	Python
DHIF	2022	Huang et al. [56]	Python
MIAE	2022	Liu et al. [75]	Python
SpfNet	2022	Liu et al. [74]	Python
UDALN	2022	Li et al. [68]	Python

B.3 Methods without Code Available

Name	Year	Author(s)
MAP	2004	Hardie et al. [49]
Bayesian	2009	Zhang et al. [158]
BayesMonteCarlo	2014	Wei et al. [128]
Hyper-sharpening	2015	Selva et al. [98]
GSE-LowRank	2016	Zhang et al. [151]
3-D-CNN	2017	Palsson et al. [90]
DeepResCNN	2017	Wang et al. [119]
NLSTF	2017	Dian et al. [27]
CollabNMF	2017	Yuan et al. [149]
HSI-DeNet	2018	Chang et al. [21]
HySR-SpaSpeF	2018	Yi et al. [143]
SSCSR	2018	Han et al. [45]
MosaicRGB	2018	Fu et al. [37]
SRIF	2018	Pan and Shen [91]
SSGLRTD	2018	Zhang et al. [152]
NPTSR	2019	Xu et al. [137]
FuVar	2019	Borsoi et al. [15]
MS-SSFNet	2019	Han et al. [47]
DeepEIL	2019	Zhang et al. [156]
LS-MDF	2020	Liu et al. [73]
RLR-MDF	2020	Liu et al. [73]
GDRRN	2020	Wei et al. [133]
HyCoNet	2020	Zheng et al. [159]
NNNLSTF	2020	Wan et al. [118]
HCTR	2020	Xu et al. [138]

Name	Year	Author(s)
SSLRR	2021	Xue et al. [139]
HL-GSNLTD	2021	Peng et al. [92]
STBIM	2021	Gao et al. [39]
ANSR	2021	Li et al. [71]
BDCF	2021	Sun et al. [106]
SSRN	2021	Chen et al. [22]
3DT-Net	2021	Ma et al. [81]
DUFL	2021	Liu et al. [77]
CTRF	2022	He et al. [50]

B.4 Extensions to HIF Methods

Name	Year	Author(s)	Code
TVTVHS	2021	Vella et al. [114]	Python
DeepGrad	2022	Wang et al. [122]	Matlab

C. Code Excerpts

C.1 Script for Automatic Processing

```
import glob
import os
import sys
import time
from pathlib import Path

SCALINGS = [4,8,16]
DATASETS = ["CAVE","EHU","Harvard"]
METHODS = ["CNMF","FUSE","SFIM","GSA","GLP","GSOMP",\
    "NSSR","SupResPALM","CNNFUS","HySure","MAPSMM",\
    "LTTR","LTMR","CSTF","BayesianSparse"]

def get_paths(dataset, method, scale, img):
    hsi_path = f"data/HS/{dataset}/{scale}/{img}.mat"
    msi_path = f"data/MS/{dataset}/{img}.mat"
    gti_path = f"data/GT/{dataset}/{img}.mat"
    sr_path = f"data/SR/{method}/{dataset}/{scale}"
    os.makedirs(sr_path, exist_ok = True)
    sri_path = f"{sr_path}/{img}.mat"
    return hsi_path, msi_path, sri_path, gti_path

def main():
    for cd, dataset in enumerate(DATASETS, start=1):
        img_paths = glob.glob(f'data/GT/{dataset}/*.mat')
        for cm, method in enumerate(METHODS, start=1):
            for cs, scale in enumerate(SCALINGS, start=1):
                for ci, img_path in enumerate(img_paths, start=1):
                    img = Path(img_path).stem
                    hsi_path, msi_path, sri_path, gti_path \
                        get_paths(dataset, method, scale, img)
                    if not os.path.exists(sri_path):
                        run(method,scale,hsi_path,msi_path,sri_path)
```

C.2 Download and Preprocess CAVE Dataset

```
import glob
import os
import sys
from pathlib import Path

import cv2 as cv
import numpy as np
import scipy.io
from PIL import Image

DATASET = 'CAVE'
SCALINGS = [4,8,16]
GT_PATH = f'data/GT/{DATASET}'
MS_PATH = f'data/MS/{DATASET}'
HS_PATH = f'data/HS/{DATASET}'

if not os.path.exists("complete_ms_data.zip"):
    os.system("wget https://www.cs.columbia.edu/CAVE/databases/
    multispectral/zip/complete_ms_data.zip")
os.system("unzip complete_ms_data -d data/GT/aux/")
os.makedirs(GT_PATH, exist_ok = True)
os.system(f"cp -r data/GT/aux/*/* {GT_PATH}/")
os.system("rm -r data/GT/aux/")
#os.system("rm complete_ms_data.zip")

for hs_path in glob.iglob(f"{GT_PATH}/*/"):
    name = Path(hs_path).stem
    hsi = None
    # read all PNGs correspnding to different spectra to form the HS
     cube
    for img_path in glob.iglob(f'{hs_path}/*.png'):
        img = np.asarray(cv.imread(img_path, cv.IMREAD_GRAYSCALE))
        img = np.expand_dims(img, axis=2)
        hsi = img if  hsi is None else np.concatenate((hsi, img),
    axis=2)
    # read BMP with the RGB image
    msi_path = glob.glob(f'{hs_path}/*.bmp')[0]
    msi = np.asarray(cv.cvtColor(cv.imread(msi_path), cv.
    COLOR_BGR2RGB))
    # save both together as MAT file
    scipy.io.savemat(f'{GT_PATH}/{name}.mat', {"hsi": hsi, "msi":
    msi})

os.system(f"rm -r {GT_PATH}/*/")
```

```
os.makedirs(MS_PATH, exist_ok = True)
for sf in SCALINGS:
    os.makedirs(f'{HS_PATH}/{sf}', exist_ok = True)

for mat_path in glob.iglob(f'{GT_PATH}/*.mat'):
    name = Path(mat_path).stem
    mat = scipy.io.loadmat(mat_path)
    msi = mat['msi']
    hsi = mat['hsi']
    # saving RGB
    scipy.io.savemat(f'{MS_PATH}/{name}.mat', {"msi": msi})
    # downsampling HS image
    for sf in SCALINGS:
        hsi_downsampled = None
        for i in range(hsi.shape[2]):
            # from np to Image
            img = Image.fromarray(hsi[:,:,i])
            img = img.resize((hsi.shape[0]//sf, hsi.shape[1]//sf), Image.LANCZOS)
            # from Image to np
            img = np.expand_dims(np.asarray(img), axis=2)
            hsi_downsampled = img if hsi_downsampled is None else np.concatenate((hsi_downsampled, img), axis=2)
        scipy.io.savemat(f'{HS_PATH}/{sf}/{name}.mat', {"hsi": hsi_downsampled})
```

C.3 Download and Preprocess Harvard Dataset

```
import glob
import os
import sys
from pathlib import Path

import numpy as np
import scipy.io
from PIL import Image

DATASET = 'Harvard'
SCALINGS = ([int(sys.argv[1])] if len(sys.argv) >= 2 else [4,8,16])
GT_PATH = f'data/GT/{DATASET}'
MS_PATH = f'data/MS/{DATASET}'
HS_PATH = f'data/HS/{DATASET}'
T = np.array([
    [0.005,0.007,0.012,0.015,0.023,0.025,0.030,0.026,0.024,0.019,\
```

```
    0.010,0.004,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0],
    [0.000,0.000,0.000,0.000,0.000,0.001,0.002,0.003,0.005,0.007,\
    0.012,0.013,0.015,0.016,0.017,0.02,0.013,0.011,0.009,0.005,\
    0.001,0.001,0.001,0.001,0.001,0.001,0.001,0.001,0.002,0.002,\
    0.003],
    [0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,0.000,\
    0.000,0.000,0.000,0.000,0.000,0.000,0.001,0.003,0.010,0.012,\
    0.013,0.022,0.020,0.020,0.018,0.017,0.016,0.016,0.014,0.014,\
    0.013]
])
T[0] = T[0] / T[0].sum() * T.shape[1]
T[1] = T[1] / T[2].sum() * T.shape[1]
T[2] = T[2] / T[2].sum() * T.shape[1]
T = T.T


if not os.path.exists("CZ_hsdbi.tgz"):
    os.system("wget http://vision.seas.harvard.edu/hyperspec/d2x5g3/
    CZ_hsdbi.tgz")
if not os.path.exists("CZ_hsdbi.tgz"):
    os.system("wget http://vision.seas.harvard.edu/hyperspec/d2x5g3/
    CZ_hsdb.tgz")
os.system("tar -xvzf CZ_hsdbi.tgz -C data/GT/")
os.system("tar -xvzf CZ_hsdb.tgz -C data/GT/")
os.makedirs(GT_PATH, exist_ok = True)
os.system(f"mv data/GT/CZ_hsdbi/* {GT_PATH}")
os.system(f"mv data/GT/CZ_hsdb/* {GT_PATH}")
os.system("rm -r data/GT/CZ_hsdbi")
os.system("rm -r data/GT/CZ_hsdb")
#os.system("rm CZ_hsdbi.tgz")
#os.system("rm CZ_hsdb.tgz")
os.makedirs(MS_PATH, exist_ok = True)
for sf in SCALINGS:
    os.makedirs(f'{HS_PATH}/{sf}', exist_ok = True)

for mat_path in glob.iglob(f'{GT_PATH}/*.mat'):
    name = Path(mat_path).stem
    print(name)
    mat = scipy.io.loadmat(mat_path)
    hsi = mat['ref']
    # downsampling HS image
    for sf in SCALINGS:
        hsi_downsampled = None
        for i in range(hsi.shape[2]):
            # from np to Image
            img = Image.fromarray(hsi[:,:,i])
```

```
            img = img.resize((hsi.shape[1]//sf, hsi.shape[0]//sf),
    Image.LANCZOS)
            # from Image to np
            img = np.expand_dims(np.asarray(img), axis=2)
            hsi_downsampled = img if  hsi_downsampled is None else
    np.concatenate((hsi_downsampled, img), axis=2)
        scipy.io.savemat(f'{HS_PATH}/{sf}/{name}.mat', {"hsi":
    hsi_downsampled})
    # simulate RGB photo with Nikon D700 camera
    msi = np.dot(hsi,T)
    scipy.io.savemat(f'{MS_PATH}/{name}.mat', {"msi": msi})
```

C.4 Download and Preprocess EHU Dataset

```
import glob
import math
import os
import sys
from pathlib import Path

import matplotlib.pyplot as plt
import numpy as np
import scipy.io
from PIL import Image

DATASET = 'EHU'
SCALINGS = ([int(sys.argv[1])] if len(sys.argv) >= 2 else [4,8,16])
GT_PATH = f'data/GT/{DATASET}'
MS_PATH = f'data/MS/{DATASET}'
HS_PATH = f'data/HS/{DATASET}'
os.makedirs(MS_PATH, exist_ok = True)
for sf in SCALINGS:
    os.makedirs(f'{HS_PATH}/{sf}', exist_ok = True)

os.makedirs(GT_PATH, exist_ok = True)
os.makedirs(MS_PATH, exist_ok = True)
os.makedirs(HS_PATH, exist_ok = True)
if not os.path.exists(f"{GT_PATH}/Indian_pines.mat"):
    os.system(f"wget http://www.ehu.eus/ccwintco/uploads/2/22/
    Indian_pines.mat -P {GT_PATH}")
)
# download process for the other images omitted for brevity

def expand2fitscaling(img, background_color=0):
```

```
    width, height = img.size
    max_scaling = max(SCALINGS)
    new_width = math.ceil(width/max_scaling) * max_scaling
    new_height = math.ceil(height/max_scaling) * max_scaling
    if new_height == height and new_width == width:
        return img
    else:
        result = Image.new(img.mode, (new_width, new_height), background_color)
        result.paste(img)
        return result

# downsampling HS image
for mat_path in glob.iglob(f'{GT_PATH}/*.mat'):
    name = Path(mat_path).stem
    print(name)
    mat = scipy.io.loadmat(mat_path)
    hsi = mat[list(mat.keys())[-1]]
    print(hsi.shape)
    new_hsi = None
    for i in range(hsi.shape[2]):
        img = Image.fromarray(hsi[:,:,i]) # np to Image
        img = expand2fitscaling(img)
        img = np.expand_dims(np.asarray(img), axis=2) # Image to np
        new_hsi = img if  new_hsi is None else np.concatenate((new_hsi, img), axis=2)
    hsi = new_hsi
    scipy.io.savemat(f'{GT_PATH}/{name}.mat', {list(mat.keys())[-1]: new_hsi})
    print(hsi.shape)
    hsi = hsi.astype("float32") / hsi.max()
    for sf in SCALINGS:
        hsi_downsampled = None
        for i in range(hsi.shape[2]):
            img = Image.fromarray(hsi[:,:,i]) # np to Image
            img = img.resize((hsi.shape[1]//sf, hsi.shape[0]//sf), Image.LANCZOS)
            img = np.expand_dims(np.asarray(img), axis=2) # Image to np
            hsi_downsampled = img if  hsi_downsampled is None else np.concatenate((hsi_downsampled, img), axis=2)
        scipy.io.savemat(f'{HS_PATH}/{sf}/{name}.mat', {"hsi": hsi_downsampled})
```

D. Color Patches Comparison

This appendix compares the error in the spectra of HIF methods according to different color patches and scaling factors. Figure D.1 presents the position of each color patch in the extended eSFR ISO 12233:2017 chart. Figures D.2 to D.5 present (from the left to the right): a color composite of the patch simulated from the HS cube; the mean ground truth of the spectra for that color patch; and, the error for scaling factors of 4x, 8x and 16x for the different HIF methods. figs. D.6 to D.8 present the JND of the reconstructed color patch according to its GT reference spectrum, for each upsampling factor, HIF method, and color patch.

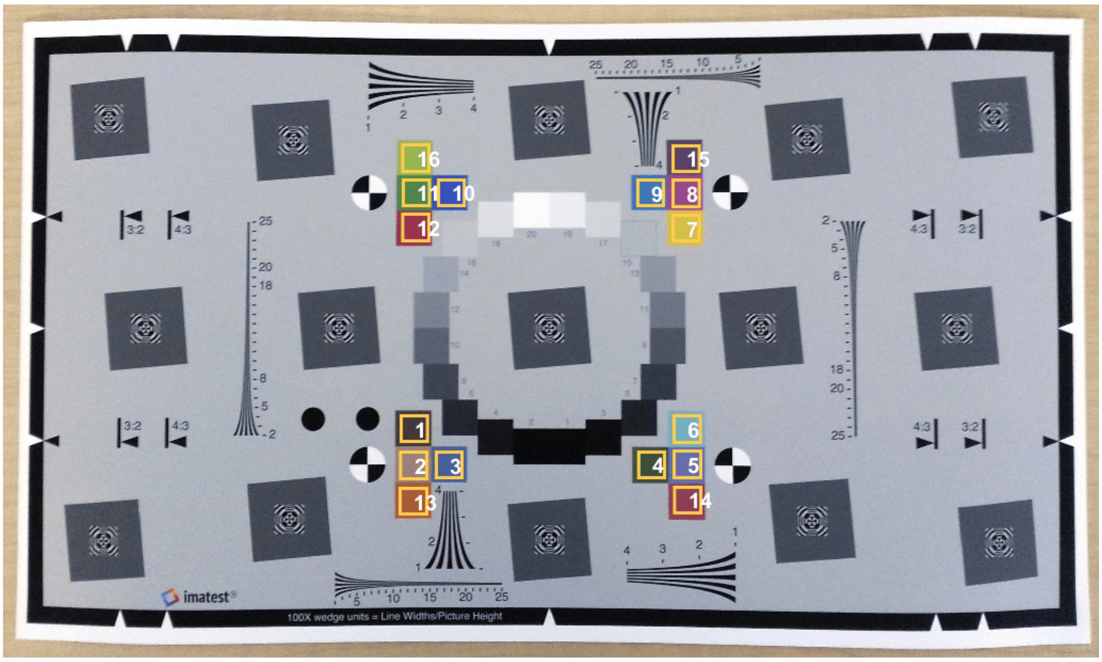


Figure D.1: Position of the color patches in the extended eSFR ISO 12233:2017 chart.

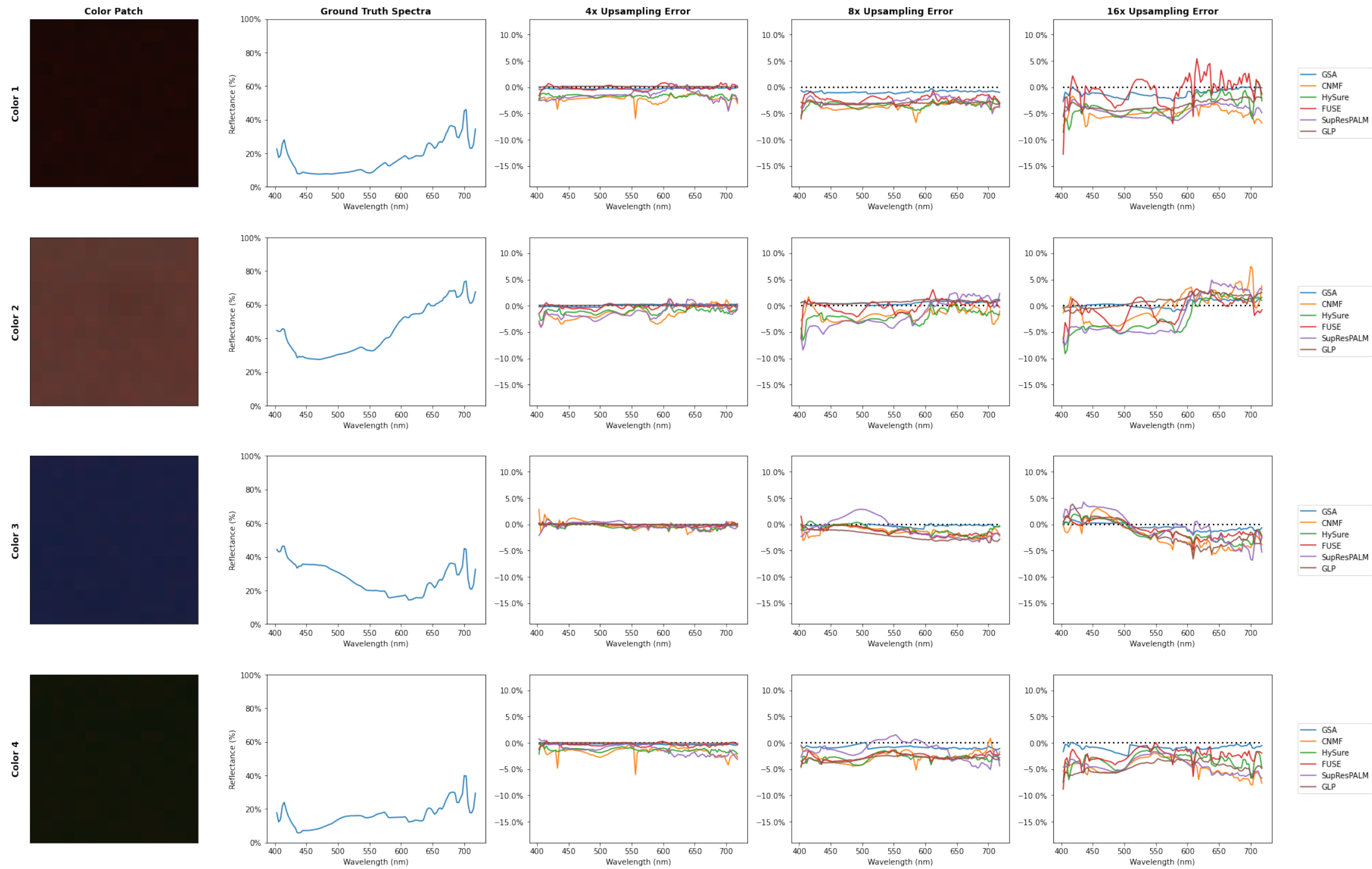


Figure D.2: Error comparison of color patches (1/4).

Figure D.3: Error comparison of color patches (2/4).

Figure D.4: Error comparison of color patches (3/4).

Figure D.5: Error comparison of color patches (4/4).

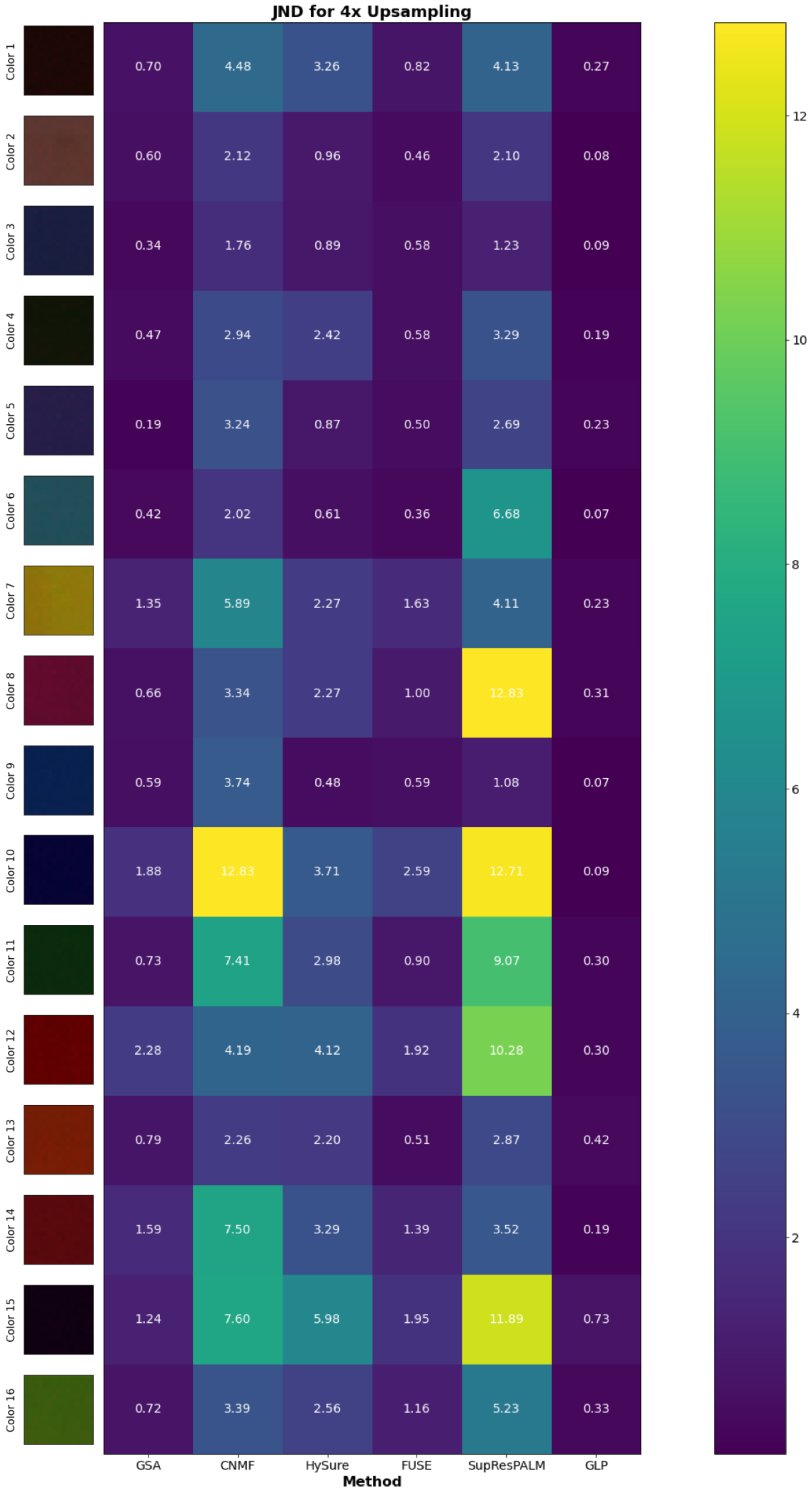


Figure D.6: JND for 4x Upsampling.

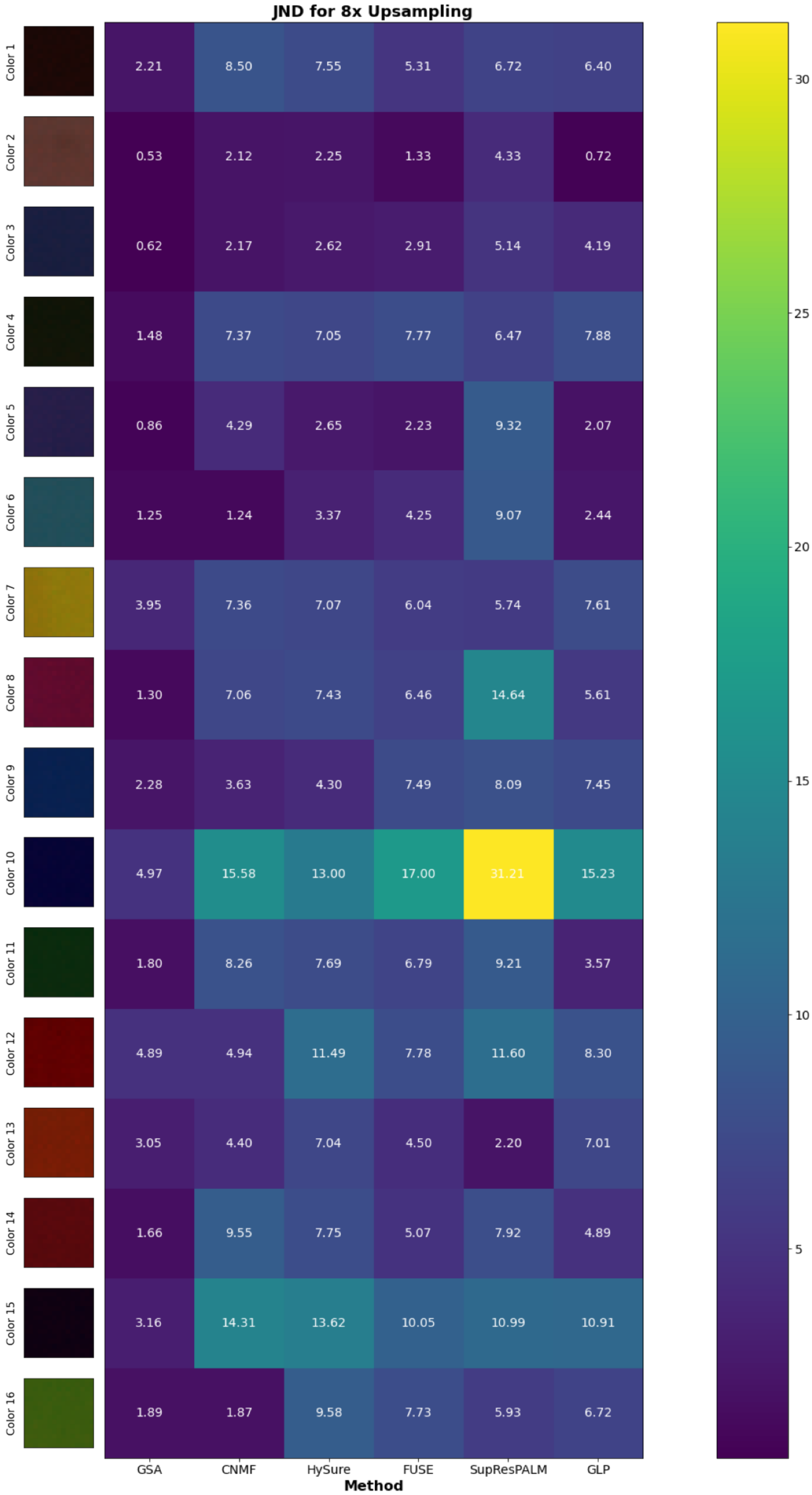


Figure D.7: JND for 8x Upsampling.

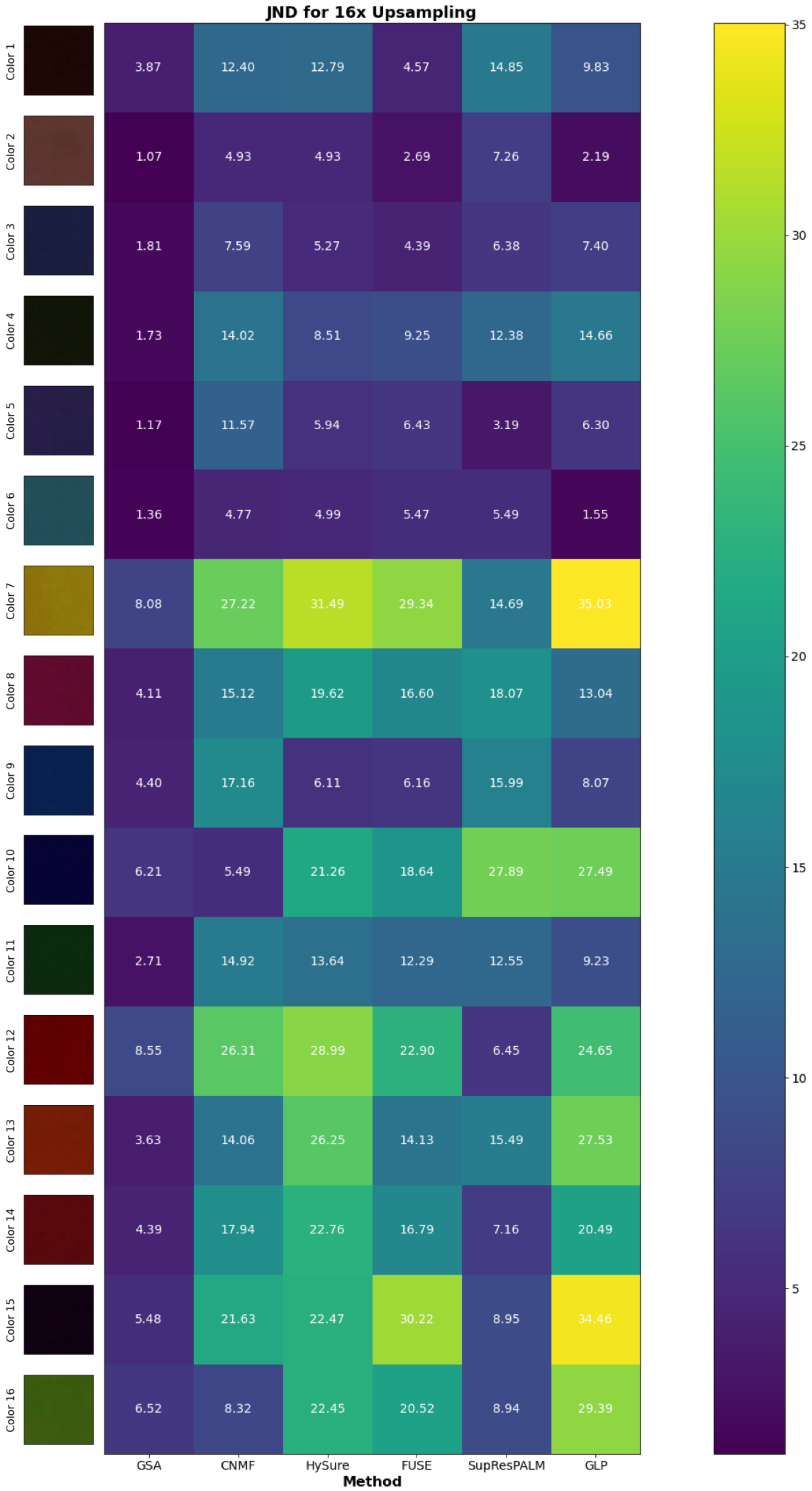


Figure D.8: JND for 16x Upsampling.

E. Numerical Results of Selected HIF Methods

E.1 GLP

E.1.1 CAVE Dataset

Table E.1: Results of GLP with CAVE dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
balloons	0.818	0.275	0.190
chart_and_stuffed_toy	0.797	0.477	0.153
pompoms	0.851	0.740	0.146
superballs	0.893	0.660	0.197
clay	0.875	0.436	0.157
cd	0.869	0.576	0.192
fake_and_real_tomatoes	0.905	0.351	0.264
fake_and_real_strawberries	0.854	0.523	0.164
sponges	0.795	0.432	0.109
real_and_fake_apples	0.873	0.277	0.187
hairs	0.904	0.606	0.158
paints	0.826	0.527	0.134
stuffed_toys	0.837	0.397	0.175
beads	0.874	0.734	0.208
fake_and_real_beers	0.933	0.524	0.082
fake_and_real_lemons	0.861	0.412	0.153
thread_spools	0.890	0.658	0.183
glass_tiles	0.870	0.751	0.189
fake_and_real_lemon_slices	0.888	0.680	0.208
jelly_beans	0.858	0.683	0.199
watercolors	0.894	0.734	0.082

Table E.1: Results of GLP with CAVE dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
real_and_fake_peppers	0.855	0.350	0.154
photo_and_face	0.912	0.313	0.168
face	0.866	0.398	0.181
flowers	0.886	0.406	0.189
oil_painting	0.884	0.757	0.149
fake_and_real_food	0.852	0.470	0.215
egyptian_statue	0.927	0.249	0.199
fake_and_real_sushi	0.884	0.475	0.237
feathers	0.816	0.545	0.169
fake_and_real_peppers	0.853	0.306	0.209
cloth	0.935	0.855	0.130

Table E.2: Results of GLP with CAVE dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
balloons	0.826	0.301	0.189
chart_and_stuffed_toy	0.792	0.488	0.157
pompoms	0.851	0.745	0.145
superballs	0.891	0.667	0.207
clay	0.874	0.449	0.164
cd	0.867	0.575	0.204
fake_and_real_tomatoes	0.921	0.334	0.277
fake_and_real_strawberries	0.853	0.538	0.165
sponges	0.804	0.460	0.108
real_and_fake_apples	0.870	0.291	0.188
hairs	0.903	0.612	0.164
paints	0.827	0.545	0.137
stuffed_toys	0.836	0.413	0.177
beads	0.860	0.739	0.228
fake_and_real_beers	0.939	0.543	0.082
fake_and_real_lemons	0.860	0.431	0.153
thread_spools	0.889	0.662	0.187
glass_tiles	0.870	0.759	0.202
fake_and_real_lemon_slices	0.885	0.684	0.213
jelly_beans	0.852	0.689	0.203
watercolors	0.892	0.745	0.085
real_and_fake_peppers	0.853	0.371	0.155
photo_and_face	0.902	0.317	0.185

Table E.2: Results of GLP with CAVE dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
face	0.866	0.403	0.183
flowers	0.885	0.419	0.193
oil_painting	0.882	0.760	0.152
fake_and_real_food	0.848	0.477	0.218
egyptian_statue	0.926	0.258	0.203
fake_and_real_sushi	0.883	0.479	0.243
feathers	0.812	0.554	0.174
fake_and_real_peppers	0.855	0.324	0.215
cloth	0.927	0.855	0.137

Table E.3: Results of GLP with CAVE dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
balloons	0.832	0.305	0.189
chart_and_stuffed_toy	0.795	0.498	0.157
pompoms	0.846	0.737	0.147
superballs	0.890	0.669	0.219
clay	0.868	0.453	0.180
cd	0.863	0.576	0.232
fake_and_real_tomatoes	0.922	0.336	0.290
fake_and_real_strawberries	0.851	0.540	0.167
sponges	0.822	0.463	0.110
real_and_fake_apples	0.871	0.293	0.192
hairs	0.900	0.610	0.175
paints	0.819	0.545	0.144
stuffed_toys	0.835	0.414	0.181
beads	0.842	0.742	0.261
fake_and_real_beers	0.944	0.544	0.081
fake_and_real_lemons	0.860	0.431	0.159
thread_spools	0.888	0.664	0.192
glass_tiles	0.868	0.766	0.218
fake_and_real_lemon_slices	0.887	0.689	0.221
jelly_beans	0.850	0.691	0.215
watercolors	0.899	0.747	0.088
real_and_fake_peppers	0.853	0.373	0.161
photo_and_face	0.897	0.319	0.195
face	0.867	0.405	0.187
flowers	0.880	0.419	0.193

Table E.3: Results of GLP with CAVE dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
oil_painting	0.879	0.755	0.157
fake_and_real_food	0.842	0.475	0.221
egyptian_statue	0.924	0.261	0.215
fake_and_real_sushi	0.884	0.483	0.251
feathers	0.808	0.557	0.183
fake_and_real_peppers	0.852	0.328	0.222
cloth	0.921	0.855	0.147

E.1.2 Harvard Dataset

Table E.4: Results of GLP with Harvard dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgc4	0.987	0.580	0.048
imgb0	0.865	0.732	0.047
imgb1	0.980	0.725	0.035
imgc5	0.985	0.429	0.033
imgb3	0.810	0.611	0.058
imgc7	0.978	0.684	0.053
imgc6	0.983	0.434	0.038
imgb2	0.932	0.689	0.061
imgb6	0.979	0.869	0.061
imgc2	0.652	0.529	0.038
imgc3	0.912	0.413	0.032
imgb7	0.903	0.566	0.066
imgc1	0.980	0.480	0.043
imgb5	0.985	0.598	0.046
imgb4	0.946	0.837	0.070
imga8	0.934	0.432	0.082
imgh0	0.979	0.504	0.048
imge7	0.941	0.819	0.038
imgd3	0.989	0.455	0.051
imgd2	0.956	0.496	0.059
imge6	0.988	0.818	0.047
imgh1	0.981	0.401	0.030
imgh3	0.953	0.644	0.053
imgd0	0.967	0.408	0.040
imge4	0.981	0.566	0.029

Table E.4: Results of GLP with Harvard dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgf8	0.986	0.707	0.095
imge5	0.974	0.703	0.031
imgd1	0.972	0.344	0.055
imgh2	0.990	0.470	0.072
imgh6	0.984	0.265	0.084
imgg9	0.996	0.317	0.078
imgd5	0.883	0.379	0.048
imge1	0.943	0.545	0.082
imge0	0.959	0.690	0.056
imgd4	0.971	0.408	0.029
imgg8	0.994	0.265	0.077
imgh7	0.994	0.423	0.081
imgh5	0.974	0.336	0.068
imge2	0.951	0.728	0.154
imgd6	0.995	0.454	0.076
imgd7	0.988	0.639	0.044
imge3	0.982	0.770	0.057
imgh4	0.893	0.375	0.056
imgg6	0.902	0.305	0.095
imgf2	0.954	0.799	0.073
imgf3	0.877	0.680	0.074
imgg7	0.986	0.428	0.052
imgf1	0.808	0.467	0.052
imgg5	0.993	0.437	0.070
imgd9	0.956	0.418	0.035
imgd8	0.985	0.456	0.045
imgg4	0.991	0.367	0.105
imgf4	0.974	0.579	0.024
imgg0	0.928	0.496	0.075
imgg1	0.988	0.350	0.092
imgf5	0.972	0.717	0.070
imgg3	0.961	0.406	0.056
imgf7	0.900	0.740	0.054
imgf6	0.838	0.677	0.090
imgg2	0.987	0.337	0.070
img3	0.986	0.333	0.057
imga5	0.966	0.358	0.050
imgb9	0.976	0.574	0.036

Table E.4: Results of GLP with Harvard dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgb8	0.914	0.755	0.059
imga4	0.910	0.462	0.071
img2	0.987	0.793	0.059
imga6	0.980	0.693	0.042
imga7	0.984	0.755	0.029
img1	0.977	0.674	0.030
img5	0.993	0.346	0.058
imga3	0.880	0.605	0.063
imga2	0.990	0.397	0.040
img4	0.981	0.353	0.077
img6	0.991	0.338	0.061
imgc8	0.974	0.648	0.041
imga1	0.987	0.699	0.042
imgc9	0.974	0.797	0.051

Table E.5: Results of GLP with Harvard dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgc4	0.980	0.580	0.057
imgb0	0.828	0.729	0.049
imgb1	0.980	0.727	0.036
imgc5	0.983	0.427	0.037
imgb3	0.767	0.604	0.068
imgc7	0.980	0.680	0.054
imgc6	0.982	0.433	0.043
imgb2	0.894	0.688	0.070
imgb6	0.971	0.869	0.061
imgc2	0.640	0.527	0.038
imgc3	0.928	0.413	0.032
imgb7	0.823	0.550	0.086
imgc1	0.976	0.473	0.048
imgb5	0.984	0.598	0.050
imgb4	0.787	0.831	0.078
imga8	0.921	0.429	0.095
imgh0	0.967	0.502	0.053
imge7	0.942	0.817	0.040
imgd3	0.984	0.452	0.059
imgd2	0.950	0.492	0.071

Table E.5: Results of GLP with Harvard dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imge6	0.986	0.818	0.051
imgh1	0.988	0.401	0.035
imgh3	0.963	0.644	0.054
imgd0	0.947	0.406	0.047
imge4	0.980	0.565	0.033
imgf8	0.983	0.706	0.096
imge5	0.973	0.702	0.033
imgd1	0.950	0.342	0.063
imgh2	0.980	0.469	0.084
imgh6	0.983	0.261	0.087
imgg9	0.994	0.314	0.084
imgd5	0.907	0.377	0.054
imge1	0.920	0.540	0.102
imge0	0.963	0.688	0.061
imgd4	0.977	0.407	0.033
imgg8	0.991	0.262	0.084
imgh7	0.994	0.419	0.084
imgh5	0.969	0.334	0.073
imge2	0.940	0.724	0.178
imgd6	0.995	0.452	0.083
imgd7	0.986	0.639	0.050
imge3	0.976	0.771	0.063
imgh4	0.892	0.373	0.062
imgg6	0.887	0.297	0.098
imgf2	0.938	0.799	0.078
imgf3	0.840	0.679	0.089
imgg7	0.984	0.427	0.059
imgf1	0.737	0.459	0.069
imgg5	0.993	0.435	0.073
imgd9	0.942	0.417	0.038
imgd8	0.979	0.453	0.053
imgg4	0.989	0.365	0.111
imgf4	0.966	0.574	0.029
imgg0	0.927	0.495	0.079
imgg1	0.987	0.347	0.096
imgf5	0.961	0.717	0.077
imgg3	0.960	0.404	0.059
imgf7	0.831	0.737	0.059

Table E.5: Results of GLP with Harvard dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgf6	0.831	0.675	0.094
imgg2	0.987	0.332	0.077
img3	0.990	0.331	0.063
imga5	0.947	0.346	0.056
imgb9	0.976	0.573	0.039
imgb8	0.901	0.752	0.064
imga4	0.867	0.459	0.082
img2	0.949	0.794	0.064
imga6	0.978	0.688	0.044
imga7	0.984	0.755	0.029
img1	0.974	0.674	0.036
img5	0.992	0.344	0.062
imga3	0.828	0.603	0.084
imga2	0.990	0.382	0.050
img4	0.979	0.350	0.081
img6	0.989	0.335	0.065
imgc8	0.968	0.648	0.047
imga1	0.985	0.700	0.045
imgc9	0.973	0.797	0.054

Table E.6: Results of GLP with Harvard dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgc4	0.973	0.999	0.063
imgb0	0.799	1.000	0.051
imgb1	0.982	1.000	0.037
imgc5	0.980	1.000	0.039
imgb3	0.797	0.999	0.073
imgc7	0.980	1.000	0.053
imgc6	0.982	1.000	0.047
imgb2	0.872	0.999	0.078
imgb6	0.959	0.999	0.058
imgc2	0.643	1.000	0.038
imgc3	0.931	1.000	0.033
imgb7	0.705	0.996	0.101
imgc1	0.967	1.000	0.056
imgb5	0.984	1.000	0.053
imgb4	0.703	0.998	0.079

Table E.6: Results of GLP with Harvard dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imga8	0.922	1.000	0.102
imgh0	0.967	1.000	0.051
imge7	0.939	0.999	0.041
imgd3	0.973	1.000	0.070
imgd2	0.934	0.999	0.075
imge6	0.986	0.999	0.051
imgh1	0.960	1.000	0.040
imgh3	0.967	1.000	0.055
imgd0	0.936	0.999	0.053
imge4	0.979	1.000	0.034
imgf8	0.984	1.000	0.091
imge5	0.973	1.000	0.036
imgd1	0.954	1.000	0.070
imgh2	0.976	1.000	0.083
imgh6	0.982	1.000	0.088
imgg9	0.993	1.000	0.089
imgd5	0.935	1.000	0.059
imge1	0.917	0.998	0.111
imge0	0.972	1.000	0.064
imgd4	0.979	1.000	0.036
imgg8	0.991	1.000	0.086
imgh7	0.995	1.000	0.083
imgh5	0.965	1.000	0.079
imge2	0.939	0.998	0.185
imgd6	0.994	1.000	0.091
imgd7	0.983	0.999	0.055
imge3	0.975	0.999	0.065
imgh4	0.873	1.000	0.067
imgg6	0.858	1.000	0.099
imgf2	0.917	0.999	0.080
imgf3	0.812	0.999	0.093
imgg7	0.981	1.000	0.065
imgf1	0.727	0.999	0.090
imgg5	0.991	1.000	0.077
imgd9	0.941	1.000	0.041
imgd8	0.972	1.000	0.058
imgg4	0.988	1.000	0.117
imgf4	0.955	0.999	0.036

Table E.6: Results of GLP with Harvard dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgg0	0.923	1.000	0.083
imgg1	0.984	1.000	0.101
imgf5	0.952	1.000	0.080
imgg3	0.958	1.000	0.062
imgf7	0.766	0.999	0.064
imgf6	0.822	1.000	0.097
imgg2	0.985	1.000	0.082
img3	0.988	1.000	0.068
imga5	0.938	1.000	0.054
imgb9	0.977	1.000	0.041
imgb8	0.884	0.999	0.068
imga4	0.839	1.000	0.094
img2	0.935	0.999	0.066
imga6	0.977	1.000	0.046
imga7	0.985	1.000	0.029
img1	0.972	0.999	0.043
img5	0.992	1.000	0.065
imga3	0.757	0.999	0.108
imga2	0.986	1.000	0.064
img4	0.979	1.000	0.084
img6	0.985	1.000	0.068
imgc8	0.953	0.999	0.052
imga1	0.986	0.999	0.045
imgc9	0.973	1.000	0.055

E.1.3 EHU Dataset

Table E.7: Results of GLP with EHU dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
KSC	0.957	0.267	0.700
Pavia	0.883	0.633	0.171
Botswana	0.924	0.380	0.152
PaviaU	0.869	0.622	0.156
SalinasA	0.921	0.430	0.204
Indian_pines	0.726	0.272	0.115
Salinas	0.876	0.246	0.176
Cuprite	0.868	0.359	-

Table E.8: Results of GLP with EHU dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
KSC	0.946	0.263	0.774
Pavia	0.860	0.629	0.197
Botswana	0.912	0.328	0.183
PaviaU	0.833	0.620	0.185
SalinasA	0.875	0.428	0.241
Indian_pines	0.655	0.305	0.149
Salinas	0.842	0.242	0.202
Cuprite	0.818	0.347	-

Table E.9: Results of GLP with EHU dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
KSC	0.923	0.266	0.840
Pavia	0.842	0.618	0.220
Botswana	0.905	0.299	0.201
PaviaU	0.812	0.602	0.207
SalinasA	0.836	0.379	0.284
Indian_pines	0.629	0.352	0.172
Salinas	0.815	0.267	0.235
Cuprite	0.806	0.405	-

E.2 GSA

E.2.1 CAVE Dataset

Table E.10: Results of GSA with CAVE dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
balloons	0.818	0.275	0.190
chart_and_stuffed_toy	0.797	0.477	0.153
pompoms	0.851	0.740	0.146
superballs	0.893	0.660	0.197
clay	0.875	0.436	0.157
cd	0.869	0.576	0.192
fake_and_real_tomatoes	0.905	0.351	0.264
fake_and_real_strawberries	0.854	0.523	0.164
sponges	0.795	0.432	0.109

Table E.10: Results of GSA with CAVE dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
real_and_fake_apples	0.873	0.277	0.187
hairs	0.904	0.606	0.158
paints	0.826	0.527	0.134
stuffed_toys	0.837	0.397	0.175
beads	0.874	0.734	0.208
fake_and_real_beers	0.933	0.524	0.082
fake_and_real_lemons	0.861	0.412	0.153
thread_spools	0.890	0.658	0.183
glass_tiles	0.870	0.751	0.189
fake_and_real_lemon_slices	0.888	0.680	0.208
jelly_beans	0.858	0.683	0.199
watercolors	0.894	0.734	0.082
real_and_fake_peppers	0.855	0.350	0.154
photo_and_face	0.912	0.313	0.168
face	0.866	0.398	0.181
flowers	0.886	0.406	0.189
oil_painting	0.884	0.757	0.149
fake_and_real_food	0.852	0.470	0.215
egyptian_statue	0.927	0.249	0.199
fake_and_real_sushi	0.884	0.475	0.237
feathers	0.816	0.545	0.169
fake_and_real_peppers	0.853	0.306	0.209
cloth	0.935	0.855	0.130

Table E.11: Results of GSA with CAVE dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
balloons	0.826	0.301	0.189
chart_and_stuffed_toy	0.792	0.488	0.157
pompoms	0.851	0.745	0.145
superballs	0.891	0.667	0.207
clay	0.874	0.449	0.164
cd	0.867	0.575	0.204
fake_and_real_tomatoes	0.921	0.334	0.277
fake_and_real_strawberries	0.853	0.538	0.165
sponges	0.804	0.460	0.108
real_and_fake_apples	0.870	0.291	0.188
hairs	0.903	0.612	0.164

Table E.11: Results of GSA with CAVE dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
paints	0.827	0.545	0.137
stuffed_toys	0.836	0.413	0.177
beads	0.860	0.739	0.228
fake_and_real_beers	0.939	0.543	0.082
fake_and_real_lemons	0.860	0.431	0.153
thread_spools	0.889	0.662	0.187
glass_tiles	0.870	0.759	0.202
fake_and_real_lemon_slices	0.885	0.684	0.213
jelly_beans	0.852	0.689	0.203
watercolors	0.892	0.745	0.085
real_and_fake_peppers	0.853	0.371	0.155
photo_and_face	0.902	0.317	0.185
face	0.866	0.403	0.183
flowers	0.885	0.419	0.193
oil_painting	0.882	0.760	0.152
fake_and_real_food	0.848	0.477	0.218
egyptian_statue	0.926	0.258	0.203
fake_and_real_sushi	0.883	0.479	0.243
feathers	0.812	0.554	0.174
fake_and_real_peppers	0.855	0.324	0.215
cloth	0.927	0.855	0.137

Table E.12: Results of GSA with CAVE dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
balloons	0.832	0.305	0.189
chart_and_stuffed_toy	0.795	0.498	0.157
pompoms	0.846	0.737	0.147
superballs	0.890	0.669	0.219
clay	0.868	0.453	0.180
cd	0.863	0.576	0.232
fake_and_real_tomatoes	0.922	0.336	0.290
fake_and_real_strawberries	0.851	0.540	0.167
sponges	0.822	0.463	0.110
real_and_fake_apples	0.871	0.293	0.192
hairs	0.900	0.610	0.175
paints	0.819	0.545	0.144
stuffed_toys	0.835	0.414	0.181

Table E.12: Results of GSA with CAVE dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
beads	0.842	0.742	0.261
fake_and_real_beers	0.944	0.544	0.081
fake_and_real_lemons	0.860	0.431	0.159
thread_spools	0.888	0.664	0.192
glass_tiles	0.868	0.766	0.218
fake_and_real_lemon_slices	0.887	0.689	0.221
jelly_beans	0.850	0.691	0.215
watercolors	0.899	0.747	0.088
real_and_fake_peppers	0.853	0.373	0.161
photo_and_face	0.897	0.319	0.195
face	0.867	0.405	0.187
flowers	0.880	0.419	0.193
oil_painting	0.879	0.755	0.157
fake_and_real_food	0.842	0.475	0.221
egyptian_statue	0.924	0.261	0.215
fake_and_real_sushi	0.884	0.483	0.251
feathers	0.808	0.557	0.183
fake_and_real_peppers	0.852	0.328	0.222
cloth	0.921	0.855	0.147

E.2.2 Harvard Dataset

Table E.13: Results of GSA with Harvard dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgc4	0.992	0.579	0.032
imgb0	0.888	0.739	0.033
imgb1	0.985	0.720	0.025
imgc5	0.987	0.432	0.027
imgb3	0.958	0.621	0.033
imgc7	0.988	0.690	0.034
imgc6	0.986	0.436	0.033
imgb2	0.841	0.688	0.043
imgb6	0.991	0.881	0.039
imgc2	0.907	0.551	0.033
imgc3	0.867	0.416	0.026
imgb7	0.978	0.600	0.046
imgc1	0.982	0.476	0.037

Table E.13: Results of GSA with Harvard dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgb5	0.990	0.598	0.032
imgb4	0.932	0.853	0.051
imga8	0.931	0.449	0.074
imgh0	0.980	0.493	0.036
imge7	0.872	0.818	0.028
imgd3	0.992	0.451	0.035
imgd2	0.887	0.490	0.038
imge6	0.990	0.806	0.036
imgh1	0.992	0.403	0.024
imgh3	0.937	0.639	0.049
imgd0	0.970	0.413	0.034
imge4	0.982	0.570	0.024
imgf8	0.991	0.690	0.074
imge5	0.978	0.694	0.026
imgd1	0.991	0.346	0.048
imgh2	0.995	0.462	0.049
imgh6	0.990	0.289	0.075
imgg9	0.994	0.312	0.075
imgd5	0.929	0.444	0.036
imge1	0.944	0.551	0.079
imge0	0.920	0.684	0.045
imgd4	0.988	0.405	0.026
imgg8	0.997	0.304	0.070
imgh7	0.996	0.429	0.079
imgh5	0.981	0.382	0.067
imge2	0.961	0.726	0.144
imgd6	0.992	0.468	0.069
imgd7	0.991	0.637	0.032
imge3	0.987	0.763	0.040
imgh4	0.921	0.380	0.050
imgg6	0.539	0.271	0.098
imgf2	0.918	0.797	0.054
imgf3	0.861	0.699	0.058
imgg7	0.987	0.447	0.049
imgf1	0.946	0.479	0.036
imgg5	0.996	0.432	0.065
imgd9	0.951	0.437	0.031
imgd8	0.989	0.461	0.035

Table E.13: Results of GSA with Harvard dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgg4	0.994	0.367	0.099
imgf4	0.984	0.578	0.018
imgg0	0.952	0.495	0.065
imgg1	0.993	0.422	0.079
imgf5	0.942	0.707	0.046
imgg3	0.977	0.402	0.053
imgf7	0.974	0.740	0.040
imgf6	0.829	0.682	0.073
imgg2	0.996	0.328	0.064
img3	0.994	0.322	0.053
imga5	0.971	0.370	0.041
imgb9	0.981	0.571	0.031
imgb8	0.861	0.766	0.035
imga4	0.900	0.475	0.062
img2	0.968	0.787	0.040
imga6	0.989	0.708	0.032
imga7	0.986	0.749	0.026
img1	0.982	0.677	0.022
img5	0.994	0.331	0.055
imga3	0.838	0.608	0.067
imga2	0.996	0.405	0.033
img4	0.992	0.356	0.072
img6	0.991	0.330	0.058
imgc8	0.977	0.643	0.032
imga1	0.992	0.697	0.029
imgc9	0.977	0.786	0.043

Table E.14: Results of GSA with Harvard dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgc4	0.991	0.578	0.035
imgb0	0.877	0.737	0.036
imgb1	0.984	0.720	0.026
imgc5	0.987	0.426	0.029
imgb3	0.904	0.620	0.040
imgc7	0.986	0.690	0.036
imgc6	0.986	0.435	0.034
imgb2	0.761	0.683	0.055

Table E.14: Results of GSA with Harvard dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgb6	0.986	0.880	0.045
imgc2	0.892	0.550	0.035
imgc3	0.854	0.415	0.027
imgb7	0.960	0.592	0.061
imgc1	0.980	0.468	0.039
imgb5	0.989	0.597	0.033
imgb4	0.900	0.853	0.063
imga8	0.906	0.448	0.089
imgh0	0.966	0.492	0.040
imge7	0.915	0.817	0.030
imgd3	0.991	0.449	0.038
imgd2	0.876	0.487	0.043
imge6	0.987	0.806	0.039
imgh1	0.950	0.402	0.025
imgh3	0.927	0.639	0.051
imgd0	0.952	0.411	0.039
imge4	0.982	0.566	0.026
imgf8	0.987	0.689	0.084
imge5	0.975	0.693	0.028
imgd1	0.977	0.345	0.050
imgh2	0.992	0.461	0.060
imgh6	0.990	0.287	0.077
imgg9	0.994	0.310	0.078
imgd5	0.927	0.442	0.040
imge1	0.934	0.547	0.097
imge0	0.901	0.683	0.049
imgd4	0.983	0.402	0.027
imgg8	0.996	0.306	0.073
imgh7	0.996	0.427	0.081
imgh5	0.977	0.381	0.070
imge2	0.946	0.723	0.169
imgd6	0.990	0.467	0.074
imgd7	0.990	0.636	0.035
imge3	0.985	0.763	0.044
imgh4	0.897	0.379	0.054
imgg6	0.527	0.269	0.102
imgf2	0.915	0.796	0.061
imgf3	0.841	0.698	0.065

Table E.14: Results of GSA with Harvard dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgg7	0.985	0.445	0.053
imgf1	0.915	0.478	0.048
imgg5	0.996	0.430	0.068
imgd9	0.949	0.436	0.032
imgd8	0.987	0.460	0.038
imgg4	0.994	0.365	0.101
imgf4	0.982	0.577	0.021
imgg0	0.949	0.494	0.069
imgg1	0.991	0.420	0.082
imgf5	0.919	0.707	0.052
imgg3	0.974	0.400	0.055
imgf7	0.978	0.740	0.044
imgf6	0.815	0.681	0.079
imgg2	0.995	0.328	0.067
img3	0.993	0.320	0.059
imga5	0.938	0.368	0.044
imgb9	0.980	0.570	0.032
imgb8	0.846	0.764	0.041
imga4	0.855	0.474	0.077
img2	0.920	0.788	0.042
imga6	0.987	0.707	0.034
imga7	0.985	0.748	0.027
img1	0.981	0.676	0.023
img5	0.994	0.330	0.056
imga3	0.806	0.608	0.099
imga2	0.995	0.406	0.034
img4	0.991	0.354	0.075
img6	0.991	0.328	0.059
imgc8	0.966	0.642	0.038
imga1	0.989	0.696	0.033
imgc9	0.970	0.785	0.048

Table E.15: Results of GSA with Harvard dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgc4	0.990	0.575	0.039
imgb0	0.871	0.737	0.038
imgb1	0.983	0.720	0.027

Table E.15: Results of GSA with Harvard dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgc5	0.986	0.418	0.029
imgb3	0.850	0.620	0.047
imgc7	0.986	0.690	0.037
imgc6	0.985	0.435	0.035
imgb2	0.724	0.683	0.062
imgb6	0.979	0.880	0.049
imgc2	0.889	0.550	0.035
imgc3	0.843	0.415	0.028
imgb7	0.942	0.592	0.071
imgc1	0.978	0.460	0.042
imgb5	0.989	0.593	0.034
imgb4	0.896	0.849	0.071
imga8	0.892	0.448	0.101
imgh0	0.954	0.492	0.042
imge7	0.913	0.817	0.031
imgd3	0.990	0.449	0.042
imgd2	0.836	0.479	0.050
imge6	0.986	0.806	0.042
imgh1	0.890	0.402	0.026
imgh3	0.922	0.639	0.053
imgd0	0.945	0.411	0.045
imge4	0.980	0.566	0.027
imgf8	0.986	0.689	0.090
imge5	0.974	0.693	0.029
imgd1	0.971	0.345	0.053
imgh2	0.989	0.461	0.070
imgh6	0.990	0.280	0.078
imgg9	0.994	0.310	0.080
imgd5	0.901	0.442	0.046
imge1	0.928	0.547	0.106
imge0	0.892	0.683	0.052
imgd4	0.980	0.402	0.027
imgg8	0.995	0.306	0.077
imgh7	0.996	0.426	0.081
imgh5	0.975	0.387	0.073
imge2	0.941	0.722	0.179
imgd6	0.989	0.477	0.080
imgd7	0.988	0.636	0.037

Table E.15: Results of GSA with Harvard dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imge3	0.984	0.763	0.046
imgh4	0.891	0.379	0.057
imgg6	0.524	0.269	0.103
imgf2	0.910	0.796	0.067
imgf3	0.834	0.698	0.071
imgg7	0.983	0.445	0.059
imgf1	0.880	0.478	0.066
imgg5	0.995	0.430	0.069
imgd9	0.948	0.436	0.033
imgd8	0.986	0.460	0.041
imgg4	0.994	0.365	0.103
imgf4	0.972	0.580	0.025
imgg0	0.947	0.494	0.072
imgg1	0.990	0.420	0.085
imgf5	0.916	0.707	0.058
imgg3	0.972	0.400	0.056
imgf7	0.971	0.740	0.047
imgf6	0.807	0.681	0.085
imgg2	0.994	0.328	0.071
img3	0.962	0.320	0.068
imga5	0.950	0.373	0.045
imgb9	0.979	0.570	0.033
imgb8	0.873	0.765	0.047
imga4	0.825	0.461	0.097
img2	0.921	0.787	0.044
imga6	0.985	0.706	0.036
imga7	0.985	0.748	0.027
img1	0.980	0.676	0.024
img5	0.994	0.329	0.057
imga3	0.638	0.602	0.132
imga2	0.994	0.397	0.037
img4	0.991	0.354	0.077
img6	0.990	0.321	0.060
imgc8	0.960	0.642	0.043
imga1	0.987	0.696	0.037
imgc9	0.967	0.785	0.051

E.2.3 EHU Dataset

Table E.16: Results of GSA with EHU dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
KSC	0.960	0.226	0.726
Pavia	0.881	0.651	0.188
Botswana	0.933	0.701	0.122
PaviaU	0.764	0.631	0.179
SalinasA	0.927	0.461	0.212
Indian_pines	0.765	0.339	0.101
Salinas	0.955	0.420	0.173
Cuprite	0.898	0.633	-

Table E.17: Results of GSA with EHU dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
KSC	0.952	0.222	0.794
Pavia	0.815	0.644	0.206
Botswana	0.928	0.700	0.132
PaviaU	0.694	0.619	0.200
SalinasA	0.895	0.453	0.229
Indian_pines	0.682	0.339	0.129
Salinas	0.939	0.410	0.184
Cuprite	0.857	0.626	-

Table E.18: Results of GSA with EHU dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
KSC	0.937	0.246	0.850
Pavia	0.723	0.632	0.225
Botswana	0.926	0.700	0.135
PaviaU	0.582	0.587	0.223
SalinasA	0.872	0.452	0.255
Indian_pines	0.619	0.343	0.169
Salinas	0.921	0.405	0.194
Cuprite	0.833	0.630	-

E.3 CNMF

E.3.1 CAVE Dataset

Table E.19: Results of CNMF with CAVE dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
balloons	0.928	0.280	0.105
chart_and_stuffed_toy	0.858	0.406	0.150
pompoms	0.936	0.673	0.107
superballs	0.943	0.598	0.151
clay	0.931	0.382	0.137
cd	0.852	0.492	0.182
fake_and_real_tomatoes	0.960	0.249	0.227
fake_and_real_strawberries	0.954	0.438	0.105
sponges	0.934	0.422	0.075
real_and_fake_apples	0.941	0.272	0.141
hairs	0.951	0.604	0.147
paints	0.813	0.513	0.137
stuffed_toys	0.880	0.382	0.132
beads	0.913	0.707	0.174
fake_and_real_beers	0.979	0.508	0.046
fake_and_real_lemons	0.918	0.354	0.139
thread_spools	0.928	0.655	0.148
glass_tiles	0.886	0.726	0.168
fake_and_real_lemon_slices	0.915	0.649	0.154
jelly_beans	0.890	0.680	0.165
watercolors	0.875	0.714	0.098
real_and_fake_peppers	0.972	0.354	0.086
photo_and_face	0.932	0.346	0.154
face	0.891	0.348	0.158
flowers	0.929	0.391	0.161
oil_painting	0.918	0.766	0.126
fake_and_real_food	0.917	0.445	0.173
egyptian_statue	0.955	0.228	0.161
fake_and_real_sushi	0.927	0.375	0.209
feathers	0.895	0.523	0.154
fake_and_real_peppers	0.883	0.271	0.191
cloth	0.957	0.844	0.112

Table E.20: Results of CNMF with CAVE dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
balloons	0.933	0.290	0.103
chart_and_stuffed_toy	0.890	0.388	0.125

Table E.20: Results of CNMF with CAVE dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
pompoms	0.934	0.680	0.105
superballs	0.934	0.627	0.167
clay	0.934	0.397	0.127
cd	0.869	0.512	0.203
fake_and_real_tomatoes	0.964	0.299	0.237
fake_and_real_strawberries	0.913	0.508	0.112
sponges	0.932	0.409	0.066
real_and_fake_apples	0.930	0.272	0.135
hairs	0.947	0.558	0.139
paints	0.787	0.515	0.136
stuffed_toys	0.882	0.376	0.123
beads	0.876	0.691	0.188
fake_and_real_beers	0.981	0.510	0.042
fake_and_real_lemons	0.914	0.313	0.118
thread_spools	0.927	0.651	0.155
glass_tiles	0.900	0.733	0.164
fake_and_real_lemon_slices	0.906	0.646	0.162
jelly_beans	0.873	0.677	0.171
watercolors	0.903	0.666	0.086
real_and_fake_peppers	0.961	0.366	0.090
photo_and_face	0.925	0.316	0.153
face	0.911	0.370	0.120
flowers	0.922	0.384	0.150
oil_painting	0.922	0.739	0.113
fake_and_real_food	0.892	0.449	0.171
egyptian_statue	0.945	0.237	0.187
fake_and_real_sushi	0.938	0.449	0.209
feathers	0.873	0.528	0.153
fake_and_real_peppers	0.887	0.275	0.188
cloth	0.934	0.865	0.118

Table E.21: Results of CNMF with CAVE dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
balloons	0.932	0.292	0.106
chart_and_stuffed_toy	0.808	0.383	0.123
pompoms	0.910	0.663	0.107
superballs	0.928	0.600	0.155

Table E.21: Results of CNMF with CAVE dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
clay	0.919	0.366	0.130
cd	0.862	0.519	0.204
fake_and_real_tomatoes	0.953	0.298	0.289
fake_and_real_strawberries	0.930	0.411	0.109
sponges	0.928	0.413	0.067
real_and_fake_apples	0.960	0.232	0.148
hairs	0.944	0.590	0.145
paints	0.755	0.458	0.134
stuffed_toys	0.876	0.370	0.129
beads	0.843	0.642	0.219
fake_and_real_beers	0.979	0.525	0.044
fake_and_real_lemons	0.934	0.423	0.101
thread_spools	0.912	0.631	0.165
glass_tiles	0.894	0.754	0.184
fake_and_real_lemon_slices	0.880	0.656	0.162
jelly_beans	0.851	0.655	0.175
watercolors	0.866	0.701	0.095
real_and_fake_peppers	0.958	0.340	0.083
photo_and_face	0.915	0.288	0.153
face	0.888	0.379	0.167
flowers	0.921	0.384	0.151
oil_painting	0.901	0.737	0.129
fake_and_real_food	0.885	0.393	0.167
egyptian_statue	0.944	0.221	0.202
fake_and_real_sushi	0.926	0.407	0.224
feathers	0.860	0.515	0.163
fake_and_real_peppers	0.912	0.226	0.171
cloth	0.937	0.846	0.126

E.3.2 Harvard Dataset

Table E.22: Results of CNMF with Harvard dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgc4	0.991	0.582	0.035
imgb0	0.937	0.738	0.036
imgb1	0.984	0.723	0.027
imgc5	0.985	0.408	0.032

Table E.22: Results of CNMF with Harvard dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgb3	0.964	0.622	0.039
imgc7	0.986	0.691	0.037
imgc6	0.985	0.434	0.035
imgb2	0.954	0.688	0.048
imgb6	0.984	0.883	0.048
imgc2	0.693	0.510	0.049
imgc3	0.873	0.402	0.032
imgb7	0.960	0.588	0.057
imgc1	0.981	0.477	0.039
imgb5	0.989	0.593	0.035
imgb4	0.968	0.859	0.057
imga8	0.955	0.419	0.078
imgh0	0.977	0.508	0.041
imge7	0.947	0.818	0.030
imgd3	0.990	0.449	0.042
imgd2	0.966	0.494	0.042
imge6	0.992	0.815	0.038
imgh1	0.983	0.395	0.030
imgh3	0.971	0.642	0.050
imgd0	0.957	0.386	0.057
imge4	0.981	0.557	0.029
imgf8	0.960	0.710	0.082
imge5	0.978	0.693	0.027
imgd1	0.969	0.288	0.092
imgh2	0.986	0.469	0.057
imgh6	0.991	0.281	0.079
imgg9	0.995	0.311	0.080
imgd5	0.912	0.418	0.050
imge1	0.928	0.526	0.110
imge0	0.968	0.687	0.049
imgd4	0.983	0.404	0.028
imgg8	0.995	0.252	0.084
imgh7	0.994	0.407	0.083
imgh5	0.989	0.358	0.070
imge2	0.912	0.712	0.182
imgd6	0.992	0.450	0.101
imgd7	0.991	0.635	0.035
imge3	0.985	0.765	0.050

Table E.22: Results of CNMF with Harvard dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgh4	0.855	0.370	0.059
imgg6	0.904	0.275	0.099
imgf2	0.936	0.796	0.055
imgf3	0.973	0.692	0.059
imgg7	0.986	0.431	0.051
imgf1	0.229	0.473	0.100
imgg5	0.996	0.433	0.070
imgd9	0.953	0.427	0.033
imgd8	0.988	0.447	0.038
imgg4	0.993	0.343	0.106
imgf4	0.979	0.568	0.019
imgg0	0.944	0.491	0.069
imgg1	0.989	0.396	0.084
imgf5	0.966	0.713	0.053
imgg3	0.977	0.363	0.068
imgf7	0.947	0.691	0.157
imgf6	0.792	0.671	0.102
imgg2	0.992	0.332	0.068
img3	0.994	0.377	0.053
imga5	0.975	0.357	0.048
imgb9	0.980	0.571	0.033
imgb8	0.942	0.765	0.040
imga4	0.942	0.477	0.053
img2	0.945	0.793	0.042
imga6	0.985	0.702	0.035
imga7	0.986	0.754	0.027
img1	0.980	0.670	0.026
img5	0.994	0.356	0.056
imga3	0.906	0.596	0.054
imga2	0.993	0.382	0.038
img4	0.980	0.344	0.076
img6	0.989	0.308	0.062
imgc8	0.953	0.642	0.033
imga1	0.991	0.697	0.031
imgc9	0.977	0.789	0.047

Table E.23: Results of CNMF with Harvard dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgc4	0.990	0.577	0.041
imgb0	0.933	0.703	0.071
imgb1	0.983	0.704	0.034
imgc5	0.984	0.404	0.035
imgb3	0.859	0.610	0.063
imgc7	0.986	0.685	0.040
imgc6	0.983	0.420	0.043
imgb2	0.820	0.686	0.073
imgb6	0.986	0.881	0.050
imgc2	0.873	0.508	0.055
imgc3	0.930	0.386	0.032
imgb7	0.942	0.587	0.066
imgc1	0.979	0.440	0.046
imgb5	0.988	0.590	0.041
imgb4	0.977	0.855	0.066
imga8	0.923	0.427	0.080
imgh0	0.947	0.489	0.059
imge7	0.895	0.780	0.050
imgd3	0.989	0.455	0.041
imgd2	0.834	0.443	0.115
imge6	0.990	0.803	0.046
imgh1	0.895	0.388	0.045
imgh3	0.953	0.642	0.050
imgd0	0.945	0.396	0.055
imge4	0.982	0.565	0.029
imgf8	0.983	0.699	0.093
imge5	0.976	0.685	0.034
imgd1	0.963	0.273	0.112
imgh2	0.975	0.468	0.065
imgh6	0.990	0.270	0.080
imgg9	0.993	0.268	0.104
imgd5	0.959	0.430	0.043
imge1	0.866	0.517	0.106
imge0	0.983	0.681	0.052
imgd4	0.978	0.362	0.036
imgg8	0.996	0.252	0.083
imgh7	0.994	0.387	0.084
imgh5	0.984	0.340	0.099

Table E.23: Results of CNMF with Harvard dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imge2	0.929	0.714	0.177
imgd6	0.992	0.451	0.104
imgd7	0.981	0.608	0.070
imge3	0.851	0.709	0.141
imgh4	0.866	0.365	0.057
imgg6	0.902	0.267	0.100
imgf2	0.838	0.793	0.061
imgf3	0.954	0.696	0.071
imgg7	0.986	0.431	0.053
imgf1	0.697	0.468	0.103
imgg5	0.990	0.390	0.118
imgd9	0.973	0.411	0.036
imgd8	0.836	0.338	0.408
imgg4	0.991	0.342	0.106
imgf4	0.953	0.535	0.048
imgg0	0.929	0.480	0.083
imgg1	0.992	0.396	0.085
imgf5	0.964	0.715	0.060
imgg3	0.961	0.391	0.061
imgf7	0.885	0.696	0.166
imgf6	0.957	0.688	0.080
imgg2	0.995	0.323	0.080
img3	0.994	0.379	0.057
imga5	0.913	0.349	0.051
imgb9	0.973	0.551	0.051
imgb8	0.924	0.764	0.046
imga4	0.932	0.475	0.061
img2	0.924	0.785	0.051
imga6	0.985	0.707	0.036
imga7	0.985	0.749	0.028
img1	0.980	0.669	0.026
img5	0.995	0.357	0.057
imga3	0.919	0.609	0.067
imga2	0.993	0.388	0.040
img4	0.963	0.348	0.077
img6	0.985	0.279	0.087
imgc8	0.932	0.626	0.087
imga1	0.990	0.697	0.037

Table E.23: Results of CNMF with Harvard dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgc9	0.973	0.788	0.052

Table E.24: Results of CNMF with Harvard dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgc4	0.983	0.563	0.049
imgb0	0.933	0.712	0.056
imgb1	0.983	0.713	0.032
imgc5	0.984	0.399	0.037
imgb3	0.930	0.618	0.060
imgc7	0.985	0.677	0.044
imgc6	0.982	0.425	0.044
imgb2	0.789	0.667	0.098
imgb6	0.980	0.878	0.053
imgc2	0.967	0.539	0.040
imgc3	0.929	0.400	0.030
imgb7	0.896	0.576	0.079
imgc1	0.977	0.425	0.050
imgb5	0.987	0.584	0.041
imgb4	0.966	0.854	0.065
imga8	0.913	0.433	0.112
imgh0	0.965	0.502	0.047
imge7	0.844	0.781	0.049
imgd3	0.186	0.413	0.209
imgd2	0.917	0.486	0.057
imge6	0.984	0.807	0.046
imgh1	0.888	0.391	0.035
imgh3	0.927	0.640	0.062
imgd0	0.938	0.393	0.051
imge4	0.979	0.545	0.032
imgf8	0.988	0.701	0.086
imge5	0.975	0.689	0.033
imgd1	0.954	0.251	0.129
imgh2	0.982	0.458	0.083
imgh6	0.991	0.284	0.079
imgg9	0.993	0.292	0.096
imgd5	0.768	0.418	0.048
imge1	0.924	0.521	0.113

Table E.24: Results of CNMF with Harvard dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imge0	0.960	0.679	0.059
imgd4	0.976	0.372	0.034
imgg8	0.993	0.264	0.100
imgh7	0.993	0.390	0.084
imgh5	0.986	0.349	0.080
imge2	0.921	0.713	0.183
imgd6	0.988	0.467	0.082
imgd7	0.970	0.588	0.095
imge3	0.982	0.755	0.063
imgh4	0.943	0.369	0.065
imgg6	0.900	0.267	0.100
imgf2	0.909	0.767	0.075
imgf3	0.788	0.647	0.147
imgg7	0.984	0.427	0.067
imgf1	0.831	0.441	0.102
imgg5	0.991	0.402	0.112
imgd9	0.955	0.419	0.035
imgd8	0.952	0.405	0.222
imgg4	0.991	0.351	0.115
imgf4	0.965	0.561	0.032
imgg0	0.927	0.485	0.086
imgg1	0.989	0.372	0.101
imgf5	0.954	0.708	0.081
imgg3	0.957	0.371	0.064
imgf7	0.964	0.713	0.090
imgf6	0.903	0.684	0.090
imgg2	0.992	0.327	0.080
img3	0.826	0.299	0.165
imga5	0.952	0.351	0.049
imgb9	0.979	0.569	0.035
imgb8	0.714	0.752	0.055
imga4	0.833	0.467	0.090
img2	0.977	0.776	0.058
imga6	0.987	0.708	0.037
imga7	0.984	0.746	0.029
img1	0.977	0.652	0.030
img5	0.994	0.331	0.065
imga3	0.684	0.590	0.075

Table E.24: Results of CNMF with Harvard dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imga2	0.992	0.375	0.049
img4	0.991	0.336	0.084
img6	0.987	0.305	0.067
imgc8	0.941	0.641	0.043
imga1	0.989	0.696	0.040
imgc9	0.970	0.781	0.054

E.3.3 EHU Dataset

Table E.25: Results of CNMF with EHU dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
KSC	0.906	0.230	1.339
Pavia	0.864	0.647	0.230
Botswana	0.924	0.651	0.140
PaviaU	0.810	0.559	0.214
SalinasA	0.931	0.395	0.218
Indian_pines	0.685	0.291	0.135
Salinas	0.939	0.388	0.200
Cuprite	0.873	0.438	-

Table E.26: Results of CNMF with EHU dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
KSC	0.898	0.334	1.208
Pavia	0.828	0.639	0.246
Botswana	0.917	0.646	0.152
PaviaU	0.777	0.529	0.242
SalinasA	0.885	0.410	0.244
Indian_pines	0.642	0.310	0.151
Salinas	0.917	0.385	0.214
Cuprite	0.859	0.421	-

Table E.27: Results of CNMF with EHU dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
KSC	0.894	0.301	0.960
Pavia	0.853	0.602	0.252

Table E.27: Results of CNMF with EHU dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
Botswana	0.923	0.600	0.143
PaviaU	0.756	0.469	0.247
SalinasA	0.875	0.360	0.258
Indian_pines	0.682	0.300	0.130
Salinas	0.905	0.389	0.224
Cuprite	0.846	0.432	-

E.4 HySure

E.4.1 CAVE Dataset

Table E.28: Results of HySure with CAVE dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
balloons	0.907	0.231	0.171
chart_and_stuffed_toy	0.802	0.425	0.167
pompoms	0.890	0.747	0.133
superballs	0.893	0.648	0.207
clay	0.910	0.411	0.158
cd	0.873	0.555	0.162
fake_and_real_tomatoes	0.940	0.345	0.224
fake_and_real_strawberries	0.877	0.547	0.167
sponges	0.855	0.360	0.102
real_and_fake_apples	0.914	0.284	0.183
hairs	0.939	0.634	0.153
paints	0.819	0.453	0.136
stuffed_toys	0.873	0.327	0.149
beads	0.888	0.653	0.194
fake_and_real_beers	0.960	0.451	0.071
fake_and_real_lemons	0.869	0.396	0.163
thread_spools	0.914	0.654	0.156
glass_tiles	0.881	0.692	0.184
fake_and_real_lemon_slices	0.896	0.637	0.209
jelly_beans	0.878	0.607	0.187
watercolors	0.884	0.725	0.093
real_and_fake_peppers	0.864	0.347	0.168
photo_and_face	0.910	0.279	0.200

Table E.28: Results of HySure with CAVE dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
face	0.895	0.373	0.147
flowers	0.911	0.349	0.155
oil_painting	0.910	0.777	0.145
fake_and_real_food	0.894	0.467	0.180
egyptian_statue	0.909	0.192	0.261
fake_and_real_sushi	0.905	0.477	0.214
feathers	0.875	0.519	0.171
fake_and_real_peppers	0.877	0.286	0.191
cloth	0.934	0.828	0.135

Table E.29: Results of HySure with CAVE dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
balloons	0.880	0.248	0.191
chart_and_stuffed_toy	0.785	0.454	0.188
pompoms	0.856	0.720	0.149
superballs	0.878	0.645	0.229
clay	0.875	0.434	0.187
cd	0.856	0.542	0.195
fake_and_real_tomatoes	0.916	0.332	0.265
fake_and_real_strawberries	0.839	0.516	0.206
sponges	0.810	0.384	0.118
real_and_fake_apples	0.889	0.284	0.221
hairs	0.759	0.499	0.399
paints	0.796	0.473	0.148
stuffed_toys	0.851	0.368	0.176
beads	0.865	0.690	0.223
fake_and_real_beers	0.954	0.521	0.079
fake_and_real_lemons	0.852	0.395	0.183
thread_spools	0.902	0.658	0.177
glass_tiles	0.864	0.724	0.207
fake_and_real_lemon_slices	0.880	0.642	0.224
jelly_beans	0.859	0.637	0.201
watercolors	0.875	0.728	0.101
real_and_fake_peppers	0.858	0.339	0.182
photo_and_face	0.832	0.254	0.485
face	0.855	0.362	0.209
flowers	0.884	0.392	0.195

Table E.29: Results of HySure with CAVE dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
oil_painting	0.888	0.764	0.165
fake_and_real_food	0.868	0.467	0.211
egyptian_statue	0.879	0.202	0.343
fake_and_real_sushi	0.873	0.458	0.279
feathers	0.857	0.554	0.190
fake_and_real_peppers	0.855	0.292	0.219
cloth	0.916	0.842	0.149

Table E.30: Results of HySure with CAVE dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
balloons	0.854	0.278	0.210
chart_and_stuffed_toy	0.774	0.467	0.204
pompoms	0.832	0.728	0.163
superballs	0.826	0.661	0.273
clay	0.832	0.436	0.230
cd	0.825	0.546	0.265
fake_and_real_tomatoes	0.819	0.337	0.408
fake_and_real_strawberries	0.838	0.540	0.230
sponges	0.763	0.403	0.153
real_and_fake_apples	0.846	0.297	0.265
hairs	0.611	0.496	0.566
paints	0.751	0.496	0.165
stuffed_toys	0.827	0.388	0.209
beads	0.845	0.718	0.244
fake_and_real_beers	0.936	0.529	0.093
fake_and_real_lemons	0.773	0.399	0.232
thread_spools	0.883	0.669	0.203
glass_tiles	0.858	0.750	0.224
fake_and_real_lemon_slices	0.860	0.666	0.242
jelly_beans	0.839	0.670	0.212
watercolors	0.877	0.733	0.110
real_and_fake_peppers	0.793	0.336	0.220
photo_and_face	0.887	0.340	0.251
face	0.834	0.384	0.259
flowers	0.816	0.398	0.264
oil_painting	0.865	0.767	0.187
fake_and_real_food	0.832	0.467	0.244

Table E.30: Results of HySure with CAVE dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
egyptian_statue	0.851	0.221	0.402
fake_and_real_sushi	0.794	0.468	0.358
feathers	0.827	0.566	0.214
fake_and_real_peppers	0.824	0.318	0.246
cloth	0.884	0.840	0.173

E.4.2 Harvard Dataset

Table E.31: Results of HySure with Harvard dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgc4	0.990	0.579	0.038
imgb0	0.962	0.746	0.036
imgb1	0.984	0.722	0.028
imgc5	0.986	0.430	0.030
imgb3	0.974	0.618	0.040
imgc7	0.986	0.683	0.038
imgc6	0.985	0.440	0.037
imgb2	0.964	0.667	0.053
imgb6	0.988	0.880	0.045
imgc2	0.649	0.528	0.036
imgc3	0.935	0.412	0.030
imgb7	0.958	0.572	0.058
imgc1	0.979	0.471	0.042
imgb5	0.988	0.591	0.038
imgb4	0.963	0.859	0.056
imga8	0.937	0.406	0.085
imgh0	0.987	0.501	0.045
imge7	0.936	0.819	0.032
imgd3	0.990	0.455	0.040
imgd2	0.965	0.496	0.049
imge6	0.990	0.819	0.040
imgh1	0.982	0.381	0.028
imgh3	0.979	0.643	0.050
imgd0	0.971	0.402	0.041
imge4	0.984	0.570	0.026
imgf8	0.987	0.702	0.086
imge5	0.976	0.683	0.028

Table E.31: Results of HySure with Harvard dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgd1	0.984	0.334	0.068
imgh2	0.994	0.466	0.060
imgh6	0.988	0.320	0.091
imgg9	0.994	0.310	0.090
imgd5	0.950	0.441	0.041
imge1	0.937	0.518	0.086
imge0	0.956	0.686	0.053
imgd4	0.987	0.405	0.028
imgg8	0.992	0.260	0.095
imgh7	0.994	0.412	0.086
imgh5	0.953	0.354	0.074
imge2	0.952	0.720	0.151
imgd6	0.994	0.460	0.095
imgd7	0.990	0.641	0.040
imge3	0.980	0.757	0.051
imgh4	0.946	0.325	0.062
imgg6	0.951	0.307	0.099
imgf2	0.970	0.788	0.057
imgf3	0.933	0.702	0.062
imgg7	0.984	0.415	0.057
imgf1	0.932	0.481	0.043
imgg5	0.994	0.437	0.081
imgd9	0.949	0.398	0.036
imgd8	0.987	0.458	0.041
imgg4	0.988	0.369	0.120
imgf4	0.981	0.581	0.020
imgg0	0.945	0.496	0.080
imgg1	0.990	0.401	0.094
imgf5	0.976	0.717	0.051
imgg3	0.974	0.404	0.061
imgf7	0.899	0.738	0.046
imgf6	0.960	0.694	0.082
imgg2	0.987	0.326	0.083
img3	0.992	0.323	0.061
imga5	0.984	0.370	0.045
imgb9	0.980	0.569	0.033
imgb8	0.949	0.761	0.039
imga4	0.964	0.468	0.062

Table E.31: Results of HySure with Harvard dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
img2	0.966	0.772	0.047
imga6	0.986	0.707	0.035
imga7	0.985	0.749	0.027
img1	0.979	0.669	0.025
img5	0.993	0.355	0.068
imga3	0.944	0.605	0.054
imga2	0.993	0.389	0.042
img4	0.989	0.349	0.087
img6	0.986	0.330	0.073
imgc8	0.979	0.630	0.037
imga1	0.990	0.691	0.033
imgc9	0.976	0.788	0.047

Table E.32: Results of HySure with Harvard dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgc4	0.989	0.580	0.042
imgb0	0.877	0.734	0.040
imgb1	0.983	0.719	0.029
imgc5	0.986	0.431	0.032
imgb3	0.962	0.625	0.046
imgc7	0.985	0.681	0.042
imgc6	0.984	0.439	0.041
imgb2	0.935	0.675	0.061
imgb6	0.978	0.876	0.051
imgc2	0.693	0.533	0.037
imgc3	0.934	0.416	0.031
imgb7	0.933	0.579	0.074
imgc1	0.978	0.464	0.046
imgb5	0.988	0.594	0.039
imgb4	0.967	0.859	0.062
imga8	0.922	0.426	0.097
imgh0	0.985	0.501	0.048
imge7	0.927	0.813	0.034
imgd3	0.987	0.454	0.046
imgd2	0.947	0.494	0.058
imge6	0.988	0.814	0.044
imgh1	0.973	0.386	0.030

Table E.32: Results of HySure with Harvard dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgh3	0.978	0.645	0.050
imgd0	0.954	0.409	0.047
imge4	0.983	0.567	0.028
imgf8	0.985	0.701	0.091
imge5	0.976	0.694	0.028
imgd1	0.972	0.332	0.070
imgh2	0.988	0.467	0.072
imgh6	0.983	0.312	0.094
imgg9	0.993	0.310	0.095
imgd5	0.911	0.432	0.047
imge1	0.911	0.516	0.106
imge0	0.849	0.672	0.059
imgd4	0.986	0.408	0.029
imgg8	0.985	0.251	0.101
imgh7	0.994	0.409	0.085
imgh5	0.952	0.351	0.080
imge2	0.942	0.721	0.176
imgd6	0.993	0.461	0.105
imgd7	0.988	0.639	0.046
imge3	0.981	0.769	0.056
imgh4	0.938	0.347	0.065
imgg6	0.951	0.308	0.099
imgf2	0.967	0.780	0.066
imgf3	0.913	0.695	0.072
imgg7	0.983	0.419	0.061
imgf1	0.886	0.482	0.062
imgg5	0.993	0.438	0.090
imgd9	0.946	0.393	0.038
imgd8	0.984	0.454	0.045
imgg4	0.983	0.366	0.136
imgf4	0.975	0.571	0.023
imgg0	0.931	0.497	0.093
imgg1	0.989	0.402	0.099
imgf5	0.963	0.716	0.064
imgg3	0.963	0.406	0.070
imgf7	0.862	0.740	0.048
imgf6	0.940	0.691	0.090
imgg2	0.985	0.324	0.093

Table E.32: Results of HySure with Harvard dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
img3	0.992	0.325	0.064
imga5	0.972	0.367	0.049
imgb9	0.980	0.569	0.034
imgb8	0.961	0.764	0.044
imga4	0.927	0.452	0.071
img2	0.985	0.778	0.049
imga6	0.984	0.704	0.037
imga7	0.985	0.754	0.028
img1	0.978	0.665	0.027
img5	0.991	0.356	0.076
imga3	0.940	0.609	0.072
imga2	0.991	0.388	0.051
img4	0.989	0.346	0.095
img6	0.985	0.330	0.077
imgc8	0.964	0.622	0.049
imga1	0.989	0.695	0.036
imgc9	0.974	0.795	0.049

Table E.33: Results of HySure with Harvard dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgc4	0.985	0.577	0.051
imgb0	0.844	0.731	0.044
imgb1	0.981	0.713	0.033
imgc5	0.984	0.424	0.035
imgb3	0.943	0.623	0.060
imgc7	0.984	0.678	0.044
imgc6	0.983	0.439	0.044
imgb2	0.886	0.682	0.074
imgb6	0.914	0.868	0.054
imgc2	0.630	0.529	0.038
imgc3	0.943	0.414	0.032
imgb7	0.830	0.570	0.099
imgc1	0.975	0.466	0.054
imgb5	0.986	0.592	0.042
imgb4	0.925	0.851	0.069
imga8	0.918	0.425	0.113
imgh0	0.977	0.508	0.052

Table E.33: Results of HySure with Harvard dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imge7	0.929	0.815	0.035
imgd3	0.979	0.453	0.054
imgd2	0.929	0.491	0.064
imge6	0.986	0.815	0.049
imgh1	0.985	0.393	0.034
imgh3	0.974	0.645	0.052
imgd0	0.946	0.407	0.059
imge4	0.983	0.564	0.029
imgf8	0.988	0.701	0.093
imge5	0.976	0.694	0.029
imgd1	0.960	0.344	0.072
imgh2	0.978	0.465	0.078
imgh6	0.978	0.304	0.096
imgg9	0.993	0.312	0.098
imgd5	0.939	0.406	0.053
imge1	0.915	0.531	0.115
imge0	0.580	0.620	0.080
imgd4	0.982	0.406	0.032
imgg8	0.989	0.242	0.109
imgh7	0.995	0.418	0.084
imgh5	0.906	0.348	0.098
imge2	0.940	0.720	0.187
imgd6	0.986	0.444	0.124
imgd7	0.985	0.638	0.058
imge3	0.978	0.763	0.064
imgh4	0.950	0.378	0.073
imgg6	0.949	0.308	0.100
imgf2	0.961	0.795	0.066
imgf3	0.909	0.688	0.083
imgg7	0.981	0.421	0.072
imgf1	0.860	0.472	0.089
imgg5	0.990	0.436	0.100
imgd9	0.940	0.392	0.045
imgd8	0.977	0.445	0.054
imgg4	0.981	0.358	0.140
imgf4	0.962	0.563	0.030
imgg0	0.923	0.499	0.095
imgg1	0.985	0.389	0.110

Table E.33: Results of HySure with Harvard dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgf5	0.943	0.716	0.081
imgg3	0.958	0.405	0.074
imgf7	0.746	0.729	0.056
imgf6	0.868	0.685	0.100
imgg2	0.981	0.339	0.088
img3	0.986	0.325	0.076
imga5	0.960	0.357	0.054
imgb9	0.979	0.565	0.037
imgb8	0.926	0.758	0.055
imga4	0.875	0.445	0.095
img2	0.974	0.788	0.051
imga6	0.981	0.696	0.042
imga7	0.985	0.753	0.030
img1	0.975	0.660	0.031
img5	0.990	0.350	0.079
imga3	0.924	0.609	0.090
imga2	0.989	0.315	0.053
img4	0.991	0.349	0.103
img6	0.984	0.326	0.080
imgc8	0.937	0.624	0.060
imga1	0.987	0.699	0.041
imgc9	0.972	0.796	0.053

E.4.3 EHU Dataset

Table E.34: Results of HySure with EHU dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
KSC	0.623	0.299	1.028
Pavia	0.874	0.633	0.207
Botswana	0.817	0.229	0.292
PaviaU	0.819	0.581	0.217
SalinasA	0.862	0.427	0.260
Indian_pines	0.558	0.151	0.208
Salinas	0.928	0.422	0.205
Cuprite	0.783	0.655	-

Table E.35: Results of HySure with EHU dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
KSC	0.643	0.420	1.045
Pavia	0.807	0.623	0.262
Botswana	0.727	0.292	0.536
PaviaU	0.678	0.532	0.328
SalinasA	0.818	0.432	0.300
Indian_pines	0.554	0.212	0.209
Salinas	0.880	0.387	0.234
Cuprite	0.707	0.653	-

Table E.36: Results of HySure with EHU dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
KSC	0.688	0.447	1.051
Pavia	0.754	0.605	0.315
Botswana	0.720	0.316	0.574
PaviaU	0.683	0.537	0.338
SalinasA	0.797	0.438	0.312
Indian_pines	0.577	0.306	0.210
Salinas	0.826	0.370	0.281
Cuprite	0.688	0.652	-

E.5 SupResPALM

E.5.1 CAVE Dataset

Table E.37: Results of SupResPALM with CAVE dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
balloons	0.866	0.289	0.321
chart_and_stuffed_toy	0.744	0.435	0.253
pompoms	0.898	0.743	0.207
superballs	0.883	0.659	0.287
clay	0.870	0.428	0.344
cd	0.838	0.556	0.340
fake_and_real_tomatoes	0.864	0.312	0.401
fake_and_real_strawberries	0.840	0.509	0.264
sponges	0.915	0.443	0.132

Table E.37: Results of SupResPALM with CAVE dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
real_and_fake_apples	0.880	0.282	0.321
hairs	0.901	0.621	0.315
paints	0.869	0.508	0.199
stuffed_toys	0.817	0.362	0.283
beads	0.787	0.637	0.346
fake_and_real_beers	0.964	0.544	0.089
fake_and_real_lemons	0.873	0.432	0.231
thread_spools	0.890	0.631	0.256
glass_tiles	0.850	0.716	0.279
fake_and_real_lemon_slices	0.883	0.603	0.363
jelly_beans	0.819	0.631	0.308
watercolors	0.906	0.710	0.128
real_and_fake_peppers	0.895	0.409	0.233
photo_and_face	0.881	0.313	0.376
face	0.872	0.398	0.345
flowers	0.829	0.397	0.356
oil_painting	0.897	0.778	0.181
fake_and_real_food	0.852	0.443	0.364
egyptian_statue	0.881	0.246	0.482
fake_and_real_sushi	0.852	0.371	0.369
feathers	0.832	0.555	0.249
fake_and_real_peppers	0.862	0.326	0.268
cloth	0.946	0.832	0.143

Table E.38: Results of SupResPALM with CAVE dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
balloons	0.900	0.308	0.301
chart_and_stuffed_toy	0.740	0.435	0.253
pompoms	0.886	0.733	0.218
superballs	0.883	0.656	0.287
clay	0.866	0.427	0.346
cd	0.836	0.556	0.340
fake_and_real_tomatoes	0.874	0.313	0.383
fake_and_real_strawberries	0.839	0.504	0.266
sponges	0.917	0.461	0.134
real_and_fake_apples	0.874	0.284	0.329
hairs	0.909	0.628	0.319

Table E.38: Results of SupResPALM with CAVE dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
paints	0.834	0.492	0.195
stuffed_toys	0.812	0.361	0.278
beads	0.768	0.610	0.369
fake_and_real_beers	0.968	0.557	0.088
fake_and_real_lemons	0.870	0.435	0.236
thread_spools	0.886	0.633	0.263
glass_tiles	0.883	0.729	0.255
fake_and_real_lemon_slices	0.862	0.625	0.380
jelly_beans	0.794	0.613	0.327
watercolors	0.911	0.736	0.123
real_and_fake_peppers	0.912	0.398	0.220
photo_and_face	0.884	0.318	0.379
face	0.876	0.368	0.348
flowers	0.834	0.393	0.343
oil_painting	0.907	0.786	0.174
fake_and_real_food	0.847	0.451	0.367
egyptian_statue	0.885	0.251	0.462
fake_and_real_sushi	0.849	0.388	0.381
feathers	0.837	0.567	0.213
fake_and_real_peppers	0.870	0.326	0.262
cloth	0.932	0.839	0.174

Table E.39: Results of SupResPALM with CAVE dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
balloons	0.865	0.284	0.329
chart_and_stuffed_toy	0.751	0.433	0.250
pompoms	0.869	0.709	0.225
superballs	0.881	0.655	0.292
clay	0.872	0.438	0.331
cd	0.861	0.556	0.344
fake_and_real_tomatoes	0.879	0.313	0.382
fake_and_real_strawberries	0.832	0.505	0.270
sponges	0.911	0.431	0.136
real_and_fake_apples	0.876	0.266	0.337
hairs	0.900	0.606	0.349
paints	0.851	0.457	0.198
stuffed_toys	0.814	0.369	0.264

Table E.39: Results of SupResPALM with CAVE dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
beads	0.764	0.612	0.367
fake_and_real_beers	0.971	0.546	0.082
fake_and_real_lemons	0.865	0.420	0.240
thread_spools	0.893	0.649	0.225
glass_tiles	0.887	0.754	0.232
fake_and_real_lemon_slices	0.865	0.609	0.396
jelly_beans	0.775	0.591	0.346
watercolors	0.909	0.731	0.118
real_and_fake_peppers	0.921	0.398	0.214
photo_and_face	0.888	0.315	0.375
face	0.870	0.343	0.344
flowers	0.829	0.384	0.343
oil_painting	0.903	0.760	0.173
fake_and_real_food	0.848	0.446	0.298
egyptian_statue	0.881	0.251	0.464
fake_and_real_sushi	0.846	0.335	0.400
feathers	0.843	0.573	0.210
fake_and_real_peppers	0.871	0.317	0.285
cloth	0.920	0.834	0.172

E.5.2 Harvard Dataset

Table E.40: Results of SupResPALM with Harvard dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgc4	0.990	0.574	0.036
imgb0	0.960	0.730	0.039
imgb1	0.985	0.716	0.027
imgc5	0.986	0.409	0.029
imgb3	0.983	0.615	0.049
imgc7	0.986	0.682	0.038
imgc6	0.984	0.414	0.038
imgb2	0.981	0.678	0.071
imgb6	0.974	0.875	0.047
imgc2	0.966	0.535	0.038
imgc3	0.966	0.404	0.029
imgb7	0.946	0.572	0.068
imgc1	0.981	0.458	0.040

Table E.40: Results of SupResPALM with Harvard dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgb5	0.989	0.586	0.035
imgb4	0.959	0.859	0.054
imga8	0.982	0.424	0.120
imgh0	0.981	0.502	0.043
imge7	0.959	0.812	0.031
imgd3	0.990	0.435	0.038
imgd2	0.985	0.473	0.057
imge6	0.991	0.810	0.037
imgh1	0.871	0.382	0.029
imgh3	0.989	0.639	0.064
imgd0	0.965	0.390	0.046
imge4	0.984	0.558	0.026
imgf8	0.983	0.701	0.083
imge5	0.978	0.690	0.027
imgd1	0.971	0.319	0.055
imgh2	0.945	0.459	0.063
imgh6	0.991	0.272	0.078
imgg9	0.996	0.302	0.078
imgd5	0.991	0.410	0.043
imge1	0.936	0.527	0.099
imge0	0.991	0.674	0.052
imgd4	0.980	0.377	0.029
imgg8	0.996	0.268	0.077
imgh7	0.957	0.386	0.082
imgh5	0.988	0.346	0.069
imge2	0.943	0.704	0.167
imgd6	0.995	0.453	0.074
imgd7	0.991	0.634	0.037
imge3	0.988	0.761	0.047
imgh4	0.969	0.343	0.062
imgg6	0.949	0.273	0.097
imgf2	0.977	0.795	0.059
imgf3	0.974	0.693	0.064
imgg7	0.986	0.419	0.051
imgf1	0.912	0.467	0.100
imgg5	0.996	0.431	0.069
imgd9	0.949	0.405	0.034
imgd8	0.988	0.434	0.038

Table E.40: Results of SupResPALM with Harvard dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgg4	0.994	0.337	0.102
imgf4	0.986	0.572	0.019
imgg0	0.877	0.492	0.069
imgg1	0.990	0.393	0.083
imgf5	0.977	0.710	0.070
imgg3	0.964	0.371	0.059
imgf7	0.993	0.738	0.042
imgf6	0.983	0.679	0.122
imgg2	0.995	0.296	0.069
img3	0.911	0.332	0.091
imga5	0.983	0.357	0.083
imgb9	0.981	0.568	0.033
imgb8	0.972	0.753	0.042
imga4	0.911	0.459	0.100
img2	0.924	0.781	0.044
imga6	0.987	0.692	0.036
imga7	0.986	0.749	0.026
img1	0.979	0.666	0.026
img5	0.995	0.332	0.056
imga3	0.967	0.609	0.066
imga2	0.994	0.374	0.038
img4	0.929	0.334	0.082
img6	0.990	0.314	0.060
imgc8	0.985	0.639	0.034
imga1	0.991	0.694	0.031
imgc9	0.978	0.790	0.045

Table E.41: Results of SupResPALM with Harvard dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgc4	0.985	0.568	0.042
imgb0	0.988	0.726	0.040
imgb1	0.984	0.713	0.028
imgc5	0.980	0.405	0.030
imgb3	0.978	0.607	0.061
imgc7	0.985	0.678	0.041
imgc6	0.984	0.414	0.038
imgb2	0.974	0.679	0.073

Table E.41: Results of SupResPALM with Harvard dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgb6	0.972	0.872	0.052
imgc2	0.968	0.537	0.038
imgc3	0.986	0.396	0.030
imgb7	0.953	0.564	0.073
imgc1	0.979	0.454	0.041
imgb5	0.988	0.583	0.037
imgb4	0.972	0.847	0.064
imga8	0.862	0.434	0.152
imgh0	0.915	0.495	0.051
imge7	0.989	0.811	0.033
imgd3	0.989	0.430	0.042
imgd2	0.925	0.472	0.086
imge6	0.990	0.806	0.041
imgh1	0.879	0.378	0.028
imgh3	0.868	0.638	0.085
imgd0	0.956	0.386	0.045
imge4	0.914	0.553	0.029
imgf8	0.924	0.691	0.095
imge5	0.976	0.683	0.029
imgd1	0.989	0.293	0.058
imgh2	0.922	0.450	0.072
imgh6	0.990	0.259	0.080
imgg9	0.976	0.298	0.081
imgd5	0.990	0.403	0.057
imge1	0.926	0.520	0.108
imge0	0.806	0.671	0.062
imgd4	0.980	0.382	0.029
imgg8	0.987	0.264	0.081
imgh7	0.961	0.380	0.084
imgh5	0.986	0.345	0.072
imge2	0.925	0.710	0.179
imgd6	0.992	0.463	0.076
imgd7	0.988	0.631	0.039
imge3	0.986	0.759	0.052
imgh4	0.992	0.348	0.065
imgg6	0.951	0.259	0.099
imgf2	0.974	0.785	0.073
imgf3	0.966	0.683	0.075

Table E.41: Results of SupResPALM with Harvard dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgg7	0.985	0.422	0.054
imgf1	0.958	0.452	0.097
imgg5	0.930	0.427	0.071
imgd9	0.953	0.409	0.034
imgd8	0.985	0.438	0.040
imgg4	0.991	0.338	0.107
imgf4	0.984	0.566	0.024
imgg0	0.966	0.493	0.071
imgg1	0.978	0.386	0.087
imgf5	0.938	0.713	0.085
imgg3	0.967	0.360	0.059
imgf7	0.984	0.730	0.047
imgf6	0.934	0.674	0.156
imgg2	0.984	0.299	0.071
img3	0.988	0.328	0.088
imga5	0.970	0.345	0.074
imgb9	0.893	0.565	0.034
imgb8	0.976	0.755	0.048
imga4	0.990	0.455	0.121
img2	0.927	0.771	0.052
imga6	0.984	0.696	0.037
imga7	0.978	0.746	0.027
img1	0.979	0.662	0.026
img5	0.995	0.338	0.056
imga3	0.958	0.607	0.130
imga2	0.935	0.373	0.042
img4	0.885	0.339	0.086
img6	0.988	0.311	0.061
imgc8	0.966	0.629	0.044
imga1	0.987	0.692	0.033
imgc9	0.974	0.785	0.050

Table E.42: Results of SupResPALM with Harvard dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgc4	0.949	0.561	0.053
imgb0	0.886	0.707	0.051
imgb1	0.981	0.699	0.032

Table E.42: Results of SupResPALM with Harvard dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgc5	0.980	0.388	0.033
imgb3	0.979	0.600	0.073
imgc7	0.983	0.668	0.047
imgc6	0.963	0.411	0.042
imgb2	0.975	0.674	0.093
imgb6	0.889	0.871	0.057
imgc2	0.967	0.529	0.040
imgc3	0.933	0.388	0.039
imgb7	0.955	0.554	0.084
imgc1	0.977	0.433	0.047
imgb5	0.986	0.566	0.042
imgb4	0.823	0.826	0.080
imga8	0.754	0.428	0.190
imgh0	0.822	0.481	0.055
imge7	0.851	0.794	0.041
imgd3	0.980	0.424	0.055
imgd2	0.856	0.456	0.108
imge6	0.945	0.800	0.047
imgh1	0.876	0.358	0.030
imgh3	0.549	0.627	0.125
imgd0	0.956	0.382	0.055
imge4	0.828	0.543	0.031
imgf8	0.894	0.677	0.107
imge5	0.976	0.683	0.030
imgd1	0.993	0.277	0.062
imgh2	0.829	0.448	0.116
imgh6	0.990	0.255	0.079
imgg9	0.979	0.277	0.088
imgd5	0.965	0.410	0.067
imge1	0.923	0.515	0.112
imge0	0.754	0.661	0.074
imgd4	0.917	0.370	0.032
imgg8	0.942	0.249	0.088
imgh7	0.921	0.369	0.086
imgh5	0.986	0.354	0.079
imge2	0.850	0.708	0.189
imgd6	0.959	0.455	0.082
imgd7	0.944	0.620	0.047

Table E.42: Results of SupResPALM with Harvard dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imge3	0.976	0.755	0.059
imgh4	0.920	0.340	0.085
imgg6	0.924	0.252	0.099
imgf2	0.965	0.781	0.083
imgf3	0.866	0.674	0.089
imgg7	0.984	0.427	0.068
imgf1	0.987	0.452	0.113
imgg5	0.911	0.401	0.075
imgd9	0.905	0.398	0.037
imgd8	0.968	0.428	0.045
imgg4	0.978	0.315	0.118
imgf4	0.978	0.558	0.030
imgg0	0.857	0.480	0.081
imgg1	0.946	0.381	0.091
imgf5	0.838	0.705	0.120
imgg3	0.993	0.350	0.060
imgf7	0.979	0.728	0.054
imgf6	0.897	0.667	0.165
imgg2	0.956	0.291	0.082
img3	0.985	0.329	0.113
imga5	0.839	0.338	0.106
imgb9	0.918	0.552	0.036
imgb8	0.975	0.747	0.058
imga4	0.976	0.462	0.135
img2	0.849	0.766	0.057
imga6	0.969	0.685	0.041
imga7	0.892	0.737	0.029
img1	0.975	0.646	0.030
img5	0.994	0.314	0.060
imga3	0.982	0.602	0.181
imga2	0.915	0.356	0.053
img4	0.748	0.320	0.094
img6	0.989	0.284	0.064
imgc8	0.973	0.627	0.063
imga1	0.966	0.689	0.037
imgc9	0.938	0.777	0.058

E.5.3 EHU Dataset

Table E.43: Results of SupResPALM with EHU dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
KSC	0.656	0.203	0.963
Pavia	0.777	0.608	0.281
Botswana	0.922	0.664	0.183
PaviaU	0.748	0.609	0.272
SalinasA	0.901	0.456	0.248
Indian_pines	0.741	0.357	0.125
Salinas	0.836	0.314	0.272
Cuprite	0.837	0.558	-

Table E.44: Results of SupResPALM with EHU dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
KSC	0.737	0.214	0.974
Pavia	0.788	0.602	0.286
Botswana	0.919	0.654	0.192
PaviaU	0.753	0.607	0.271
SalinasA	0.882	0.463	0.258
Indian_pines	0.714	0.377	0.130
Salinas	0.861	0.326	0.271
Cuprite	0.823	0.589	-

Table E.45: Results of SupResPALM with EHU dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
KSC	0.695	0.180	0.975
Pavia	0.818	0.614	0.274
Botswana	0.916	0.687	0.189
PaviaU	0.719	0.617	0.267
SalinasA	0.857	0.441	0.281
Indian_pines	0.631	0.412	0.158
Salinas	0.836	0.274	0.281
Cuprite	0.815	0.619	-

E.6 FUSE

E.6.1 CAVE Dataset

Table E.46: Results of FUSE with CAVE dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
balloons	0.955	0.289	0.083
chart_and_stuffed_toy	0.915	0.465	0.098
pompoms	0.933	0.725	0.086
superballs	0.936	0.655	0.147
clay	0.945	0.424	0.117
cd	0.916	0.573	0.142
fake_and_real_tomatoes	0.860	0.144	0.347
fake_and_real_strawberries	0.941	0.549	0.104
sponges	0.915	0.440	0.071
real_and_fake_apples	0.959	0.279	0.110
hairs	0.229	0.188	0.761
paints	0.919	0.525	0.090
stuffed_toys	0.931	0.376	0.095
beads	0.928	0.693	0.161
fake_and_real_beers	0.973	0.521	0.044
fake_and_real_lemons	0.939	0.415	0.095
thread_spools	0.939	0.668	0.125
glass_tiles	0.913	0.732	0.155
fake_and_real_lemon_slices	0.933	0.673	0.151
jelly_beans	0.926	0.658	0.134
watercolors	0.942	0.724	0.060
real_and_fake_peppers	0.932	0.330	0.103
photo_and_face	0.952	0.250	0.136
face	0.941	0.270	0.113
flowers	0.942	0.383	0.105
oil_painting	0.929	0.769	0.117
fake_and_real_food	0.935	0.470	0.129
egyptian_statue	0.918	0.091	0.220
fake_and_real_sushi	0.914	0.348	0.186
feathers	0.917	0.559	0.118
fake_and_real_peppers	0.945	0.311	0.117
cloth	0.936	0.815	0.137

Table E.47: Results of FUSE with CAVE dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
balloons	0.928	0.239	0.101
chart_and_stuffed_toy	0.867	0.435	0.125

Table E.47: Results of FUSE with CAVE dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
pompoms	0.902	0.694	0.102
superballs	0.920	0.642	0.177
clay	0.931	0.411	0.135
cd	0.892	0.554	0.171
fake_and_real_tomatoes	0.792	0.125	0.475
fake_and_real_strawberries	0.901	0.488	0.139
sponges	0.874	0.394	0.085
real_and_fake_apples	0.939	0.268	0.137
hairs	0.519	0.351	0.479
paints	0.880	0.486	0.108
stuffed_toys	0.899	0.346	0.119
beads	0.890	0.691	0.200
fake_and_real_beers	0.956	0.497	0.056
fake_and_real_lemons	0.919	0.384	0.112
thread_spools	0.919	0.659	0.151
glass_tiles	0.882	0.721	0.190
fake_and_real_lemon_slices	0.909	0.657	0.179
jelly_beans	0.885	0.640	0.163
watercolors	0.908	0.699	0.074
real_and_fake_peppers	0.906	0.307	0.122
photo_and_face	0.614	0.083	0.625
face	0.903	0.248	0.162
flowers	0.914	0.367	0.136
oil_painting	0.908	0.764	0.135
fake_and_real_food	0.909	0.447	0.156
egyptian_statue	0.870	0.085	0.281
fake_and_real_sushi	0.889	0.340	0.235
feathers	0.881	0.532	0.146
fake_and_real_peppers	0.917	0.282	0.161
cloth	0.916	0.836	0.150

Table E.48: Results of FUSE with CAVE dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
balloons	0.901	0.250	0.120
chart_and_stuffed_toy	0.842	0.446	0.146
pompoms	0.864	0.697	0.123
superballs	0.901	0.645	0.205

Table E.48: Results of FUSE with CAVE dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
clay	0.910	0.418	0.162
cd	0.870	0.553	0.208
fake_and_real_tomatoes	0.711	0.108	0.605
fake_and_real_strawberries	0.875	0.479	0.158
sponges	0.845	0.407	0.096
real_and_fake_apples	0.920	0.273	0.171
hairs	0.431	0.339	0.546
paints	0.823	0.483	0.132
stuffed_toys	0.864	0.354	0.142
beads	0.852	0.701	0.240
fake_and_real_beers	0.945	0.507	0.062
fake_and_real_lemons	0.894	0.388	0.138
thread_spools	0.897	0.661	0.176
glass_tiles	0.865	0.738	0.215
fake_and_real_lemon_slices	0.887	0.649	0.204
jelly_beans	0.847	0.648	0.198
watercolors	0.878	0.699	0.087
real_and_fake_peppers	0.876	0.310	0.147
photo_and_face	0.925	0.325	0.161
face	0.878	0.254	0.188
flowers	0.890	0.385	0.159
oil_painting	0.887	0.760	0.155
fake_and_real_food	0.879	0.446	0.180
egyptian_statue	0.820	0.080	0.356
fake_and_real_sushi	0.852	0.321	0.304
feathers	0.852	0.546	0.173
fake_and_real_peppers	0.883	0.291	0.188
cloth	0.904	0.844	0.160

E.6.2 Harvard Dataset

Table E.49: Results of FUSE with Harvard dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgc4	0.991	0.577	0.035
imgb0	0.858	0.729	0.039
imgb1	0.985	0.725	0.027
imgc5	0.987	0.428	0.029

Table E.49: Results of FUSE with Harvard dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgb3	0.789	0.607	0.050
imgc7	0.984	0.675	0.041
imgc6	0.986	0.431	0.034
imgb2	0.946	0.687	0.046
imgb6	0.976	0.868	0.049
imgc2	0.623	0.522	0.037
imgc3	0.962	0.417	0.028
imgb7	0.919	0.562	0.061
imgc1	0.982	0.480	0.038
imgb5	0.988	0.575	0.036
imgb4	0.910	0.832	0.062
imga8	0.944	0.429	0.076
imgh0	0.984	0.491	0.044
imge7	0.963	0.816	0.030
imgd3	0.991	0.456	0.036
imgd2	0.971	0.494	0.039
imge6	0.991	0.820	0.036
imgh1	0.989	0.384	0.027
imgh3	0.969	0.644	0.045
imgd0	0.971	0.393	0.041
imge4	0.985	0.564	0.025
imgf8	0.988	0.711	0.081
imge5	0.978	0.698	0.027
imgd1	0.984	0.321	0.055
imgh2	0.992	0.456	0.061
imgh6	0.990	0.308	0.079
imgg9	0.996	0.308	0.079
imgd5	0.872	0.386	0.041
imge1	0.927	0.544	0.092
imge0	0.977	0.688	0.048
imgd4	0.987	0.407	0.026
imgg8	0.996	0.252	0.077
imgh7	0.996	0.419	0.082
imgh5	0.967	0.351	0.067
imge2	0.951	0.727	0.159
imgd6	0.995	0.429	0.076
imgd7	0.991	0.638	0.035
imge3	0.983	0.755	0.051

Table E.49: Results of FUSE with Harvard dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgh4	0.889	0.375	0.052
imgg6	0.947	0.313	0.097
imgf2	0.955	0.798	0.056
imgf3	0.883	0.682	0.060
imgg7	0.987	0.436	0.050
imgf1	0.836	0.468	0.041
imgg5	0.996	0.428	0.071
imgd9	0.956	0.410	0.033
imgd8	0.987	0.449	0.041
imgg4	0.994	0.354	0.103
imgf4	0.983	0.575	0.018
imgg0	0.955	0.493	0.074
imgg1	0.992	0.382	0.085
imgf5	0.974	0.716	0.051
imgg3	0.961	0.396	0.056
imgf7	0.899	0.736	0.047
imgf6	0.872	0.686	0.076
imgg2	0.988	0.347	0.066
img3	0.993	0.334	0.054
imga5	0.980	0.370	0.044
imgb9	0.981	0.574	0.032
imgb8	0.875	0.752	0.042
imga4	0.930	0.449	0.062
img2	0.993	0.788	0.044
imga6	0.981	0.695	0.039
imga7	0.986	0.754	0.026
img1	0.979	0.661	0.026
img5	0.995	0.332	0.060
imga3	0.799	0.601	0.058
imga2	0.990	0.331	0.041
img4	0.992	0.324	0.078
img6	0.991	0.318	0.062
imgc8	0.979	0.647	0.030
imga1	0.992	0.699	0.030
imgc9	0.978	0.794	0.046

Table E.50: Results of FUSE with Harvard dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgc4	0.989	0.576	0.041
imgb0	0.797	0.725	0.041
imgb1	0.985	0.725	0.027
imgc5	0.986	0.424	0.030
imgb3	0.640	0.611	0.048
imgc7	0.984	0.674	0.042
imgc6	0.985	0.430	0.035
imgb2	0.901	0.689	0.052
imgb6	0.961	0.869	0.053
imgc2	0.620	0.527	0.036
imgc3	0.956	0.418	0.028
imgb7	0.800	0.547	0.078
imgc1	0.981	0.475	0.041
imgb5	0.988	0.586	0.038
imgb4	0.713	0.823	0.074
imga8	0.924	0.426	0.090
imgh0	0.983	0.501	0.043
imge7	0.962	0.815	0.032
imgd3	0.989	0.455	0.040
imgd2	0.951	0.491	0.049
imge6	0.989	0.819	0.039
imgh1	0.992	0.382	0.027
imgh3	0.969	0.644	0.046
imgd0	0.954	0.398	0.044
imge4	0.984	0.564	0.027
imgf8	0.986	0.710	0.088
imge5	0.978	0.697	0.027
imgd1	0.958	0.294	0.061
imgh2	0.985	0.465	0.062
imgh6	0.990	0.309	0.079
imgg9	0.996	0.308	0.081
imgd5	0.913	0.386	0.045
imge1	0.922	0.540	0.104
imge0	0.969	0.684	0.054
imgd4	0.986	0.407	0.027
imgg8	0.994	0.247	0.081
imgh7	0.995	0.416	0.083
imgh5	0.957	0.351	0.069

Table E.50: Results of FUSE with Harvard dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imge2	0.942	0.724	0.176
imgd6	0.993	0.430	0.084
imgd7	0.990	0.638	0.038
imge3	0.983	0.768	0.052
imgh4	0.913	0.373	0.056
imgg6	0.941	0.308	0.098
imgf2	0.952	0.799	0.061
imgf3	0.844	0.682	0.072
imgg7	0.986	0.435	0.053
imgf1	0.724	0.464	0.053
imgg5	0.995	0.431	0.073
imgd9	0.949	0.413	0.034
imgd8	0.983	0.452	0.041
imgg4	0.993	0.349	0.107
imgf4	0.977	0.566	0.021
imgg0	0.945	0.497	0.075
imgg1	0.991	0.384	0.087
imgf5	0.965	0.717	0.058
imgg3	0.959	0.398	0.057
imgf7	0.806	0.736	0.047
imgf6	0.847	0.684	0.082
imgg2	0.986	0.346	0.069
img3	0.993	0.334	0.057
imga5	0.980	0.369	0.045
imgb9	0.980	0.572	0.033
imgb8	0.827	0.750	0.048
imga4	0.897	0.444	0.076
img2	0.974	0.786	0.047
imga6	0.981	0.701	0.038
imga7	0.985	0.754	0.027
img1	0.978	0.660	0.027
img5	0.995	0.332	0.060
imga3	0.718	0.595	0.086
imga2	0.987	0.287	0.048
img4	0.994	0.321	0.080
img6	0.991	0.316	0.063
imgc8	0.972	0.648	0.035
imga1	0.990	0.700	0.033

Table E.50: Results of FUSE with Harvard dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgc9	0.974	0.795	0.048

Table E.51: Results of FUSE with Harvard dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imgc4	0.980	0.572	0.050
imgb0	0.763	0.723	0.044
imgb1	0.984	0.723	0.029
imgc5	0.984	0.419	0.032
imgb3	0.769	0.602	0.058
imgc7	0.984	0.671	0.044
imgc6	0.984	0.431	0.037
imgb2	0.855	0.689	0.061
imgb6	0.915	0.868	0.054
imgc2	0.600	0.525	0.037
imgc3	0.947	0.416	0.029
imgb7	0.711	0.542	0.093
imgc1	0.978	0.466	0.047
imgb5	0.987	0.585	0.042
imgb4	0.611	0.816	0.081
imga8	0.914	0.425	0.098
imgh0	0.976	0.502	0.043
imge7	0.957	0.814	0.034
imgd3	0.982	0.456	0.048
imgd2	0.930	0.489	0.057
imge6	0.987	0.817	0.042
imgh1	0.984	0.384	0.029
imgh3	0.971	0.644	0.047
imgd0	0.936	0.401	0.049
imge4	0.983	0.563	0.028
imgf8	0.986	0.700	0.092
imge5	0.976	0.692	0.029
imgd1	0.954	0.328	0.060
imgh2	0.975	0.469	0.068
imgh6	0.989	0.307	0.080
imgg9	0.995	0.308	0.085
imgd5	0.935	0.384	0.049
imge1	0.921	0.538	0.111

Table E.51: Results of FUSE with Harvard dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imge0	0.975	0.683	0.058
imgd4	0.984	0.407	0.028
imgg8	0.986	0.203	0.094
imgh7	0.996	0.420	0.083
imgh5	0.948	0.350	0.074
imge2	0.936	0.722	0.184
imgd6	0.992	0.432	0.092
imgd7	0.987	0.637	0.046
imge3	0.975	0.755	0.060
imgh4	0.917	0.374	0.061
imgg6	0.925	0.304	0.098
imgf2	0.940	0.800	0.065
imgf3	0.823	0.680	0.081
imgg7	0.984	0.438	0.059
imgf1	0.621	0.453	0.075
imgg5	0.995	0.431	0.076
imgd9	0.950	0.401	0.035
imgd8	0.981	0.452	0.044
imgg4	0.990	0.344	0.114
imgf4	0.963	0.556	0.029
imgg0	0.936	0.498	0.079
imgg1	0.988	0.361	0.094
imgf5	0.953	0.717	0.069
imgg3	0.956	0.398	0.058
imgf7	0.688	0.724	0.057
imgf6	0.819	0.684	0.090
imgg2	0.983	0.346	0.074
img3	0.991	0.336	0.061
imga5	0.982	0.364	0.045
imgb9	0.979	0.569	0.035
imgb8	0.785	0.746	0.057
imga4	0.861	0.442	0.091
img2	0.972	0.788	0.049
imga6	0.980	0.696	0.040
imga7	0.985	0.754	0.027
img1	0.973	0.637	0.032
img5	0.994	0.329	0.062
imga3	0.614	0.588	0.118

Table E.51: Results of FUSE with Harvard dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
imga2	0.981	0.263	0.060
img4	0.994	0.283	0.086
img6	0.989	0.311	0.066
imgc8	0.963	0.648	0.041
imga1	0.989	0.700	0.036
imgc9	0.972	0.796	0.051

E.6.3 EHU Dataset

Table E.52: Results of FUSE with EHU dataset and a 4x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
KSC	0.709	0.239	0.943
Pavia	0.818	0.605	0.223
Botswana	0.807	0.247	0.386
PaviaU	0.719	0.549	0.271
SalinasA	0.776	0.342	0.286
Indian_pines	0.552	0.159	0.204
Salinas	0.677	0.193	0.372
Cuprite	0.756	0.430	-

Table E.53: Results of FUSE with EHU dataset and a 8x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
KSC	0.740	0.262	0.969
Pavia	0.782	0.604	0.261
Botswana	0.746	0.216	0.487
PaviaU	0.662	0.536	0.330
SalinasA	0.758	0.373	0.310
Indian_pines	0.530	0.189	0.214
Salinas	0.693	0.218	0.340
Cuprite	0.558	0.249	-

Table E.54: Results of FUSE with EHU dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
KSC	0.747	0.261	0.998
Pavia	0.745	0.586	0.303

Table E.54: Results of FUSE with EHU dataset and a 16x scaling factor.

Image	SSIM ↑	SCC ↑	SAM ↓
Botswana	0.707	0.223	0.549
PaviaU	0.687	0.535	0.317
SalinasA	0.763	0.350	0.322
Indian_pines	0.565	0.284	0.209
Salinas	0.664	0.220	0.389
Cuprite	0.582	0.373	-

Bibliography

[1] Elhadi Adam, Onisimo Mutanga, and Denis Rugege. Multispectral and hyperspectral remote sensing for identification and mapping of wetland vegetation: a review. *Wetlands Ecology and Management*, 18(3):281–296, 2010.

[2] Bruno Aiazzi, L Alparone, Stefano Baronti, Andrea Garzelli, and Massimo Selva. Mtf-tailored multiscale fusion of high-resolution ms and pan imagery. *Photogrammetric Engineering & Remote Sensing*, 72(5):591–596, 2006. URL https://openremotesensing.net/wp-content/uploads/2017/11/HSMSFusionToolbox.zip.

[3] Bruno Aiazzi, Stefano Baronti, and Massimo Selva. Improving component substitution pansharpening through multivariate regression of ms + pan data. *IEEE Transactions on Geoscience and Remote Sensing*, 45(10):3230–3239, 2007. URL https://openremotesensing.net/wp-content/uploads/2017/11/HSMSFusionToolbox.zip.

[4] Naveed Akhtar, Faisal Sahfait, and Ajmal Mian. Repeated constrained sparse coding with partial dictionaries for hyperspectral unmixing. In *IEEE Winter Conference on Applications of Computer Vision*, pages 953–960. IEEE, 2014.

[5] Naveed Akhtar, Faisal Shafait, and Ajmal Mian. Sparse spatio-spectral representation for hyperspectral image super-resolution. In *European conference on computer vision*, pages 63–78. Springer, 2014. URL http://staffhome.ecm.uwa.edu.au/~00053650/code.html.

[6] Naveed Akhtar, Faisal Shafait, and Ajmal Mian. Bayesian sparse representation for hyperspectral image super resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3631–3640, 2015. URL https://github.com/magamig/hif-benchmarking/tree/main/methods/BayesianSparse.

[7] Luciano Alparone, Lucien Wald, Jocelyn Chanussot, Claire Thomas, Paolo Gamba, and Lori Mann Bruce. Comparison of pansharpening algorithms: Out-

come of the 2006 grs-s data-fusion contest. *IEEE Transactions on Geoscience and Remote Sensing*, 45(10):3012–3021, 2007.

[8] B Arad, O Ben-Shahar, R NTIRE Timofte, L Van Gool, L Zhang, and M NTIRE Yang. challenge on spectral reconstruction from rgb images. In *Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA*, pages 18–22, 2018.

[9] Boaz Arad and Ohad Ben-Shahar. Sparse recovery of hyperspectral signal from natural rgb images. In *European Conference on Computer Vision*, pages 19–34. Springer, 2016.

[10] Marion F Baumgardner, Larry L Biehl, and David A Landgrebe. 220 band aviris hyperspectral image data set: June 12, 1992 indian pine test site 3. *Purdue University Research Repository*, 10:R7RX991C, 2015.

[11] Olivier Berné, A Helens, Paolo Pilleri, and Christine Joblin. Non-negative matrix factorization pansharpening of hyperspectral data: An application to mid-infrared astronomy. In *2010 2nd Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing*, pages 1–4. IEEE, 2010.

[12] Roy S Berns. The science of digitizing paintings for color-accurate image archives: a review. *Journal of imaging science and technology*, 45(4):305–325, 2001.

[13] José M Bioucas-Dias, Antonio Plaza, Nicolas Dobigeon, Mario Parente, Qian Du, Paul Gader, and Jocelyn Chanussot. Hyperspectral unmixing overview: Geometrical, statistical, and sparse regression-based approaches. *IEEE journal of selected topics in applied earth observations and remote sensing*, 5 (2):354–379, 2012.

[14] José M Bioucas-Dias, Antonio Plaza, Gustavo Camps-Valls, Paul Scheunders, Nasser Nasrabadi, and Jocelyn Chanussot. Hyperspectral remote sensing data analysis and future challenges. *IEEE Geoscience and remote sensing magazine*, 1(2):6–36, 2013.

[15] Ricardo Augusto Borsoi, Tales Imbiriba, and José Carlos Moreira Bermudez. Super-resolution for hyperspectral and multispectral image fusion accounting for seasonal spectral variability. *IEEE transactions on image processing*, 29:116–127, 2019.

[16] Leon Bungert, David A Coomes, Matthias J Ehrhardt, Jennifer Rasch, Rafael Reisenhofer, and Carola-Bibiane Schönlieb. Blind image fusion for hyperspectral imaging with the directional total variation. *Inverse Problems*, 34(4): 044003, 2018. URL https://github.com/mehrhardt/blind_remote_sensing.

[17] Wjoseph Carper, Thomasm Lillesand, and Ralphw Kiefer. The use of intensity-hue-saturation transformations for merging spot panchromatic and multispectral image data. *Photogrammetric Engineering and remote sensing*, 56(4): 459–467, 1990.

[18] Mufit Cetin and Nebiye Musaoglu. Merging hyperspectral and panchromatic image data: qualitative and quantitative analysis. *International Journal of Remote Sensing*, 30(7):1779–1804, 2009.

[19] Ayan Chakrabarti and Todd Zickler. Statistics of real-world hyperspectral images. In *CVPR 2011*, pages 193–200. IEEE, 2011.

[20] Chein-I Chang. An information-theoretic approach to spectral variability, similarity, and discrimination for hyperspectral image analysis. *IEEE Transactions on information theory*, 46(5):1927–1932, 2000.

[21] Yi Chang, Luxin Yan, Houzhang Fang, Sheng Zhong, and Wenshan Liao. Hsidenet: Hyperspectral image restoration via convolutional neural network. *IEEE Transactions on Geoscience and Remote Sensing*, 57(2):667–682, 2018.

[22] Wenjing Chen, Xiangtao Zheng, and Xiaoqiang Lu. Hyperspectral image super-resolution with self-supervised spectral-spatial residual network. *Remote Sensing*, 13(7):1260, 2021.

[23] Zhao Chen, Hanye Pu, Bin Wang, and Geng-Ming Jiang. Fusion of hyperspectral and multispectral images: A novel framework based on generalization of pan-sharpening methods. *IEEE Geoscience and Remote Sensing Letters*, 11(8): 1418–1422, 2014.

[24] Alex Clark and Fredrik Lundh. Pillow (pil fork) documentation. *readthedocs*, 2022. URL https://pillow.readthedocs.io/.

[25] Costanza Cucci and Andrea Casini. Hyperspectral imaging for artworks investigation. In *Data Handling in Science and Technology*, volume 32, pages 583–604. Elsevier, 2020.

[26] Renwei Dian and Shutao Li. Hyperspectral image super-resolution via subspace-based low tensor multi-rank regularization. *IEEE Transactions on Image Processing*, 28(10):5135–5146, 2019. URL https://github.com/renweidian/LTMR.

[27] Renwei Dian, Leyuan Fang, and Shutao Li. Hyperspectral image super-resolution via non-local sparse tensor factorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5344–5353, 2017.

[28] Renwei Dian, Shutao Li, Anjing Guo, and Leyuan Fang. Deep hyperspectral image sharpening. *IEEE transactions on neural networks and learning systems*, 29(11):5345–5355, 2018. URL https://github.com/renweidian/DHSIS.

[29] Renwei Dian, Shutao Li, and Leyuan Fang. Learning a low tensor-train rank representation for hyperspectral image super-resolution. *IEEE transactions on neural networks and learning systems*, 30(9):2672–2683, 2019. URL https://github.com/renweidian/LTTR.

[30] Renwei Dian, Shutao Li, and Xudong Kang. Regularizing hyperspectral and multispectral image fusion by cnn denoiser. *IEEE transactions on neural networks and learning systems*, 32(3):1124–1135, 2020. URL https://github.com/renweidian/CNN-FUS.

[31] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

[32] Weisheng Dong, Fazuo Fu, Guangming Shi, Xun Cao, Jinjian Wu, Guangyu Li, and Xin Li. Hyperspectral image super-resolution via non-negative structured sparse representation. *IEEE Transactions on Image Processing*, 25(5):2337–2352, 2016. URL https://see.xidian.edu.cn/faculty/wsdong/Code_release/NSSR_HSI_SR.rar.

[33] Weisheng Dong, Chen Zhou, Fangfang Wu, Jinjian Wu, Guangming Shi, and Xin Li. Model-guided deep hyperspectral image super-resolution. *IEEE Transactions on Image Processing*, 30:5754–5768, 2021. URL https://see.xidian.edu.cn/faculty/wsdong/Projects/MoG-DCN.htm.

[34] Claude E Duchon. Lanczos filtering in one and two dimensions. *Journal of Applied Meteorology and Climatology*, 18(8):1016–1022, 1979.

[35] Michael Theodore Eismann. *Resolution enhancement of hyperspectral imagery using maximum a posteriori estimation with a stochastic mixing model*. University of Dayton, 2004. URL https://openremotesensing.net/wp-content/uploads/2017/11/HSMSFusionToolbox.zip.

[36] João Fortuna, Harald Martens, and Tor Arne Johansen. Multivariate image fusion: A pipeline for hyperspectral data enhancement. *Chemometrics and Intelligent Laboratory Systems*, 205:104097, 2020.

[37] Ying Fu, Yinqiang Zheng, Hua Huang, Imari Sato, and Yoichi Sato. Hyperspectral image super-resolution with a mosaic rgb image. *IEEE Transactions on Image Processing*, 27(11):5539–5552, 2018.

[38] Ying Fu, Tao Zhang, Yinqiang Zheng, Debing Zhang, and Hua Huang. Hyperspectral image super-resolution with optimized rgb guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11661–11670, 2019. URL https://github.com/ColinTaoZhang/HSI-SR.

[39] Han Gao, Guifeng Zhang, and Min Huang. Hyperspectral image superresolution via structure-tensor-based image matting. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:7994–8007, 2021.

[40] Andrea Garzelli and Filippo Nencini. Hypercomplex quality assessment of multi/hyperspectral images. *IEEE Geoscience and Remote Sensing Letters*, 6 (4):662–665, 2009.

[41] Richard B Gomez, Amin Jazaeri, and Menas Kafatos. Wavelet-based hyperspectral and multispectral image fusion. In *Geo-Spatial Image and Data Exploitation II*, volume 4383, pages 36–42. International Society for Optics and Photonics, 2001.

[42] María González-Audícana, José Luis Saleta, Raquel García Catalán, and Rafael García. Fusion of multispectral and panchromatic images using improved ihs and pca mergers based on wavelet decomposition. *IEEE Transactions on Geoscience and Remote sensing*, 42(6):1291–1299, 2004.

[43] Aoife A Gowen, Colm P O'Donnell, Patrick J Cullen, Gérard Downey, and Jesus M Frias. Hyperspectral imaging–an emerging process analytical tool for food quality and safety control. *Trends in food science & technology*, 18(12): 590–598, 2007.

[44] Claas Grohnfeldt, Xiao Xiang Zhu, and Richard Bamler. Splitting the hyperspectral-multispectral image fusion problem autonomously into weighted pan-sharpening tasks—the spectral grouping concept. In *2015 7th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–4. IEEE, 2015.

[45] Xian-Hua Han, Boxin Shi, and Yinqiang Zheng. Self-similarity constrained sparse representation for hyperspectral image super-resolution. *IEEE Transactions on Image Processing*, 27(11):5625–5637, 2018.

[46] Xian-Hua Han, Boxin Shi, and Yinqiang Zheng. Ssf-cnn: Spatial and spectral fusion with cnn for hyperspectral image super-resolution. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2506–2510. IEEE, 2018. URL https://github.com/hw2hwei/SSRNET/blob/master/models/SSFCNN.py.

[47] Xian-Hua Han, YinQiang Zheng, and Yen-Wei Chen. Multi-level and multi-scale spatial and spectral fusion cnn for hyperspectral image super-resolution.

In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[48] Renlong Hang, Qingshan Liu, and Zhu Li. Spectral super-resolution network guided by intrinsic properties of hyperspectral imagery. *IEEE Transactions on Image Processing*, 30:7256–7265, 2021.

[49] Russell C Hardie, Michael T Eismann, and Gregory L Wilson. Map estimation for hyperspectral image resolution enhancement using an auxiliary sensor. *IEEE Transactions on Image Processing*, 13(9):1174–1184, 2004.

[50] Wei He, Yong Chen, Naoto Yokoya, Chao Li, and Qibin Zhao. Hyperspectral super-resolution via coupled tensor ring factorization. *Pattern Recognition*, 122:108280, 2022.

[51] Jin-Fan Hu, Ting-Zhu Huang, and Liang-Jian Deng. Fusformer: A transformer-based fusion approach for hyperspectral image super-resolution. *arXiv preprint arXiv:2109.02079*, 2021. URL https://github.com/J-FHu/Fusformer.

[52] Jin-Fan Hu, Ting-Zhu Huang, Liang-Jian Deng, Tai-Xiang Jiang, Gemine Vivone, and Jocelyn Chanussot. Hyperspectral image super-resolution via deep spatiospectral attention convolutional neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2021. URL https://liangjiandeng.github.io/Projects_Res/HSRnet_2021tnnls.html.

[53] Yu Hen Hu, HB Lee, and FL Scarpace. Optimal linear spectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 37(1):639–644, 1999.

[54] Bo Huang, Huihui Song, Hengbin Cui, Jigen Peng, and Zongben Xu. Spatial and spectral image fusion using sparse matrix factorization. *IEEE Transactions on Geoscience and Remote Sensing*, 52(3):1693–1704, 2013.

[55] Leping Huang, Zhongwen Hu, Xin Luo, Qian Zhang, Jingzhe Wang, and Guofeng Wu. Stepwise fusion of hyperspectral, multispectral and panchromatic images with spectral grouping strategy: A comparative study using gf5 and gf1 images. *Remote Sensing*, 14(4):1021, 2022.

[56] Tao Huang, Weisheng Dong, Jinjian Wu, Leida Li, Xin Li, and Guangming Shi. Deep hyperspectral image fusion network with iterative spatio-spectral regularization. *IEEE Transactions on Computational Imaging*, 8:201–214, 2022. URL https://see.xidian.edu.cn/faculty/wsdong/Projects/TCI2022-DHIF-Net/DHIF-Net.htm.

[57] Francisco H Imai and Roy S Berns. High-resolution multi-spectral image archives: a hybrid approach. In *Color and imaging conference*, 1, pages 224–227. Society for Imaging Science and Technology, 1998.

[58] P Jagalingam and Arkal Vittal Hegde. A review of quality metrics for fused image. *Aquatic Procedia*, 4:133–142, 2015.

[59] Manjunath Joshi and André Jalobeanu. Map estimation for multiresolution fusion in remotely sensed images using an igmrf prior model. *IEEE Transactions on Geoscience and Remote Sensing*, 48(3):1245–1255, 2009.

[60] Peter K Kaiser and Robert M Boynton. *Human color vision*. Optical Society of America, 1996.

[61] Charilaos I Kanatsoulis, Xiao Fu, Nicholas D Sidiropoulos, and Wing-Kin Ma. Hyperspectral super-resolution: A coupled tensor factorization approach. *IEEE Transactions on Signal Processing*, 66(24):6503–6517, 2018. URL https://github.com/marhar19/HSR_via_tensor_decomposition.

[62] Rei Kawakami, Yasuyuki Matsushita, John Wright, Moshe Ben-Ezra, Yu-Wing Tai, and Katsushi Ikeuchi. High-resolution hyperspectral imaging via matrix factorization. In *CVPR 2011*, pages 2329–2336. IEEE, 2011. URL http://www.nae-lab.org/~rei/research/hh/index.html.

[63] Zohaib Khan, Faisal Shafait, and Ajmal Mian. Hyperspectral imaging for ink mismatch detection. In *2013 12th International Conference on Document Analysis and Recognition*, pages 877–881. IEEE, 2013.

[64] Nikos Koutsias, Michael Karteris, and Emilio Chuvico. The use of intensity-hue-saturation transformation of landsat-5 thematic mapper data for burned land mapping. *Photogrammetric Engineering and Remote Sensing*, 66(7):829–840, 2000.

[65] P Kwarteng and A Chavez. Extracting spectral contrast in landsat thematic mapper image data using selective principal component analysis. *Photogramm. Eng. Remote Sens*, 55(1):339–348, 1989.

[66] Hyeokhyen Kwon and Yu-Wing Tai. Rgb-guided hyperspectral image upsampling. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 307–315, 2015.

[67] Charis Lanaras, Emmanuel Baltsavias, and Konrad Schindler. Hyperspectral super-resolution by coupled spectral unmixing. In *Proceedings of the IEEE international conference on computer vision*, pages 3586–3594, 2015. URL https://github.com/lanha/SupResPALM.

[68] Jiaxin Li, Ke Zheng, Jing Yao, Lianru Gao, and Danfeng Hong. Deep unsupervised blind hyperspectral and multispectral data fusion. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022. URL https://github.com/JiaxinLiCAS/UDALN_GRSL.

[69] Ke Li, Dengxin Dai, and Luc Van Gool. Hyperspectral image super-resolution with rgb image super-resolution as an auxiliary task. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3193–3202, 2022. URL https://github.com/kli8996/HSISR.

[70] Shutao Li, Renwei Dian, Leyuan Fang, and José M Bioucas-Dias. Fusing hyperspectral and multispectral images via coupled sparse tensor factorization. *IEEE Transactions on Image Processing*, 27(8):4118–4130, 2018. URL https://drive.google.com/open?id=12eleEjv7wKQxFCBUcIGkEl-wiUiJxwTv.

[71] Xuesong Li, Youqiang Zhang, Zixian Ge, Guo Cao, Hao Shi, and Peng Fu. Adaptive nonnegative sparse representation for hyperspectral image super-resolution. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:4267–4283, 2021.

[72] JG Liu. Smoothing filter-based intensity modulation: A spectral preserve image fusion technique for improving spatial details. *International Journal of Remote Sensing*, 21(18):3461–3472, 2000. URL https://openremotesensing.net/wp-content/uploads/2017/11/HSMSFusionToolbox.zip.

[73] Jianjun Liu, Zebin Wu, Liang Xiao, Jun Sun, and Hong Yan. A truncated matrix decomposition for hyperspectral image super-resolution. *IEEE Transactions on Image Processing*, 29:8028–8042, 2020.

[74] Jianjun Liu, Dunbin Shen, Zebin Wu, Liang Xiao, Jun Sun, and Hong Yan. Patch-aware deep hyperspectral and multispectral image fusion by unfolding subspace-based optimization model. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2022. URL https://github.com/liuofficial/SpfNet.

[75] Jianjun Liu, Zebin Wu, Liang Xiao, and Xiao-Jun Wu. Model inspired autoencoder for unsupervised hyperspectral image super-resolution. *IEEE Transactions on Geoscience and Remote Sensing*, 2022. URL https://github.com/liuofficial/MIAE/.

[76] Xiangyu Liu, Qingjie Liu, and Yunhong Wang. Remote sensing image fusion based on two-stream fusion network. *Information Fusion*, 55:1–15, 2020. URL https://github.com/hw2hwei/SSRNET/blob/master/models/TFNet.py.

[77] Zhe Liu, Yinqiang Zheng, and Xian-Hua Han. Deep unsupervised fusion learning for hyperspectral image super resolution. *Sensors*, 21(7):2348, 2021.

[78] Laetitia Loncan, Luis B De Almeida, José M Bioucas-Dias, Xavier Briottet, Jocelyn Chanussot, Nicolas Dobigeon, Sophie Fabre, Wenzhi Liao, Giorgio A Licciardi, Miguel Simoes, et al. Hyperspectral pansharpening: A review. *IEEE Geoscience and remote sensing magazine*, 3(3):27–46, 2015.

[79] Guolan Lu and Baowei Fei. Medical hyperspectral imaging: a review. *Journal of biomedical optics*, 19(1):010901, 2014.

[80] Ruiying Lu, Bo Chen, Ziheng Cheng, and Penghui Wang. Rafnet: Recurrent attention fusion network of hyperspectral and multispectral images. *Signal Processing*, 177:107737, 2020. URL `https://github.com/RuiyingLu/RAFnet`.

[81] Qing Ma, Junjun Jiang, Xianming Liu, and Jiayi Ma. Learning a 3d-cnn and transformer prior for hyperspectral image super-resolution. *arXiv preprint arXiv:2111.13923*, 2021.

[82] Dimitris Manolakis, David Marden, Gary A Shaw, et al. Hyperspectral image processing for automatic target detection applications. *Lincoln laboratory journal*, 14(1):79–116, 2003.

[83] Shaohui Mei, Xin Yuan, Jingyu Ji, Yifan Zhang, Shuai Wan, and Qian Du. Hyperspectral image spatial super-resolution via 3d full convolutional neural network. *Remote Sensing*, 9(11):1139, 2017.

[84] Audrey Minghelli-Roman, Marc Mangolini, Michel Petit, and Laurent Polidori. Spatial resolution improvement of meris images by fusion with tm images. *IEEE Transactions on Geoscience and Remote Sensing*, 39(7):1533–1536, 2001.

[85] Audrey Minghelli-Roman, Laurent Polidori, Sandrine Mathieu-Blanc, Lionel Loubersac, and François Cauneau. Spatial resolution improvement by merging meris-etm images for coastal water monitoring. *IEEE Geoscience and Remote Sensing Letters*, 3(2):227–231, 2006.

[86] Curtis K Munechika. Resolution enhancement of multispectral image data to improve classification accuracy. *Photogrammetric Engineering and Remote Sensing*, 59(1):67–72, 1993.

[87] Rang MH Nguyen, Dilip K Prasad, and Michael S Brown. Training-based spectral reconstruction from a single rgb image. In *European Conference on Computer Vision*, pages 186–201. Springer, 2014.

[88] Jiangtao Nie, Lei Zhang, Wei Wei, Chen Ding, and Yanning Zhang. Unsupervised deep hyperspectral super-resolution with unregistered images. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2020.

[89] Peter Orbanz. Lecture notes on bayesian nonparametrics. *Journal of Mathematical Psychology*, 56:1–12, 2012.

[90] Frosti Palsson, Johannes R Sveinsson, and Magnus O Ulfarsson. Multispectral and hyperspectral image fusion using a 3-d-convolutional neural network. *IEEE Geoscience and Remote Sensing Letters*, 14(5):639–643, 2017.

[91] Zhi-Wei Pan and Hui-Liang Shen. Multispectral image super-resolution via rgb image fusion and radiometric calibration. *IEEE Transactions on Image Processing*, 28(4):1783–1797, 2018.

[92] Yidong Peng, Weisheng Li, Xiaobo Luo, and Jiao Du. Hyperspectral image superresolution using global gradient sparse and nonlocal low-rank tensor decomposition with hyper-laplacian prior. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:5453–5469, 2021.

[93] Ying Qu, Hairong Qi, and Chiman Kwan. Unsupervised sparse dirichlet-net for hyperspectral image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2511–2520, 2018. URL https://github.com/aicip/uSDN.

[94] Ying Qu, Hairong Qi, Chiman Kwan, Naoto Yokoya, and Jocelyn Chanussot. Unsupervised and unregistered hyperspectral image super-resolution with mutual dirichlet-net. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–18, 2021. URL https://github.com/yingutk/u2MDN.

[95] Rizwan Qureshi, Muhammad Uzair, Khurram Khurshid, and Hong Yan. Hyperspectral document image processing: Applications, challenges and future prospects. *Pattern Recognition*, 90:12–22, 2019.

[96] Thierry Ranchin and Lucien Wald. Fusion of high spatial and spectral resolution images: The arsis concept and its implementation. *Photogrammetric engineering and remote sensing*, 66(1):49–61, 2000.

[97] Massimo Selva, Bruno Aiazzi, Francesco Butera, Leandro Chiarantini, and Stefano Baronti. Hyper-sharpening of hyperspectral data: A first approach. In *2014 6th Workshop on Hyperspectral Image and Signal Processing: Evolution in Remote Sensing (WHISPERS)*, pages 1–4. IEEE, 2014.

[98] Massimo Selva, Bruno Aiazzi, Francesco Butera, Leandro Chiarantini, and Stefano Baronti. Hyper-sharpening: A first approach on sim-ga data. *IEEE Journal of selected topics in applied earth observations and remote sensing*, 8 (6):3008–3024, 2015.

[99] Hamid R Sheikh and Alan C Bovik. Image information and visual quality. *IEEE Transactions on image processing*, 15(2):430–444, 2006.

[100] Dunbin Shen, Jianjun Liu, Zhiyong Xiao, Jinlong Yang, and Liang Xiao. A twice optimizing net with matrix decomposition for hyperspectral and multispectral image fusion. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13:4095–4110, 2020. URL https://github.com/liuofficial/TONWMD.

[101] Dunbin Shen, Jianjun Liu, Zebin Wu, Jinlong Yang, and Liang Xiao. Admm-hfnet: A matrix decomposition-based deep approach for hyperspectral image fusion. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–17, 2021. URL https://github.com/liuofficial/ADMM-HFNet.

[102] Raju Shrestha, Ruven Pillay, Sony George, and Jon Yngve Hardeberg. Quality evaluation in spectral imaging–quality factors and metrics. *Journal of the International Colour Association*, 2014.

[103] Oleksii Sidorov and Jon Yngve Hardeberg. Deep hyperspectral prior: Single-image denoising, inpainting, super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019. URL https://github.com/acecreamu/deep-hs-prior.

[104] Miguel Simoes, José Bioucas-Dias, Luis B Almeida, and Jocelyn Chanussot. Hyperspectral image superresolution: An edge-preserving convex formulation. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 4166–4170. IEEE, 2014. URL https://github.com/alfaiate/HySure.

[105] Ahmad AL Smadi, Shuyuan Yang, Atif Mehmood, Ahed Abugabah, Min Wang, and Muzaffar Bashir. Smart pansharpening approach using kernel-based image filtering. *IET Image Processing*, 2021.

[106] Weiwei Sun, Kai Ren, Xiangchao Meng, Chenchao Xiao, Gang Yang, and Jiangtao Peng. A band divide-and-conquer multispectral and hyperspectral image fusion method. *IEEE Transactions on Geoscience and Remote Sensing*, 60: 1–13, 2021.

[107] Yuliya Tarabalka, Jocelyn Chanussot, and Jón Atli Benediktsson. Segmentation and classification of hyperspectral images using minimum spanning forest grown from automatically selected markers. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 40(5):1267–1279, 2009.

[108] Xin Tian, Wei Zhang, Yuerong Chen, Zhongyuan Wang, and Jiayi Ma. Hyperfusion: A computational approach for hyperspectral, multispectral, and panchromatic image fusion. *IEEE Transactions on Geoscience and Remote Sensing*, 2021. URL https://github.com/saber-zero/HyperFusion.

[109] Ken Turkowski. Filters for common resampling tasks. *Graphics gems*, pages 147–165, 1990.

[110] Tatsumi Uezato, Danfeng Hong, Naoto Yokoya, and Wei He. Guided deep decoder: Unsupervised image pair fusion. In *European Conference on Computer Vision*, pages 87–102. Springer, 2020. URL https://github.com/tuezato/guided-deep-decoder.

[111] Muhammad Uzair, Arif Mahmood, and Ajmal S Mian. Hyperspectral face recognition using 3d-dct and partial least squares. In *BMVC*, volume 1, page 10, 2013.

[112] F Van Der Meer. Iterative spectral unmixing (isu). *International Journal of Remote Sensing*, 20(17):3431–3436, 1999.

[113] Hien Van Nguyen, Amit Banerjee, and Rama Chellappa. Tracking via object reflectance using a hyperspectral video camera. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, pages 44–51. IEEE, 2010.

[114] Marija Vella, Bowen Zhang, Wei Chen, and João FC Mota. Enhanced hyperspectral image super-resolution via rgb fusion and tv-tv minimization. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 3837–3841. IEEE, 2021. URL https://github.com/marijavella/hs-sr-tvtv.

[115] Lucien Wald. Quality of high resolution synthesised images: Is there a simple criterion? In *Third conference" Fusion of Earth data: merging point measurements, raster maps and remotely sensed images"*, pages 99–103. SEE/URISCA, 2000.

[116] Lucien Wald, Thierry Ranchin, and Marc Mangolini. Fusion of satellite images of different spatial resolutions: Assessing the quality of resulting images. *Photogrammetric engineering and remote sensing*, 63(6):691–699, 1997.

[117] Gregory K Wallace. The jpeg still picture compression standard. *IEEE transactions on consumer electronics*, 38(1):xviii–xxxiv, 1992.

[118] Wei Wan, Weihong Guo, Haiyang Huang, and Jun Liu. Nonnegative and nonlocal sparse tensor factorization-based hyperspectral image super-resolution. *IEEE Transactions on Geoscience and Remote Sensing*, 58(12):8384–8394, 2020.

[119] Chen Wang, Yun Liu, Xiao Bai, Wenzhong Tang, Peng Lei, and Jun Zhou. Deep residual convolutional neural network for hyperspectral image super-resolution. In *International conference on image and graphics*, pages 370–380. Springer, 2017.

[120] Wu Wang, Weihong Zeng, Yue Huang, Xinghao Ding, and John Paisley. Deep blind hyperspectral image fusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4150–4159, 2019. URL https://github.com/wwhappylife/Deep-Blind-Hyperspectral-Image-Fusion.

[121] Xiuheng Wang, Jie Chen, Qi Wei, and Cédric Richard. Hyperspectral image super-resolution via deep prior regularization with parameter estimation. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021. URL

https://github.com/xiuheng-wang/Sylvester_TSFN_MDC_HSI_superresolution.

[122] Xiuheng Wang, Jie Chen, and Cédric Richard. Hyperspectral image super-resolution with deep priors and degradation model inversion. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2814–2818. IEEE, 2022. URL https://github.com/xiuheng-wang/Deep_gradient_HSI_superresolution.

[123] Yao Wang, Xi'ai Chen, Zhi Han, Shiying He, et al. Hyperspectral image super-resolution via nonlocal low-rank tensor approximation and total variation regularization. *Remote Sensing*, 9(12):1286, 2017.

[124] Zhijun Wang, Djemel Ziou, Costas Armenakis, Deren Li, and Qingquan Li. A comparative analysis of image fusion methods. *IEEE transactions on geoscience and remote sensing*, 43(6):1391–1402, 2005.

[125] Zhou Wang and Alan C Bovik. A universal image quality index. *IEEE signal processing letters*, 9(3):81–84, 2002.

[126] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003.

[127] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[128] Qi Wei, Nicolas Dobigeon, and Jean-Yves Tourneret. Bayesian fusion of multispectral and hyperspectral images with unknown sensor spectral response. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 698–702. IEEE, 2014.

[129] Qi Wei, José Bioucas-Dias, Nicolas Dobigeon, and Jean-Yves Tourneret. Hyperspectral and multispectral image fusion based on a sparse representation. *IEEE Transactions on Geoscience and Remote Sensing*, 53(7):3658–3668, 2015. URL http://wei.perso.enseeiht.fr/demo/SparseFusion_2014-12-03.zip.

[130] Qi Wei, Nicolas Dobigeon, and Jean-Yves Tourneret. Fast fusion of multi-band images based on solving a sylvester equation. *IEEE Transactions on Image Processing*, 24(11):4109–4121, 2015.

[131] Qi Wei, José Bioucas-Dias, Nicolas Dobigeon, Jean-Yves Tourneret, Marcus Chen, and Simon Godsill. Multiband image fusion based on spectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 54(12):7236–7249, 2016. URL https://github.com/qw245/FUMI.

[132] Qi Wei, José Bioucas-Dias, Nicolas Dobigeon, Jean-Yves Tourneret, and Simon Godsill. Blind model-based fusion of multi-band and panchromatic images. In *2016 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pages 21–25. IEEE, 2016. URL https://github.com/qw245/BlindFuse.

[133] Wei Wei, Jiangtao Nie, Yong Li, Lei Zhang, and Yanning Zhang. Deep recursive network for hyperspectral image super-resolution. *IEEE Transactions on Computational Imaging*, 6:1233–1244, 2020.

[134] Wei Wei, Jiangtao Nie, Lei Zhang, and Yanning Zhang. Unsupervised recurrent hyperspectral imagery super-resolution using pixel-aware refinement. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–15, 2020. URL https://github.com/JiangtaoNie/Rec_HSISR_PixAwaRefin.

[135] Eliot Wycoff, Tsung-Han Chan, Kui Jia, Wing-Kin Ma, and Yi Ma. A non-negative sparse promoting algorithm for high resolution hyperspectral imaging. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1409–1413. IEEE, 2013. URL https://mx.nthu.edu.tw/~tsunghan/download/SNNMF.rar.

[136] Qi Xie, Minghao Zhou, Qian Zhao, Deyu Meng, Wangmeng Zuo, and Zongben Xu. Multispectral and hyperspectral image fusion by ms/hs fusion net. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1585–1594, 2019. URL https://github.com/XieQi2015/MHF-net.

[137] Yang Xu, Zebin Wu, Jocelyn Chanussot, and Zhihui Wei. Nonlocal patch tensor sparse representation for hyperspectral image super-resolution. *IEEE Transactions on Image Processing*, 28(6):3034–3047, 2019.

[138] Yang Xu, Zebin Wu, Jocelyn Chanussot, and Zhihui Wei. Hyperspectral images super-resolution via learning high-order coupled tensor ring representation. *IEEE transactions on neural networks and learning systems*, 31(11):4747–4760, 2020.

[139] Jize Xue, Yong-Qiang Zhao, Yuanyang Bu, Wenzhi Liao, Jonathan Cheung-Wai Chan, and Wilfried Philips. Spatial-spectral structured sparse low-rank representation for hyperspectral image super-resolution. *IEEE Transactions on Image Processing*, 30:3084–3097, 2021.

[140] Jingxiang Yang, Yong-Qiang Zhao, and Jonathan Cheung-Wai Chan. Hyperspectral and multispectral image fusion via deep two-branches convolutional neural network. *Remote Sensing*, 10(5):800, 2018. URL https://github.com/polwork/Hyperspectral-and-Multispectral-fusion-via-Two-branch-CNN.

[141] Jing Yao, Danfeng Hong, Jocelyn Chanussot, Deyu Meng, Xiaoxiang Zhu, and Zongben Xu. Cross-attention in coupled unmixing nets for unsupervised hyperspectral super-resolution. In *European Conference on Computer Vision*, pages 208–224. Springer, 2020. URL https://github.com/danfenghong/ECCV2020_CUCaNet.

[142] Fumihito Yasuma, Tomoo Mitsunaga, Daisuke Iso, and Shree K Nayar. Generalized assorted pixel camera: postcapture control of resolution, dynamic range, and spectrum. *IEEE transactions on image processing*, 19(9):2241–2253, 2010.

[143] Chen Yi, Yong-Qiang Zhao, and Jonathan Cheung-Wai Chan. Hyperspectral image super-resolution based on spatial and spectral correlation fusion. *IEEE Transactions on Geoscience and Remote Sensing*, 56(7):4165–4177, 2018.

[144] Changhoon Yim and Alan Conrad Bovik. Quality assessment of deblocked images. *IEEE Transactions on Image Processing*, 20(1):88–98, 2010.

[145] Naoto Yokoya and Akira Iwasaki. Airborne hyperspectral data over chikusei. *Space Appl. Lab., Univ. Tokyo, Tokyo, Japan, Tech. Rep. SAL-2016-05-27*, 2016.

[146] Naoto Yokoya, Takehisa Yairi, and Akira Iwasaki. Coupled non-negative matrix factorization (cnmf) for hyperspectral and multispectral data fusion: Application to pasture classification. In *2011 IEEE International Geoscience and Remote Sensing Symposium*, pages 1779–1782. IEEE, 2011. URL https://naotoyokoya.com/assets/zip/CNMF_MATLAB.zip.

[147] Naoto Yokoya, Claas Grohnfeldt, and Jocelyn Chanussot. Hyperspectral and multispectral data fusion: A comparative review of the recent literature. *IEEE Geoscience and Remote Sensing Magazine*, 5(2):29–56, 2017.

[148] Qiangqiang Yuan, Yancong Wei, Xiangchao Meng, Huanfeng Shen, and Liangpei Zhang. A multiscale and multidepth convolutional neural network for remote sensing imagery pan-sharpening. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(3):978–989, 2018. URL https://github.com/hw2hwei/SSRNET/blob/master/models/MSDCNN.py.

[149] Yuan Yuan, Xiangtao Zheng, and Xiaoqiang Lu. Hyperspectral image superresolution by transfer learning. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 10(5):1963–1974, 2017.

[150] Roberta H Yuhas, Alexander FH Goetz, and Joe W Boardman. Discrimination among semi-arid landscape endmembers using the spectral angle mapper (sam) algorithm. In *JPL, Summaries of the Third Annual JPL Airborne Geoscience Workshop. Volume 1: AVIRIS Workshop*, 1992.

[151] Kai Zhang, Min Wang, and Shuyuan Yang. Multispectral and hyperspectral image fusion based on group spectral embedding and low-rank factorization. *IEEE Transactions on Geoscience and Remote Sensing*, 55(3):1363–1371, 2016.

[152] Kai Zhang, Min Wang, Shuyuan Yang, and Licheng Jiao. Spatial–spectral-graph-regularized low-rank tensor decomposition for multispectral and hyperspectral image fusion. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(4):1030–1040, 2018.

[153] Lei Zhang, Wei Wei, Chengcheng Bai, Yifan Gao, and Yanning Zhang. Exploiting clustering manifold structure for hyperspectral imagery super-resolution. *IEEE Transactions on Image Processing*, 27(12):5969–5982, 2018. URL zhang2018exploiting.

[154] Lei Zhang, Jiangtao Nie, Wei Wei, Yong Li, and Yanning Zhang. Deep blind hyperspectral image super-resolution. *IEEE Transactions on Neural Networks and Learning Systems*, 32(6):2388–2400, 2020. URL https://github.com/JiangtaoNie/DBSR.

[155] Lei Zhang, Jiangtao Nie, Wei Wei, Yanning Zhang, Shengcai Liao, and Ling Shao. Unsupervised adaptation learning for hyperspectral imagery super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3073–3082, 2020. URL https://github.com/JiangtaoNie/UAL-CVPR2020.

[156] Tao Zhang, Ying Fu, Lizhi Wang, and Hua Huang. Hyperspectral image reconstruction using deep external and internal learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8559–8568, 2019.

[157] Xueting Zhang, Wei Huang, Qi Wang, and Xuelong Li. Ssr-net: Spatial–spectral reconstruction network for hyperspectral and multispectral image fusion. *IEEE Transactions on Geoscience and Remote Sensing*, 59(7):5953–5965, 2020. URL https://github.com/hw2hwei/SSRNET.

[158] Yifan Zhang, Steve De Backer, and Paul Scheunders. Noise-resistant wavelet-based bayesian fusion of multispectral and hyperspectral images. *IEEE Transactions on Geoscience and Remote Sensing*, 47(11):3834–3843, 2009.

[159] Ke Zheng, Lianru Gao, Wenzhi Liao, Danfeng Hong, Bing Zhang, Ximin Cui, and Jocelyn Chanussot. Coupled convolutional neural network with adaptive response function learning for unsupervised hyperspectral super resolution. *IEEE Transactions on Geoscience and Remote Sensing*, 59(3):2487–2502, 2020.

[160] Ke Zheng, Lianru Gao, Danfeng Hong, Bing Zhang, and Jocelyn Chanussot. Nonregsrnet: A nonrigid registration hyperspectral super-resolution network.

IEEE Transactions on Geoscience and Remote Sensing, 60:1–16, 2021. URL https://github.com/saber-zero/NonRegSRNet.

[161] Yanfei Zhong, Xin Hu, Chang Luo, Xinyu Wang, Ji Zhao, and Liangpei Zhang. Whu-hi: Uav-borne hyperspectral with high spatial resolution (h2) benchmark datasets and classifier for precise crop identification based on deep convolutional neural network with crf. *Remote Sensing of Environment*, 250:112012, 2020.

[162] Jie Zhou, Daniel L Civco, and JA Silander. A wavelet transform method to merge landsat tm and spot panchromatic data. *International journal of remote sensing*, 19(4):743–757, 1998.

[163] Yuan Zhou, Anand Rangarajan, and Paul D Gader. An integrated approach to registration and fusion of hyperspectral and multispectral images. *IEEE Transactions on Geoscience and Remote Sensing*, 58(5):3020–3033, 2019. URL https://github.com/zhouyuanzxcv/Hyperspectral.

[164] Zhiyu Zhu, Junhui Hou, Jie Chen, Huanqiang Zeng, and Jiantao Zhou. Hyperspectral image super-resolution via deep progressive zero-centric residual learning. *IEEE Transactions on Image Processing*, 30:1423–1438, 2020. URL https://github.com/zbzhzhy/PZRes-Net.

[165] Boris Zhukov, Dieter Oertel, Franz Lanzl, and Gotz Reinhackel. Unmixing-based multisensor multiresolution image fusion. *IEEE Transactions on Geoscience and Remote Sensing*, 37(3):1212–1226, 1999.

[166] Raul Zurita-Milla, Jan GPW Clevers, and Michael E Schaepman. Unmixing-based landsat tm and meris fr data fusion. *IEEE Geoscience and Remote Sensing Letters*, 5(3):453–457, 2008.