Analysis and comparison of superresolution methods on a multispectral environment

Rubén Llaquet Granado

Abstract– The use of multispectral data is increasingly frequent in a wide range of fields due to multispectral sensors becoming more affordable and how valuable is the information they provide. This information, however, usually needs to be combined with data coming from other sensors, which commonly causes resolution disparity. On the other hand, the recent high interest in deep learning algorithms and its great performance made it inevitable for the scientific community to explore its potential on the superresolution field. In the last years, a lot of great solutions showed impressive results when upscaling colour images. In this project, we adapted a state-of-the-art superresolution method to work with data composed of N bands and studied its performance compared to a selection of conventional methods in order to get a good solution to the aforementioned resolution disparity problem. Most of the results provided show that deep learning methods can be befittingly used to upscale N-band images, although there's still room for improvement.

Keywords– Multiband Images, Multispectral Dataset, Super-Resolution, Quality Estimation, Deep Learning, Neural Network, SRGAN, Generative Adversarial Networks.

1 Introduction

Multispectral images [1] capture data within specific wavelength ranges across the electromagnetic spectrum. While the most common example of this are RGB images, which capture the Green, Red and Blue spectral bands, the use of sensors sensible to wavelength ranges out of the visible spectrum allows the extraction of valuable information that the human eye would fail to capture with its biological receptors.

The ability to capture this non-visible information makes multispectral imaging adequate for a wide range of fields, such as precision farming, meteorological prediction, aerial traffic control, surveillance and medical imaging, where the information of some non-visible spectral bands complements the information of the visible range. Most of the currently used remote sensing technologies obtain information from non-visible ranges of the spectrum [2], and the increasing affordability of the sensors used for this task is boosting its use in a large number of commercial systems [3]. This approach to the commercial sector is increasing the demand for better data analysis algorithms that take advantage of this non-visible information.

In order to use information of a wide part of the spectra, it is commonly needed to fuse the data obtained using sensors with different technologies, which usually capture information with different spatial resolution. This resolution disparity makes it harder to directly combine the data, and it is commonly needed to apply either super-resolution techniques to lower resolution spectral bands or downscaling techniques to higher resolution spectral bands. The direct way to avoid resolution mismatch between two images is to downscale the image with higher resolution so it coincides with the one with lower resolution. However, this technique means a loss of information of the higher resolution images. In order to keep this information, it is possible to upscale the lower resolution images approximating the value of the new pixels using information of the known pixels. There are numerous super-resolution methods used to enhance the resolution of images which can be used to equalise the resolution of multiple channels obtained from sensors with different resolutions while keeping the information of the higher resolution channels. These methods include deep learning algorithms, which have proven to provide great results when applied to colour images.

In this document, we compare the performance of a selection of the most used super-resolution methods, adjusting them to an environment with images composed of N bands and studying its performance when used to upscale this kind of data.

This document is organised as follows: Sections 2 and 3 show the established goals and the methodology used during this project respectively. Section 4 explains all the methods used, and section 5 is focused towards the implementation of the methods and the comparison software. Finally, sections 6 and 7 show the obtained results and conclusion.

• Contact e-mail: rubentolva@gmail.com
• Specialisation: Computer science
• Work tutored by: Daniel Ponsa Mussarra
• School year 2017/18

2 PROJECT GOALS

The main goal of this project has been to study the different superresolution methods and its performance when used as a solution for spatial resolution disparity in a multispectral environment. Among the studied methods, we focused on the study and adaption to a multispectral environment of a deep learning solution designed for colour image quality enhancement. The main motivation has been to understand and maximise the capabilities of multispectral data analysis and to generate a system capable of carrying through this task efficiently.

In order to complete this objective and establish a better organisation that let us keep track of the project process, we defined multiple sub-objectives, which we divided into several tasks and worked on in an iterative way. The main identified objectives are:

1. Implementing some of the most used superresolution methods, adapting them to multispectral images.
2. Implementing performance metrics to measure the performance of the implemented superresolution methods.
3. Establishing a multispectral dataset.
4. Developing a pipeline to automatically evaluate the performance of the different superresolution methods based on the implemented quality metrics, adapting it to work with multispectral images.
5. Studying the adequacy of the implemented performance metrics on multispectral environments, making a quantitative and qualitative comparative.
6. Studying the performance of the implemented superresolution methods in each situation, testing them with both multispectral and colour images.

3 METHODOLOGY

For this project, we decided to use an agile software development methodology [4], which we consider appropriate due to the large quantity of review sessions associated to the deliveries and the nature of the objectives. This methodology includes the following features:

- It is based on iterations. The project is divided into small incremental sections which contribute to the evolution of the product. Each of these sections involves a planning and design phase before the implementation, and the work done is revised at the end of each iteration.
- It features an efficient communication with the TFG tutor, who agrees to make a personal commitment after each iteration.
- It focuses on quality software over comprehensive documentation. This doesn't apply to the deliverable documents, since they are part of the project goals.

The project work has been organised to follow the organisation shown in 13 in order to fit the established delivery dates. This Gantt chart representation has been made to facilitate tracking the time spent in each of the identified tasks. Some of the tasks related to the development of the pipeline and the implementation of methods have been divided into subtasks, each of them representing one iteration upon all the process of design, implementation and testing.

In order to determine which design was the most appropriate for the developed software, we took into account the following requirements, directly derived from the project goals:

- The system shall provide the user with the ability to select which preprocessing to apply depending on the input data.
- The system shall provide the user with the ability to select one or more superresolution methods to apply to the input dataset.
- The system shall provide the user with the ability to select one or more quality metrics to measure the performance of the superresolution methods applied.
- The system shall provide the user with the ability to easily integrate new superresolution or quality metric implementations.

Since those features clearly indicated the need for a modular design, we decided to design and developing the software using Object Oriented Programming. We decided to implement the software using the MATLAB program language due to the large quantity of already implemented tools it provides, and its ease of use for image processing tasks. For the implementation of all the tasks involving deep learning algorithms we decided to use Python and the TensorFlow framework because of the high availability of implemented code and the community support.

The project files are organised in a git repository using the GitLab framework, which, along with the synchronised storage and organisation, provides a great solution for version control and backup tasks. This repository will be made public once we elaborate the appropriate documentation, and a download links for the used datasets will be provided.

4 STATE OF THE ART

During the first phase of the project, we have been focused towards the research of the current solutions for image superresolution. On one hand, in order to measure and compare the performance of the implemented superresolution methods, we studied which were the most used quality measuring methods. In section 4.1, we explain the different studied methods. On the other hand we studied which are the most used superresolution methods in order to select and implement some of them, focusing mainly on the implementation and adaption to a multispectral environment of a deep learning solution. The studied methods are explained in section 4.2.

4.1 Quality metrics

To compute the performance of superresolution methods, the most common approach is to apply a downscaling process to a set of high resolution images and use said superresolution methods to upscale them again in order to match the previous resolution. Having a high resolution image I_{HR} and the reconstructed image I_{SR}, the quality measuring methods compute the performance of the superresolution methods by comparing them.

In this project, we implemented the following two conventional quality metrics:

- Mean Squared Error (MSE) [5], which simply computes the average of a set of errors, using squared errors to remove negative signs as follows:

$$MSE(I_{HR}, I_{SR}) = \frac{1}{n}\sum_{i=1}^{n}(I_{HR_i} - I_{SR})^2,$$

  where n is the number of pixels of the image. This metric is widely used in image processing tasks due to its simplicity, but it does not take into account the human perception, thus performing poorly when measuring perceptual change.

- Peak Signal-to-Noise Ratio, or PSNR [6], is defined via the MSE and computed as follows:

$$PSNR(I_{HR}, I_{SR}) = 10 \cdot \log_{10}\left(\frac{MAX_I^2}{MSE(I_{HR}, I_{SR})}\right)$$

  where MAX_I is the maximum possible pixel value of the image. It tries to resemble the human perception of reconstruction quality, although it has proven to perform poorly when compared to other perceptual quality metrics that take into account the Human Visual System.

Measuring the performance of superresolution methods in a quantitative way has proven to be a very difficult task, since most superresolution methods create reconstructions of the images with a high level of detail, which, although very different, they are visually perceived as very similar to their high quality representation. That is why the state-of-the-art methods use iterative algorithms that try to capture this perceptual similarity taking into account the Human Visual System (HVS) when comparing both images [7]. For this project, we implemented the following quality metrics that take into account the HVS:

- Multi-Scale Structural Similarity (MSSIM) [8], which, on contrast with the previously mentioned techniques, considers structural information to compute the perceived change, taking into account the information carried by the dependencies between locally near pixels.

- PSNR-HVS [9], which improves the PSNR measure by taking into account HVS properties. Many studies confirm that the HVS is more sensitive to low frequency distortions, contrast changes and noise. This metric takes this into account, and has proven to provide a higher fidelity compared to the previously mentioned methods when comparing perceptually similar images.

- PSNR-HVS-M [10], which improves the PSNR-HVS measure by being able to cope with an effect of the HVS known as visual masking. This effect makes some distortions in spatial frequencies practically unnoticeable when there exist one or more frequency components that are considerably more intensive.

4.2 Super-resolution methods

There are numerous super-resolution methods, going from classic interpolation algorithms with simple and fast implementations [11] such as bilinear or bicubic, to the use of deep learning algorithms [12] trained to perform image super-resolution.

The classic interpolation methods mentioned above compute an arithmetic approximation of the value of each pixels using the nearest known pixel values.

A more advanced approach is using dictionary-based superresolution methods, which consist on learning a dictionary of features from a set of low resolution images and their equivalent high resolution images. Using this features, these dictionary-based methods upscale new images by finding the nearest representation of each low resolution to high resolution projection, thus relying on precalculating and storing this projections to improve execution speed and performance. Examples of this are the Anchored Neighbourhood Regression method [13], and the Super Resolution Forests [14].

Currently, deep learning algorithms are being applied to superresolution tasks. these methods use datasets of low-resolution images and their high-resolution counterparts, to learn a mapping between them [15]. These algorithms have proven to work very well in terms of image quality, although the need to train them during long periods of time with large datasets of images makes them often harder to use. Examples of this are the SRResNet [16] and the SRGAN [17].

Besides, when extra information is available, such as single band representations of the same image with higher resolution, pansharpening algorithms can be used [18]. These algorithms project the information of low resolution image using the known features of the high resolution equivalent, generating a sharpening effect [19]. However, this technique acts as an extra step to improve the image quality after applying superresolution.

Even though all these approaches technically provide just an approximation of the information —which can be less precise when the difference of resolutions is considerable— they have proven to perform very efficiently, and they could improve the precision of the results of multispectral image analysis.

In this project, we have implemented and adapted to a multispectral environment three conventional methods (see figure 1) and one deep learning method:

- Nearest neighbour interpolation [20], which simply selects the value of the nearest point, and does not consider the values of neighbouring points at all. Since the information remains the same after upscaling the images, we will use this method to show a magnified version of the low resolution images.

- Bilinear interpolation [21], which sets the new pixel value based on a weighted average of the 4 pixels in the nearest 2×2 neighbourhood of the pixel in the original image.
- Bicubic interpolation [22], which performs similar to bilinear, but using a more sophisticated bicubic function considering 16 pixels in the nearest 4×4 neighbourhood of the pixel in the original image.

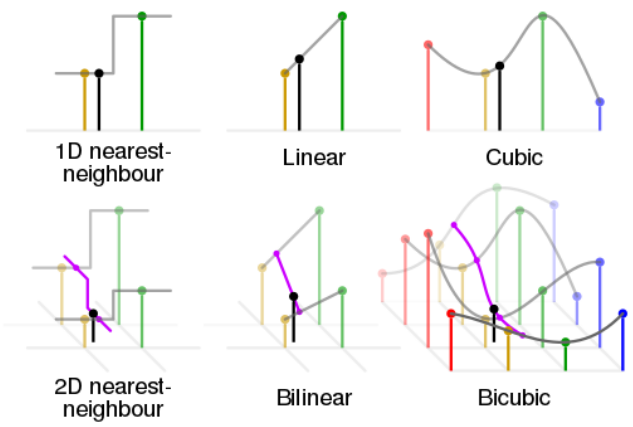


Fig. 1: Visual comparison between Near Neighbour interpolation, Bilinear interpolation and Bicubic interpolation. The black dot corresponds to the interpolated point and the coloured points correspond to the neighbouring samples.

- Super Resolution Adversarial Network (SRGAN) [17], which uses a Generative Adversarial Network (GAN) to learn the mapping between pairs of images. To learn this mapping, the network minimises a given loss function that estimates how different the generated image and the high resolution representation are. While most of the deep learning superresolution methods used MSE as loss function, the SRGAN introduces a new loss function. This loss function is calculated as a weighted sum of two loss functions: content loss and adversarial loss. The former is aimed to ensure the similarity between the pair of images and it is computed directly comparing the feature maps of both images (obtained through a pretrained VGG network [23]). The latter is computed by the Discriminator Network, a network trained to estimate the similarity between the data distribution of both images. The architecture of both the Generator and the Discriminator Network are shown in figure 2

  Since upscaling larger images can be too slow, some implementations convert the image to a YCbCr colour space and upscale only the component Y (which denotes the luminance [24] since human vision is more sensitive to the luminance differences than chromatic differences), using faster methods for upscaling the other channels. However, our implementation applies superresolution to all three RGB channels.

5 Developed Software

In order to ease the comparison between super-resolution methods, we propose an execution pipeline capable of computing the performance of a list of method implementations in an automatic way, while keeping track of the experiments and allowing certain customisation. Besides, we implemented some optional modules that allow the software to process multiband datasets when specified. Since most of the superresolution and quality measuring methods are implemented to work with images with three bands, we studied and implemented different ways to apply those methods to multispectral data.

5.1 Multispectral image processing

One easy way to apply superresolution methods to multispectral images is to upscale each band individually and fuse the N bands afterwards. However, since using the SRGAN to upscale images with a large number of bands can be very slow, we have studied and implemented alternative ways to process multispectral images faster. We first apply PCA on the image to scale and project its content according to the computed eigenvectors so that all the information is represented in the first bands. Using this representation, we implemented two possible ways to apply superresolution:

- We use the SRGAN to upscale the first three PCA bands, upscaling the other bands with simpler superresolution methods, such as Bilinear interpolation.
- We use the SRGAN to upscale only the first PCA band, upscaling the other bands with simpler superresolution methods, such as Bilinear interpolation.

After superresolution is applied, we revert the application of PCA to get the upscaled representation in the multispectral space. We decided to use all the PCA bands to convert the image to the multispectral space on both approaches, although a faster solution could be to use only the first PCA bands, assuming the consequent loss of information.

5.2 Pipeline scheme

We used MATLAB classes to create a pipeline as modular as possible in order to make it easier to add features. We defined abstract classes for both upscaling and evaluation methods, and designed the pipeline to work with these templates so the addition of new methods do not need the pipeline to be adapted.

A diagram of the work flow of the pipeline is shown in figure 3. The main modules of the pipeline are the following:

- Multispectral preprocessing module. This module is composed by two independent submodules that are only applied when explicitly said, and it is designed to preprocess the input data when it is composed of more than three bands.
  - Channel correlation module. Since some sensors capture the different bands with a small variation on perspective, it is needed to make a channel correlation in order to improve the visibility of the qualitative results.
  - PCA module. As mentioned before, we use PCA to project the multispectral data so that most of the information is represented on the first bands.

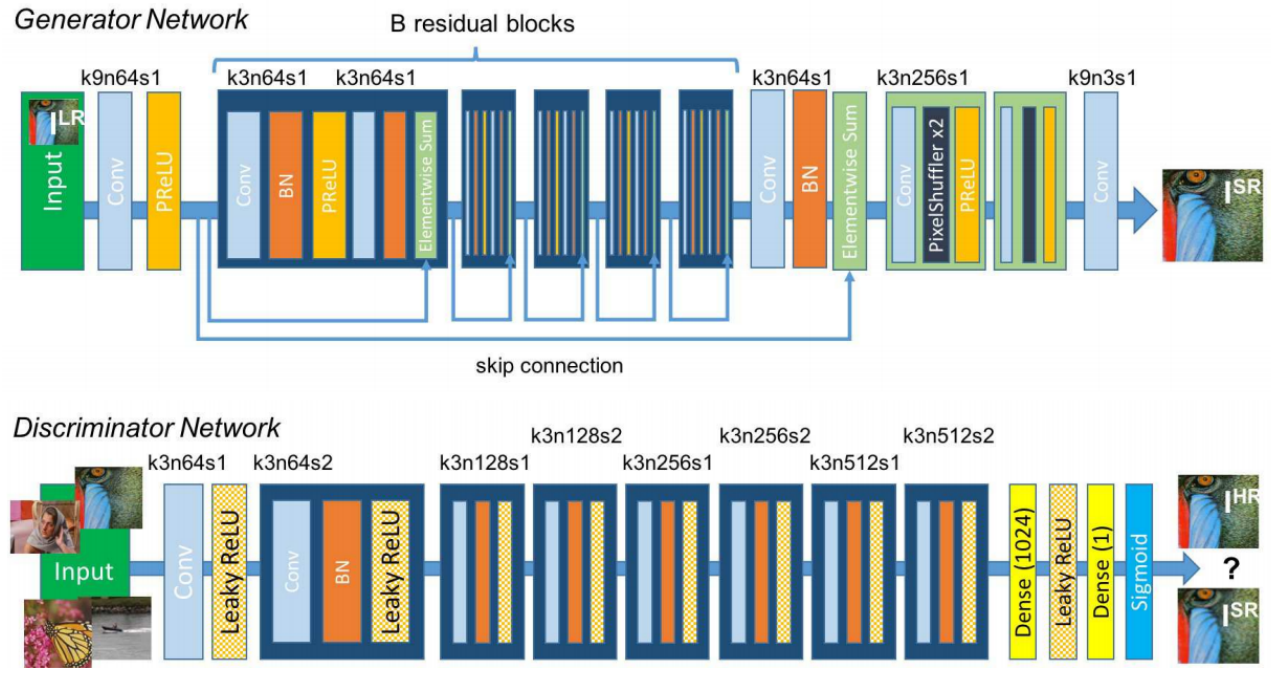


Fig. 2: Architecture of the Generator and Discriminator Networks of the SRGAN

- Downscaling module. This module computes the low resolution images I_{LR} downscaling the high resolution images I_{HR} to a specified factor.

$$I_{LR} = f_{downscale}(I_{HR}, factor)$$

- Upscaling module. This module uses the selected upscaling methods to generate high resolution reconstructions I_{SR} from the set of low resolution images I_{LR}.

$$I_{SR} = f_{upscale}(I_{LR}, factor, method)$$

- Multispectral postprocessing module. This module is only applied when PCA has been performed, and applies the inverse operation to the high resolution reconstructions to convert them to its original basis.

- Evaluation module. This module uses the selected evaluation methods to compare the high resolution reconstructions I_{SR} generated by the upscaling module and the high resolution image I_{HR}, and stores it in a history file.

$$Result = f_{measure}(I_{SR}, I_{HR})$$

6 Results

6.1 Dataset selection

For this project, we have researched which are the most used datasets on super-resolution tasks. The following datasets have been used during the implementation, training and testing of the developed software:

- To study the performance of the implemented super-resolution methods on colour images we used the DIV2K dataset [25], which is a 2K resolution image dataset used for image restoration tasks. it consists of 800 training images and 100 validation images, which we used during the training process of the SRGAN, and 100 test images, which we used to study the performance of all the implemented methods.

- During the research process, we identified numerous multispectral datasets used for image processing tasks, but they presented two features that made them inappropriate for our purpose:

  - They were usually composed of a large quantity of bands, making it hard to make a qualitative comparison between the studied methods.
  - The images of a same dataset were too similar, since they were usually used on the research of a specific problem.

  Thus, in order to establish a dataset that did not present those features, we decided to create a custom dataset composed of 15 images captured with a Parrot SEQUOIA+ sensor, which captures up to seven spectral bands: four discrete spectral bands (Green, NIR, Red and RedEdge) and 3 non-discrete spectral bands (as part of the RGB image). Figure 10 shows an example of these bands.

6.2 Results on an RGB environment

In a first experiment, we used the 100 test images from the DIV2K dataset to analyse the performance of the implemented methods, along with the reliability of the quality measures computed. From this dataset, we isolated a set of four images that concisely represent most of the encountered results. Figure 11 shows a comparison between the selected colour images when upscaled using bicubic interpolation and the SRGAN. The original high resolution image

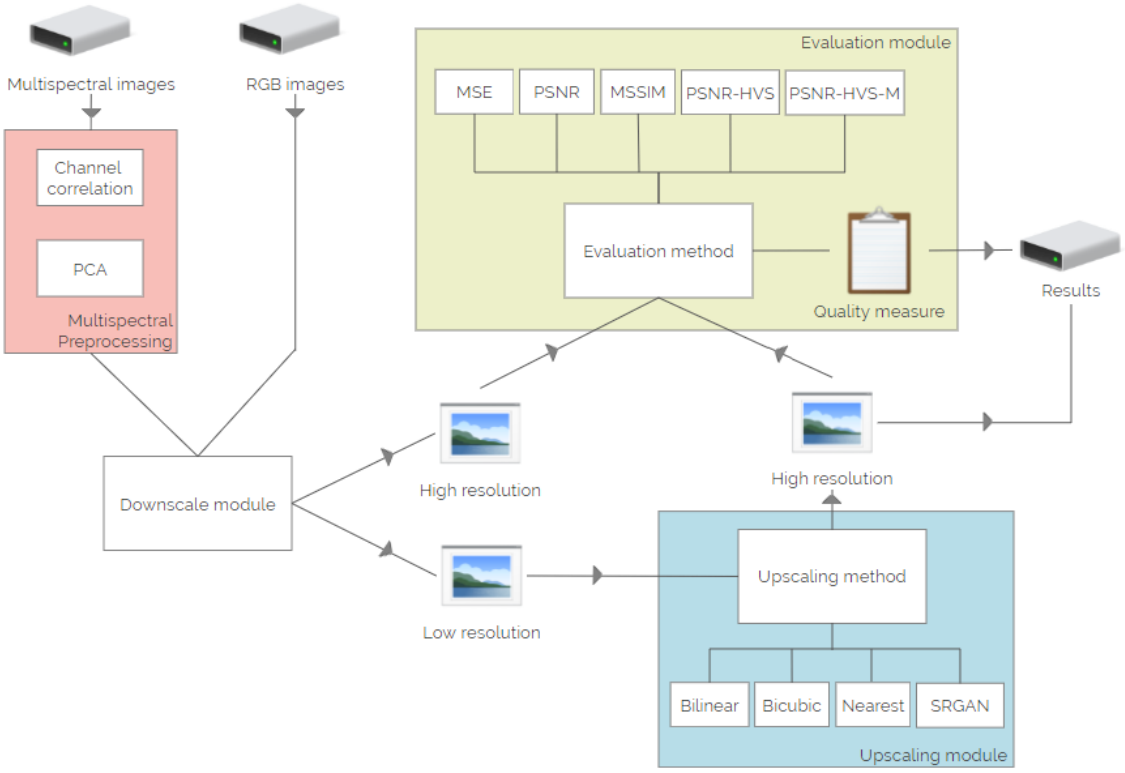


Fig. 3: Diagram of the work flow of the execution pipeline implemented

and a magnified version of the low resolution (upscaled using Nearest neighbour interpolation) are also shown. Table 1 shows the quantitative metrics for each image, obtained with each one of the implemented evaluation methods.

The images obtained with the SRGAN can be perceived in most of the experiments as better quality images in comparison with the ones obtained through a bicubic interpolation, although the SRGAN fails to generate a credible output in some spots. Figure 4 shows a comparison between our SRGAN model and the results obtained with another training of the same network architecture, where we can observe those effects with more detail.

(a) (b)

Fig. 4: Visual comparison between (a) the results of our SRGAN model and (b) results obtained with the same network architecture [26].

Specifically, our model generates a black spot on the hair of the person on the painting and wrong textures on the top left of the image. A more exhaustive fine-tuning and training of the network would help solve these problems and improve its performance.

It is also noticeable how the image obtained through bicubic interpolation shows a blur effect, while the image upscaled using the SRGAN looks much more sharp. This is due to the nature of both algorithms. On one hand, the bicubic interpolation discards high frequency information, resulting in smooth textures that often imply poor perceptual quality. On the other side, the SRGAN generates a reconstruction with much more high frequency content and even adds noise, trying to simulate high quality textures. Figures 6 and 5 show two magnified regions of the first image where this effect is more visible. The textures on the wing and the leaf, while being different than the ones in the original image, have a better perceptual quality in comparison with the images obtained through bicubic interpolation

(a) (b) (c)

Fig. 5: Magnified detail of a a leaf for the images generated with a (a) bicubic interpolation and the (b) SRGAN and the (c) original image

This predictive generation of the high frequency infor-

(a) (b) (c)

Fig. 6: Magnified detail of a wing of the butterfly for the images generated with a (a) bicubic interpolation and the (b) SRGAN and the (c) original image

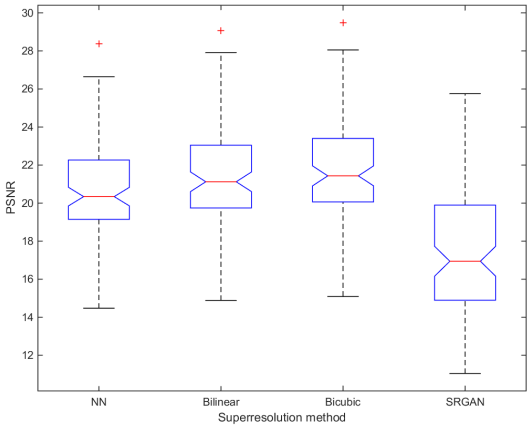


Fig. 7: Box plot representation of the performance of the implemented superresolution methods on the DIV2K dataset according to the PSNR metric.

mation leads to a problem when comparing the quality of the reconstructed and the original images in a quantitative way. This textures often have a better perceptual quality, however, the generated image details notably differ from the ones in the original image, making it hard to measure the difference statistically. Because of this, and as we can see in table 1, the MSE score classifies the images generated with the SRGAN as the more distant to the original with much higher scores. The perceptual quality measures, such as MSSIM, PSNR-HVS and PSNR-HVS-M, show a more suitable score according to the perceptual quality, although they still fail to capture the perceptual similarity in most of the cases. Figures 7 and 8 show the behaviour of the studied methods in accordance to a conventional quality metric (PSNR) and a perceptual metric (PSNR-HVS-M). It is noticeable how the PSNR-HVS-M metric captures much better the perceptual similarity of the images generated with the SRGAN in comparison with PSNR.

6.3 Results on a multispectral environment

For this experiment, we used the custom SEQUOIA+ multispectral dataset, isolating two images to illustrate the obtained results.

In the case of the conventional quality metrics we computed the metric to each one of the bands individually, computing the mean of the result of each channel afterwards. However, the metrics that take into account the Human Visual System can only be applied to colour images. In order

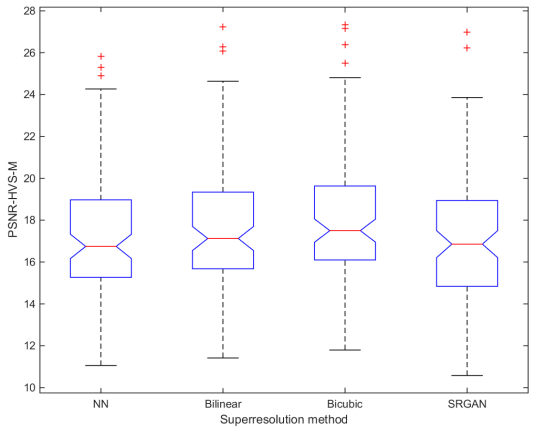


Fig. 8: Box plot representation of the performance of the implemented superresolution methods on the DIV2K dataset according to the PSNR-HVS-M metric.

to test if this kind of metrics are appropriate for measuring the performance of superresolution methods on multiband images, we selected the three multispectral bands whose wavelengths are close to those of the RGB representation. In our case, we used the Green, Near Infrared and Red bands. We also used these three bands to study the results in a qualitative way. This results are shown in the figure 12, and the table 2 gathers the obtained quantitative measures.

In this case, we review the results of applying the SRGAN in the three aforementioned ways:

- Upscaling only the first PCA band using the SRGAN and upscaling the rest using Bicubic interpolation (labelled as SRGAN V1 on table 2).
- Upscaling the three first PCA bands using the SRGAN and upscaling the rest using Bicubic interpolation (labelled as SRGAN V2 on table 2).
- Upscaling each of the bands that compose the image (labelled as SRGAN V3 on table 2).

Figure 9 shows a comparison between these three approaches. We can observe how the images obtained after upscaling the first three bands of the PCA representation present much more noise and multiple black spots in comparison with the rest of the images. One possible explanation for this is that PCA gathers all the important information in the first channel, leaving in the other two the details. Since the RSGAN has been trained with images RGB, it is not adapted to images with so different structural features. This issue can be solved training the SRGAN with a dataset of this sort of images, so its weights adapt to all the important information being in one channel. Since the information gathered on the first channel presents similar structural features in comparison with one band of a colour image, the result of upscaling only the first band of the PCA representation does not present this kind of problems. Besides, it performs very similar in comparison with the results of the image obtained upscaling each one of the bands, making it a good solution for environments with time restrictions.

We can also see how both methods still fail to generate a credible output in some spots, creating the same effect as on

an RGB environment, which also makes it harder for all the quality metrics to perceive its similarity. A more exhaustive fine-tuning and training of the network would help solve these problems and improve its performance.

7 Conclusions

In this project, we have implemented a MATLAB framework to efficiently compare superresolution methods, along with an interface that lets us use methods implemented in Python directly from it. We designed and developed this framework to be compatible with multispectral images and proposed a way to adapt multispectral images to upscale them using superresolution methods designed for colour images. We implemented a series of commonly used superresolution methods and quality metrics, including both conventional and state-of-the-art approaches. Finally, we studied the reliability of the quality metrics and the performance of the implemented superresolution methods in a qualitative and quantitative way, aiming to determine a good solution for superresolution of multispectral images.

The current results show the great potential of the SRGAN when upscaling RGB images, although there is a lack of consistency on our implementation due to the apparition of black spots and not suitable textures in some of the images. This potential can be befittingly extended to its application on multispectral images. Although it has proven to perform poorly on images when their structural features differ notably from the ones of the RGB images, other approaches let us beneficially take advantage of its potential. We have also seen how PCA can be used to noticeably reduce the temporal cost of upscaling multispectral images with the SRGAN without a big impact on its performance.

Regarding the quality measures studied, we observed that metrics that do not take into account human perception fail to capture the perceptual quality of the images obtained with the SRGAN. This perceptual similarity between the image generated with the SRGAN and the original is reflected in perceptual metrics, such as PSNR-HVS or PSNR-HVS-M, although, due to the low reliability of our SRGAN implementation in some spots, they still occasionally show a worse score in comparison with the other methods.

A future line of this project could be focused towards the improvement of the results of the SRGAN when applying superresolution on multispectral images, which can be done by training it with multispectral data and fine-tuning the network parameters. Other future lines include the addition and study of more superresolution methods and quality metric. Finally, we could improve the readability, documentation and robustness of the code, making it easier to use for new users.

Aknowledgement

We would like to express our great appreciation to Daniel Ponsa, who worked actively to provide a constant feedback and helped to find a good solution for the problems encountered. We would like to thank Felipe Lumbreras, who provided insight and expertise that greatly assisted us through the research process, and Daniel Azemar, who helped us to carry out some specific tasks. We would also like to show our gratitude to Alex Llaquet, who kindly lent us the needed gear to construct the multispectral dataset.

References

[1] Wikipedia, "Multispectral image — Wikipedia, the free encyclopedia." http://en.wikipedia.org/w/index.php?title=Multispectral%20image&oldid=822187823, 2018. [Online; accessed 02-March-2018].

[2] Wikipedia, "Remote sensing — Wikipedia, the free encyclopedia." http://en.wikipedia.org/w/index.php?title=Remote%20sensing&oldid=828851083, 2018. [Online; accessed 03-March-2018].

[3] V. C. Coffey, "Multispectral imaging moves into the mainstream." https://www.osa-opn.org/home/articles/volume_23/issue_4/features/multispectral_imaging_moves_into_the_mainstream/, 2012. [Online; accessed on 02-March-2018].

[4] Wikipedia, "Agile software development — Wikipedia, the free encyclopedia." http://en.wikipedia.org/w/index.php?title=Agile%20software%20development&oldid=829719821, 2018. [Online; accessed 10-March-2018].

[5] Wikipedia, "Mean squared error — Wikipedia, the free encyclopedia." http://en.wikipedia.org/w/index.php?title=Mean%20squared%20error&oldid=841359709, 2018. [Online; accessed 26-May-2018].

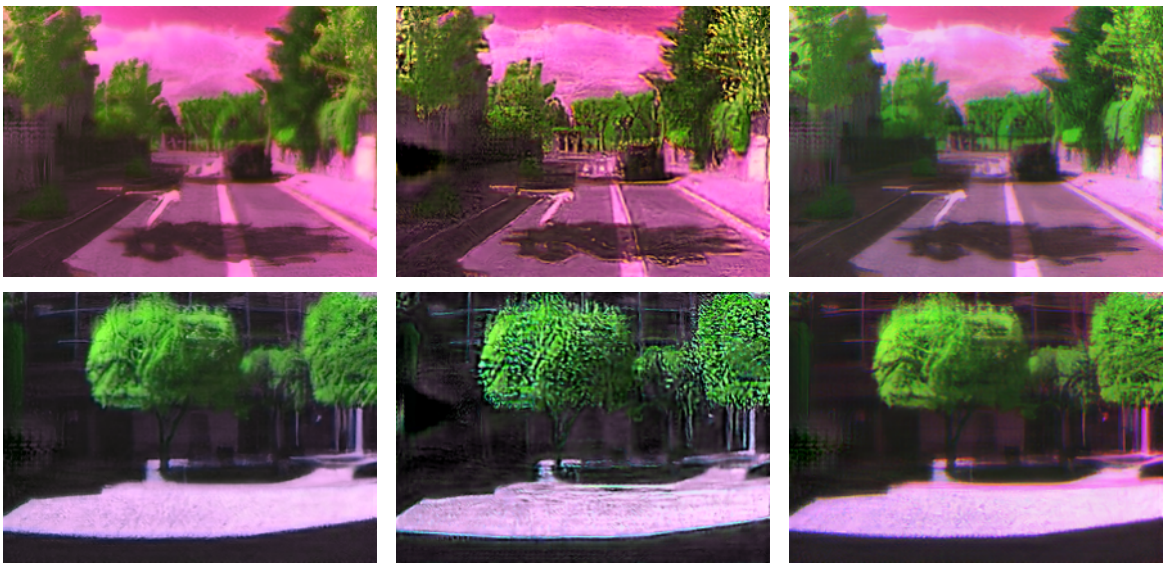
[6] Wikipedia, "Peak signal-to-noise ratio — Wikipedia, the free encyclopedia." http://en.wikipedia.org/w/index.php?title=Peak%20signal-to-noise%20ratio&oldid=841928208, 2018. [Online; accessed 26-May-2018].

[7] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual Losses for Real-Time Style Transfer and Super-Resolution," *ArXiv e-prints*, Mar. 2016.

[8] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, vol. 2, pp. 1398–1402 Vol.2, Nov 2003.

[9] K. Egiazarian, J. Astola, N. Ponomarenko, V. Lukin, F. Battisti, and M. Carli, "New full-reference quality metrics based on hvs," in *Proceedings of the Second International Workshop on Video Processing and Quality Metrics*, vol. 4, 2006.

[10] N. Ponomarenko, F. Silvestri, K. Egiazarian, M. Carli, J. Astola, and V. Lukin, "On between-coefficient contrast masking of dct basis functions," in *Proceedings of the third international workshop on video processing and quality metrics*, vol. 4, 2007.



(a) (b) (c)

Fig. 9: Qualitative comparison between (a) the image generated upscaling only the first band of the PCA representation using the SRGAN, (b) the image generated upscaling the first three PCA bands using the SRGAN and (c) the image generated upscaling each of the bands independently.

[11] H. Prashanth, H. Shashidhara, and B. M. KN, "Image scaling comparison using universal image quality index," in *Advances in Computing, Control, & Telecommunication Technologies, 2009. ACT'09. International Conference on*, pp. 859–863, IEEE, 2009.

[12] K. Kańska and P. Goliński, "Using deep learning for single image super resolution," 2017.

[13] R. Timofte, V. De, and L. Van Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, pp. 1920–1927, IEEE, 2013.

[14] S. Schulter, C. Leistner, and H. Bischof, "Fast and accurate image upscaling with super-resolution forests," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3791–3799, 2015.

[15] K. Hayat, "Super-Resolution via Deep Learning," *ArXiv e-prints*, June 2017.

[16] C. Dong, C. Change Loy, K. He, and X. Tang, "Image Super-Resolution Using Deep Convolutional Networks," *ArXiv e-prints*, Dec. 2015.

[17] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," *ArXiv e-prints*, Sept. 2016.

[18] Wikipedia, "Pansharpened image — Wikipedia, the free encyclopedia." http://en.wikipedia.org/w/index.php?title=Pansharpened%20image&oldid=771389354, 2018. [Online; accessed 22-April-2018].

[19] C. Padwick, M. Deskevich, F. Pacifici, and S. Smallwood, "Worldview-2 pan-sharpening," in *Proceedings of the ASPRS 2010 Annual Conference, San Diego, CA, USA*, vol. 2630, 2010.

[20] Wikipedia, "Nearest-neighbor interpolation — Wikipedia, the free encyclopedia." http://en.wikipedia.org/w/index.php?title=Nearest-neighbor%20interpolation&oldid=763429489, 2018. [Online; accessed 26-May-2018].

[21] Wikipedia, "Bilinear interpolation — Wikipedia, the free encyclopedia." http://en.wikipedia.org/w/index.php?title=Bilinear%20interpolation&oldid=809623103, 2018. [Online; accessed 26-May-2018].

[22] Wikipedia, "Bicubic interpolation — Wikipedia, the free encyclopedia." http://en.wikipedia.org/w/index.php?title=Bicubic%20interpolation&oldid=804689773, 2018. [Online; accessed 26-May-2018].

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[24] Wikipedia, "Luma (video) — Wikipedia, the free encyclopedia." http://en.wikipedia.org/w/index.php?title=Luma%20(video)&oldid=815152575, 2018. [Online; accessed 10-March-2018].

[25] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[26] brade31919, "Srgan-tensorflow." https://github.com/brade31919/SRGAN-tensorflow, 2017.

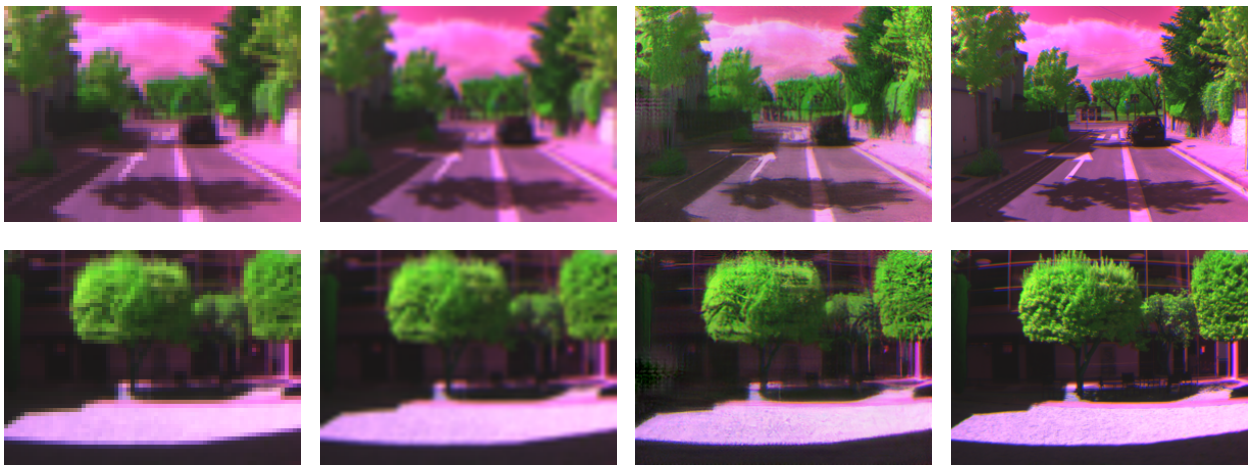
APPENDIX

(a) (b) (c) (d) (e)

Fig. 10: (a) Green, (b) Near Infra Red, (c) Red and (d) Red Edge bands from an image captured with a Parrot SEQUOIA+. The RGB image (e) is also shown.

(a) (b) (c) (d)

Fig. 11: Qualitative comparison between (a) the downscaled image, magnified using nearest neighbor, (b) an upscaled version using bicubic interpolation, (c) an upscaled version using the SRGAN and (d) the original color image.



(a) (b) (c) (d)

Fig. 12: Qualitative comparison between (a) the downscaled image, magnified using nearest neighbor, (b) a version upscaled using bicubic interpolation, (c) a version upscaled using the SRGAN (V3) and (d) the original multispectral image.

TABLE 1: QUANTITATIVE COMPARISON BETWEEN THE DIFFERENT UPSCALING METHODS APPLIED ON RGB IMAGES, ACCORDING TO THE STUDIED QUALITY METRICS.

Image	Upscaling Method	MSE	PSNR	MSSIM	PSNR-HVS	PSNR-HVS-M
Butterfly	Nearest	506.54	21.085	0.8804	17.579	16.426
	Bilinear	408.52	22.019	0.87983	18.159	17.079
	Bicubic	**378.79**	**22.347**	**0.89695**	18.692	17.534
	SRGAN	493.43	21.199	0.87377	**18.913**	**17.688**
Lion	Nearest	367.8	22.475	0.8586	21.806	20.366
	Bilinear	305.09	23.287	0.85783	22.413	21.019
	Bicubic	**288.46**	**23.53**	**0.87458**	**22.864**	**21.379**
	SRGAN	582.42	20.478	0.83899	17.47	16.446
Mushrooms	Nearest	531.29	20.878	0.84567	18.145	16.529
	Bilinear	474.63	21.367	0.82578	18.429	16.856
	Bicubic	**442.65**	**21.67**	0.85412	**19.082**	**17.378**
	SRGAN	617.35	20.226	**0.85623**	17.989	16.419
Landscape	Nearest	724.93	19.528	0.85024	15.703	14.615
	Bilinear	622.26	20.191	0.84678	16.445	15.295
	Bicubic	**619.22**	**20.212**	**0.85787**	**16.493**	**15.332**
	SRGAN	2139.9	14.827	0.71036	12.609	11.441

TABLE 2: QUANTITATIVE COMPARISON BETWEEN THE DIFFERENT UPSCALING METHODS APPLIED ON MSI IMAGES ACCORDING TO THE STUDIED QUALITY METRICS.

Image	Upscaling Method	MSE	PSNR	MSSIM	PSNR-HVS	PSNR-HVS-M
Road	Nearest	335.362	22.876	0.92105	18.088	16.599
	Bilinear	295.865	23.420	0.91675	18.032	16.644
	Bicubic	**266.916**	**23.867**	0.93167	18.815	17.281
	SRGAN (V1)	378.156	22.354	**0.93446**	17.811	16.458
	SRGAN (V2)	712.696	19.601	0.84132	13.923	12.801
	SRGAN (V3)	397.735	22.134	0.93182	**19.138**	**17.563**
Yard	Nearest	389.205	22.229	0.91751	16.263	15.500
	Bilinear	319.750	23.083	0.91152	16.763	16.013
	Bicubic	**290.110**	**23.505**	**0.92547**	**17.136**	**16.359**
	SRGAN (V2)	401.132	22.098	0.91948	15.983	15.346
	SRGAN (V2)	997.411	18.107	0.76975	13.934	13.332
	SRGAN (V3)	392.067	23.329	0.92336	16.614	15.869

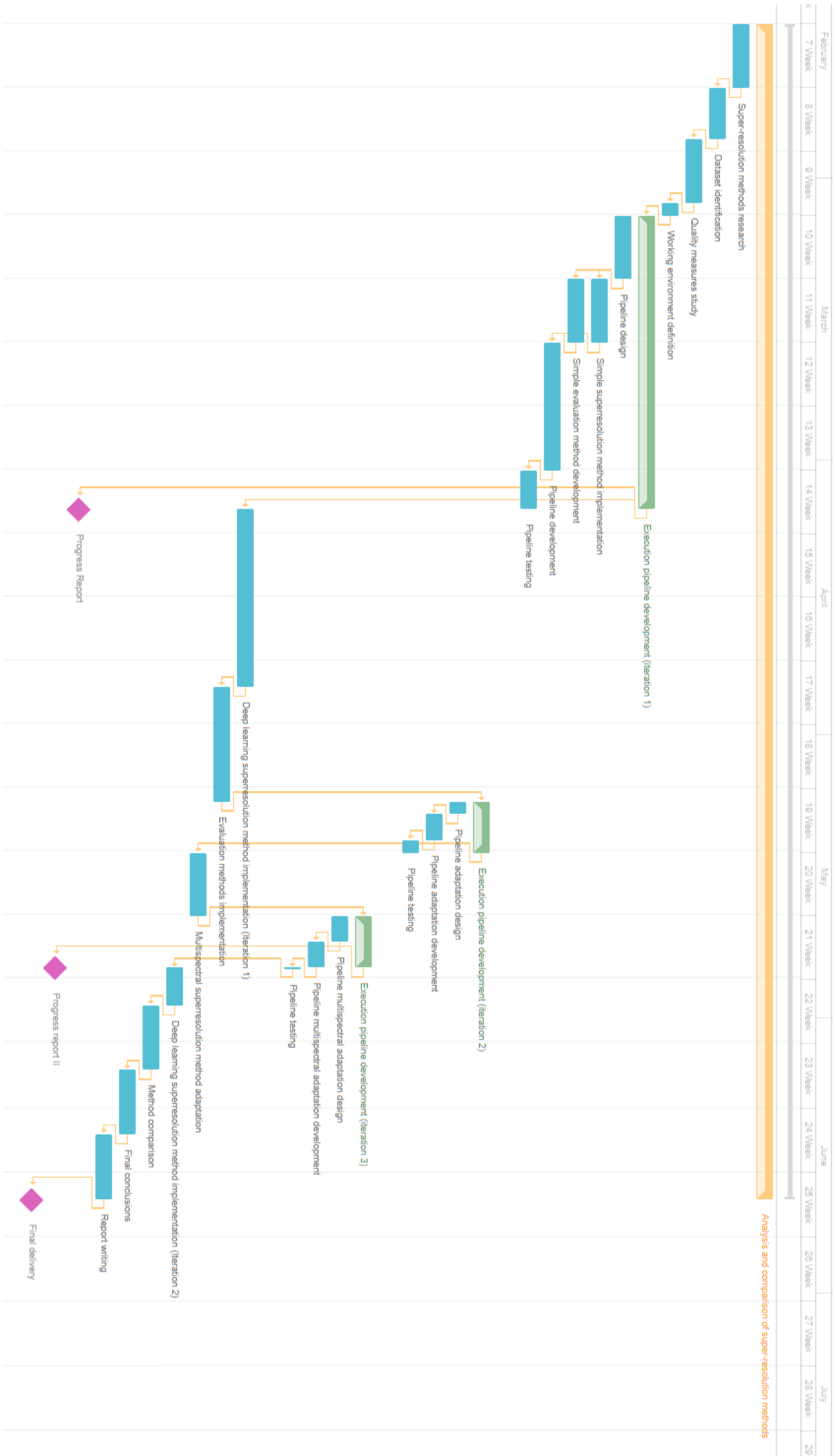


Fig. 13: Gantt chart showing the organisation of the identified tasks during the span of the project.