De Montfort University
Leicester, England, United Kingdom
Faculty of Computing, Engineering and Media

Single-Image Super-Resolution: Towards the Enhancement of Sentinel-2 Satellite Imagery

A thesis submitted in partial fulfilment of the requirements for the degree of Master of Science

by

Adam Leonard Hubble

Supervisor

Dr. Lipika Deka

September, 2021

Acknowledgements

For the supervisory figures of the work researched, developed, and delivered, a particular appreciation is awarded to Dr. Karl Hemsi, as the visionary of the work presented, as well as Dr. Charis Lanaras, whose work is adapted and more so extended to investigate the proposal put forth. For both individual's intervention, I am truly grateful. Moreover, at the forefront of supervision, I would like to announce my appreciation for both Dr. Lipika Deka and Prof. David Elizondo, given their persistent support with my queries, and interest shown for my wellbeing and academic development throughout.

Complimenting the efforts of both Dr. Lipika Deka and Prof. David Elizondo at De Montfort University, Leicester, I would also like to extend my gratitude to all other responsible University officials, especially within the Computing, Engineering and Media (CEM) faculty, for the availability of online resources that have resultingly enabled me to achieve what has been possible from a virtual-based learning experience.

Given the prolongment of what has been the COVID-19 pandemic, it goes without saying that distant company has been crucial to maintaining a degree of sanity, happiness, and a sense of hope; with this said, I would like to specially devote my thanks to my partner Shirley Lagaran Gocotano, who has offered me companionship, a sense of comfort, and a near future prospect that awaits my excitement. Furthermore, without the visiting period and regular interactions with family and friends, I feel that my physical loneliness would overcome the happiness I have been able to accumulate in these harsh times; to those responsible, thank you.

Devoted to my fellow colleagues and companions, namely Jack Wilson Moorin, Alfred John Horton, and Łukasz Pszonak, I am grateful to you each for what has been a memorable and life-changing experience, as a university student. Without your contributions to my understanding and perseverance towards my learning etiquette, I would not have become who I have evolved to be today. For this, I thank you.

Adam Leonard Hubble

Abstract

The image super-resolution (SR) problem, particularly single image super-resolution (SISR), is an infamously troublesome and ill-posed challenge that has gained a long-standing and increasing research attention for decades, in the computer vision community. Fundamentally, SISR aims to reconstruct a high-resolution (HR) image from a single low-resolution (LR) image. With the emergence of deep learning, convolutional neural network (CNN) based SR methods have been capable of inheriting the powerful capacity of deep learning, hence the initiation of deep convolutional neural networks (DCNN) and have achieved significant performance improvements over their shallower predecessors. Where the progress of SR so far, has been mainly driven by the supervised or example-based learning of LR-HR images pairs. Alongside the efforts put forth for architectural enhancements, data augmentation (DA) techniques have also been recently introduced for cost-effectively improving upon the predictive capabilities of example-based learning models. However, the implications are yet to be realised for the implementation of DA techniques in deep learning (DL) model applications, particularly within a model's predictive phase. Such that there has only been studies led into simple geometric manipulations with traditional SR models and shallow learning-based models, as of now.

Parallel to the advancements contributed to the SR community, with the proliferation of aviation technology development, an increasing number of satellites in operation, and the demand for very high-resolution (VHR) images, CNNs have attracted an increasing interest in many remote sensing (RS) applications. One of such applications is this paper's focus, that is the spatial enhancement of low-resolution satellite imagery, for advancing the surveillance and classification capability of land cover and land occupancy developments, climate change, emergency management, and crisis prevention, to name a few. Where amongst the RS community, it is mutually agreed that one of the most significant research areas in RS currently, is to develop methods for super-resolving the lower-resolution spectral bands processed by the respective satellites, to having the highest spatial resolution possible.

Given this relation, this paper explores the application of DA techniques, namely geometric self-ensemble, extended with a randomised state-of-the-art translation scheme, in the predictive phase of a deep learning SR model, to inexpensively enhance the spatial resolution of LR multispectral images. For the purposes of this investigation, a state-of-the-art deep learning model is adapted for super-resolving the low-resolution (20m Ground Sampling Distance - GSD) spectral band images of the Sentinel-2 satellite mission, to support more detailed and accurate information extraction.

Keywords: super-resolution; ill-posed challenge; computer vision; reconstruct; deep learning; deep convolutional neural network; example-based learning; data augmentation; predictive phase; geometric manipulations; remote sensing; spatial enhancement; satellite imagery; surveillance; classification; super-resolving; spatial resolution; geometric self-ensemble; translation; multispectral images; Sentinel-2

Table of Contents

List of Tables

List of Figures

Chapter 1 | Introduction

Satellite RS renditions have established vast “applications in Earth surface observations” [1][1]. Such that global monitoring is nowadays addressed by a large and increasing number of satellites [14], to uphold the demand and importance of the “societal applications” [15] aforementioned. However, Earth observation missions are conventionally known to operate at “medium to low resolution ranges” [16], to satisfy both a larger satellite swath (see *Figure 1*) and shorter temporal revisit period of the same observation site. Furthermore, with the “recent launch” [17] of multispectral instruments (MSI’s), many “widely used satellite imagers” [18] can acquire images with “multiple spectral bands with different spatial resolutions” (see *Figure 2*) [19][2]. This presents other motives for recording observational data at varying spatial resolutions, including: “storage and transmission bandwidth restrictions, improved signal-to-noise ratio (SNR) in some bands through larger pixels, and bands designed for specific purposes that do not require high spatial resolution” [18]. Still, “it is often desired to have all bands available at the highest spatial resolution”, so as to support more “detailed and accurate information extraction”, for both observation and classification studies alike.

[1] Relevant to agricultural monitoring [2], environmental conservation [3], geophysical variable estimation [4], land-use and coverage [5], urban planning [6], climate change [7], risk management [4, 8], cartography [9], biodiversity [10], geology [11], hydrology [12], and oceanography [13].
[2] Typically differing by scale factors of two to six.

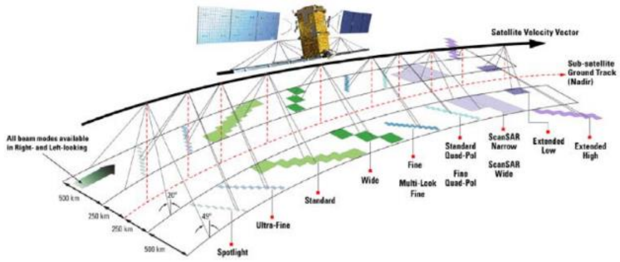


Figure 1: Visualisation of some observation geometries, nominal swath widths, and land coverage modes of the RADARSAT-2 satellite [20].

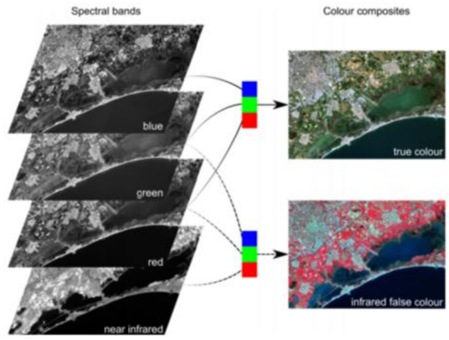


Figure 2: Visualisation of the construction of true and infra-red colour composite images, from the respective spectral band raster dataset [21].

Parallel to the development of aviation technologies and expanding industrial pressures [22], RS (see *Figure 4*) has evolved to being an "increasingly popular field in the modern society". Such that one of the "most important" [19] research areas in RS currently, is to acquire high-quality renditions from "sensors mounted on satellites" [22] (see *Figure 3*), through the advancement of super-resolution techniques [19]. This invites study efforts to develop methods for "super-resolving the lower-resolution bands", in the enablement of all image bands sharing the "highest spatial resolution" possible[3]. However, with the "continuous updating of optical instruments" [23], the spatial resolution employed by satellite images is "constantly improving", but the "imaging chips and optical components become prohibitively expensive" [24, 25] for captivating very-high spatial resolution (VHSR) imagery. Which can render a satellite "impractical when large areas have to be covered or if multi-temporal analysis have to be undertaken" [17]. Nevertheless, when also factoring the limitations of "sensor accuracy, satellite orbital altitudes, and space-ground communication bandwidth" [22], to name a few, "many" satellites are not technologically capable of fulfilling the "fast-growing" spatial resolution demands of "new generation" societal applications, for both scientific and industrial interests. Hence why it has become of "paramount importance" [1] to develop "novel post-correction methods" [22], for enhancing the spatial resolution of native satellite observations.

[3] As higher spatially resolved images "represent more detailed information of the Earth's surface" [22].

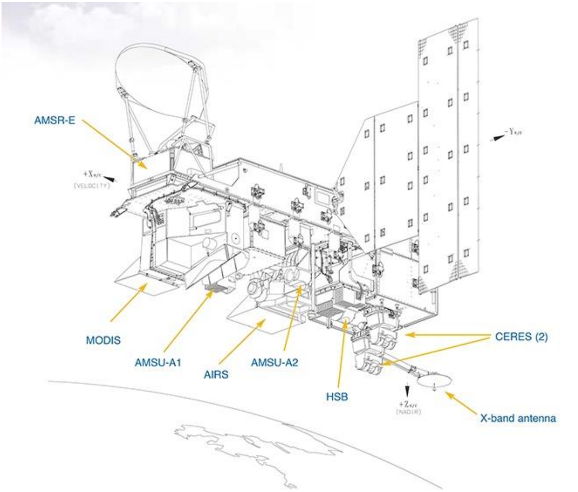


Figure 3: Visualisation of six state-of-the-art imaging instruments used in a near-polar low-Earth orbit, employed by the Aqua (EOS PM) satellite [26].

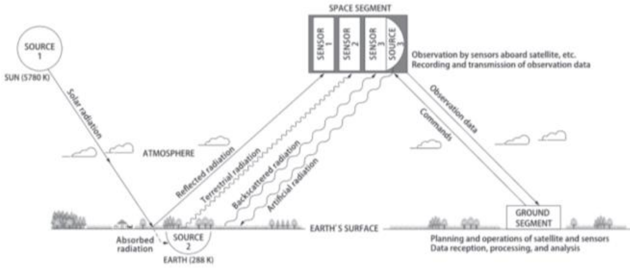


Figure 4: Visualisation of a satellite remote sensing system with five components: sources of radiation (the Sun (Source 1), the Earth (Source 2), and an artificial radiation source (Source 3)); sensor interaction with the atmosphere and the Earth's surface. Demonstrating the exchange of instruction and observation data between the space (sensors) and ground segments [27].

Aligned with the requirements for post-correction methods, several super-resolution (SR) methods have been introduced in RS [19] and have "attracted much attention" [22], with regards to the enhancement of "low-quality low-resolution RS images". Where SR technologies have provided a "promising computational imaging approach to generate high-resolution images via an existing low-resolution image or image sequences" [28], for a multitude of applications[4]. Given such, SR can be characterised as the process of "deriving images of higher resolution by applying an algorithm to a low-resolution image" [32], or sequence of low-resolution images, respectively. This presents two problems, namely single-image SR (SISR) and multi-frame image SR (MISR) [33]. For the purposes of this study and although it is a "challenging setting" [34], SISR (see *Figure 5*) is focused upon, providing its recent developments [35] that has enabled it to emerge "as a possible way" [36] to increase the spatial resolution of low-resolution satellite imagery; without requiring "additional information from other sources" [37][5]. From a practical perspective, SISR caters for an "unlimited amount of LR input images" [28], given the ability to synthetically augment data [18], which also provides a matter of convenience for efforts put forth to investigate this study at vaster measures. Achieving greater spatial resolutions of satellite imagery, allows for a "finer" [36] posterior analysis [34] of Earth's observation missions, through acquiring "greater details" and an increasing amount of knowledge regarding the "true conditions of the Earth"[6].

[4] Including medical diagnostic imaging [29], radar imaging systems [30], and satellite sensor imaging [31], to name a few.
[5] SISR methods depend upon the spatial features of an original, high-resolution variant (learning-based SR) of a given image, to "increase its resolution".
[6] Both presently and historically.

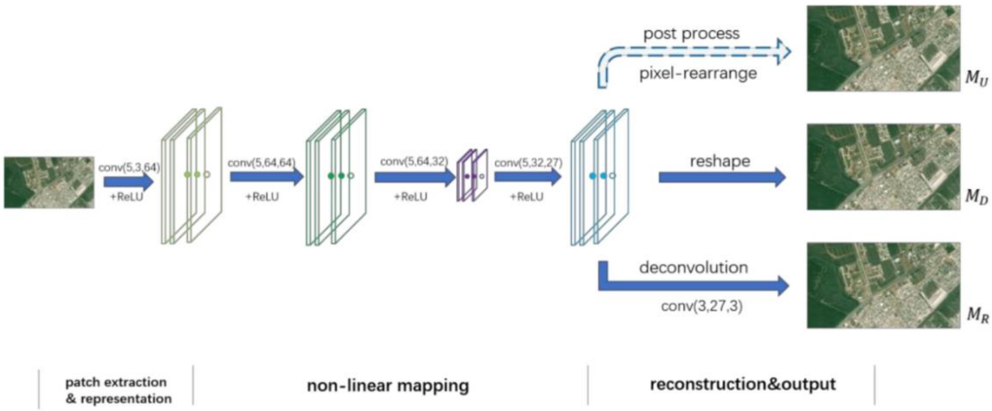


Figure 5: Visualisation of a convolutional neural network (CNN) framework, demonstrating end-to-end up-sampling for single-image super-resolution (SISR) [38].

Currently, deep learning-based methods have become "predominant in every image processing and computer vision task" [40, 34, 39], due to their performance offerings [41]. And in recent years, deep neural networks (DNNs), particularly convolutional neural networks (CNNs), have been investigated and discovered to be "very effective" [19] for combating SR problem areas; one of such areas is SISR, where CNNs have "excelled" [34] and "demonstrated superior performance" [28] within. Conventionally, CNNs are most widely regarded for their contributions to image classification [42], semantic segmentation [43], and facial recognition [44] tasks [36, 41]. However, beyond the scope of "standard" computer vision problems, DCNNs are actively being purposed for RS applications [45, 46, 47] also, due to their "effectiveness and appeal" [48] in a rising number of works[7]. This capability is mostly fulfilled by the ability of DNNs to learn "very complex non-linear relationships" [49], that when amalgamated with CNN architectures, can utilise the "high-order features of images" to construct HR renditions of LR counterparts, and ultimately "improve the performance of SR". Hence why learning-based methods have been used in image SR (ISR), for "the last decade" [50].

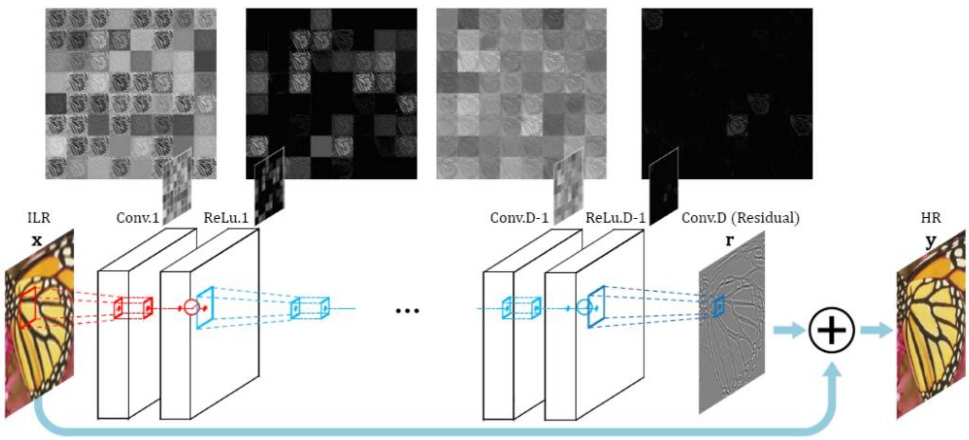


Figure 6: Visualisation of the Very Deep Super-Resolution (VDSR) network architecture [51].

[7] Which has been "very helpful" for addressing global monitoring missions, given the plethora of societal applications that exist.

Expanding upon the architectural advancements of SR, DA is renowned to be one of the most "practical ways to enhance model performance" [52], without incurring additional computation cost in the predictive phase. Currently, it is deemed that a variety of DA methods [53, 54, 55] have been proposed for "high-level" [52] computer vision tasks[8]. Whereas for "low-level" [52] computer vision tasks, including SISR, the application of DA methods has been "scarcely investigated". Such that only "simple geometric manipulations with traditional SR models and a very shallow learning-based model" have been studied at this time. Therefore, as DA is respected as an effective way to improve the performance of DL models [59], this study proposes to explore the application of geometric self-ensemble [60], extended with a randomised, state-of-the-art translation scheme [61], in the predictive phase of a state-of-the-art DCNN model, purposed to the SR of multispectral satellite images. Notably, this study focuses on enhancing the spatial resolution of Sentinel-2 satellite [62] imagery, which is synthetically observed at lower evaluation scale(s) (80→40m), in correspondence with the capacity of the available hardware.

1.1 Motivation

The European Space Agency (ESA) [63] is providing a wealth of research regarding Earth observation, to "new horizons" [41], through its Copernicus program [64] under Sentinel satellite missions [34, 36]. Each mission focuses on "different aspects" [41] of data extraction, through RS monitoring operations, which target the Earth's atmospheric, oceanic, and geological conditions [65]. In consideration of the series of Sentinel satellites available, this paper specifically focuses upon the Sentinel-2 (S2) mission (see *Figure 7*). As a supporter of "new generation" [19] satellite multispectral sensors, S2 is dedicated to the acquisition of "high-resolution optical imagery" [34], supporting multiple spectral bands that vary by spatial resolution [19][9] (see *Table 1*), and have high "radiometric and temporal resolution" [18] compared to other, similar instruments. The mission comprises a constellation of two identical, "polar-orbiting satellites" [62], namely S2-A and S2-B [18], that share the "same sun-synchronous orbit, phased at 180 degrees to each other" [62], to thus decrease their "repeat and revisit periods" [18, 66] and capture multispectral images with "13 spectral bands every five days at the equator" [36]. Sentinel-2 significantly contributes to Copernicus themes akin to "climate change, land monitoring, emergency management, and security" [66][10].

[8] Concerned with object recognition [56], object tracking [57], and human pose estimation [58] focuses.
[9] Of 10m, 20m, and 60m ground-sampling distances.
[10] Such that the mission requirements of S2 are tailored to a choice of priority services, concerned with natural hazard management, European land use, European land cover state and changes, forest monitoring, food security, global change issues, and humanitarian aid crises [67].

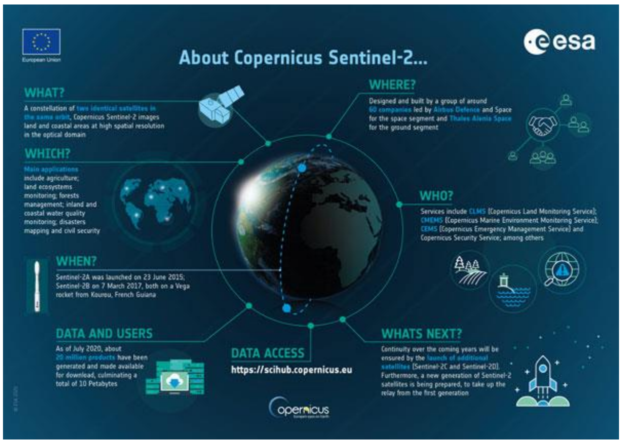


Figure 7: A Sentinel-2 mission infographic, highlighting the details and achievements of the mission after its first five years of operations [62].

Table 1: Tabularised wavelengths and bandwidths of the three spatial resolutions sponsored by the multispectral instruments of Sentinel-2A and Sentinel-2B [68].

Spatial Resolution (m)	Band Number	S2A		S2B	
		Central Wavelength (nm)	Bandwidth (nm)	Central Wavelength (nm)	Bandwidth (nm)
10	2	492.4	66	492.1	66
	3	559.8	36	559.0	36
	4	664.6	31	664.9	31
	8	832.8	106	832.9	106
20	5	704.1	15	703.8	16
	6	740.5	15	739.1	15
	7	782.8	20	779.7	20
	8a	864.7	21	864.0	22
	11	1613.7	91	1610.4	94
	12	2202.4	175	2185.7	185
60	1	442.7	21	442.2	21
	9	945.1	20	943.2	21
	10	1373.5	31	1376.9	30

Despite its recency, S2 mission data has already been "extensively used" [18], thanks to the ESA's "open data initiative" [34], that enables the data assembled by the S2 satellite pair to be "freely accessed". Thus, allowing researchers and services to purpose such data for a dynamic range of applications. Also, given the quality and "world-wide coverage" [18] of the mission, assumes S2 to be an "important tool" for both present and future Earth observation operations. It is the importance, current applications, and the resulting relevance of S2 mission data, which motivates the work proposed, alongside other prior and current works alike [18][11]. To note, the contributions of this work are also transferrable to SR studies surrounding other similarly operating satellite missions [69, 70], given the genericity it sponsors.

1.2 Research Hypotheses

Following from the contextual background of the work proposed, this paper engages the following hypotheses:

- Can state-of-the-art data augmentation techniques be applied to state-of-the-art deep learning super-resolution models, to further advance the spatial resolution exhibited by Sentinel-2 satellite imagery?
- To what extent, if any, does a combinative use-case of state-of-the-art data augmentation techniques have on enhancing the spatial resolution of Sentinel-2 satellite imagery?
- If a state-of-the-art deep learning model incorporates image granulation into its approach to super-resolution, does the sequence in which data augmentation techniques are applied and that the images are decomposed, impact the peak spatial resolution attainable by the super-resolution model?

[11] In knowing that not all spectral band images "are available at the same spatial resolution" [41].

1.3 Research Objectives

Parallel to the hypotheses formulated for the work proposed, this paper contributes to the abovementioned study areas, as such:

- Identify whether the low-resolution spectral band images issued by the Sentinel-2 mission can be spatially enhanced, via data augmentation techniques, for supporting more detailed and accurate information extraction.
- Identify whether a combinative use-case of state-of-the-art data augmentation techniques, can further enhance the spatial resolution achieved by a state-or-the-art super-resolution model.
- Identify an optimal configuration for the elected data augmentation techniques, suited to attaining peak spatial detailing of the spectral images resolved by the super-resolution model purposed.

1.4 Thesis Structure

The proceeding chapters of this paper are organised as follows. In Chapter 2, the backgrounding focuses of the study are explained and further explored, with developments led into DL, SR, and DA settings. Therein, also reviews the innovative and state-of-the-art contributions to the relevant fields. Then, Chapter 3 presents the proposal put forth for the study, in greater detail, and the approach devised for the study's undertaking is also identified and justified. Chapter 4 then focuses upon the implementation and appliance of DA techniques, for a state-of-the-art deep learning model that is identified in the preceding section. Thereafter, Chapter 5 elucidates the numerical findings of the study, and derives the implications of applying DA techniques to DCNN models, tailored to the enhancement of LR satellite imagery. Supplementary to the prior evaluation, Chapter 6 encapsulates all concluding discussions regarding the discoveries and limiting aspects of the study, as well as prospects for further investigation. Lastly, Chapter 7 consults the adherence demonstrated towards the study's progression, particularly in respect of the approach to software development. Where rationale is provided for any design and implementation challenge encountered, alongside aberrations from the initial proposal.

Chapter 2 | Related Research

2.1 Background

In this section, the backgrounding focuses of the study are clarified and explored further, to provide a supplementary understanding of the notions employed by the work proposed, foregoing the literature review.

2.1.1 Remote Sensing

Generically, RS can be characterised as the process of "acquiring information about an object, area or phenomenon from a distance" [71], as opposed to being in "direct physical contact" [72, 73] with it. However, more relevant to the study in question, RS typically refers to the acquisition of information regarding the conditions of the "Earth's surface (land and ocean), and atmosphere" [74], via sensors onboard "airborne or space-borne platforms", like that of satellite technologies. In providing a "wealth of data about Earth systems" [75], information attainment is achieved by "detecting and recording" [73] the "reflected or emitted electromagnetic energy" (see *Figure 8*) of a targeted surface area, in the field-of-view (FOV) of "one or multiple remote sensing instruments" [72]. Electromagnetic radiation is normally applicated as an "information carrier in RS" [74], given its properties, that enable it to propagate information concerning the distance between an instrument and a phenomenon, as well as the direction, intensity, wavelength, and polarisation of the radiation [72]. Collectively, these measurements can offer "positional information" of phenomena and indications as to identifying the properties of Earth's "surface materials".

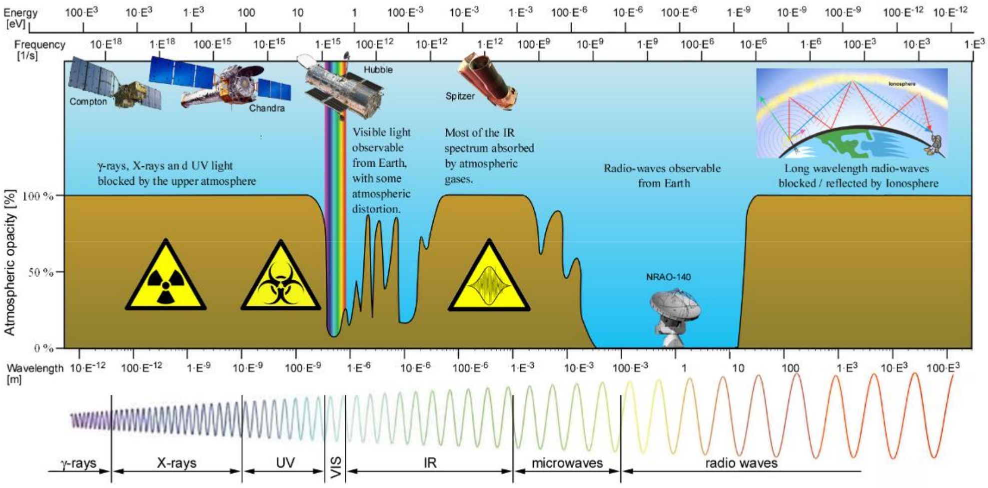


Figure 8: Visualisation of the electromagnetic spectrum and its various domains, featuring depictions of observation satellites at their respective detection ranges in the spectrum. The abscissa outlines the several modes of reference, given as wavelength (m), energy (eV), and frequency (s^{-1}) [76].

The interaction between a sensing instrument and the Earth's surface can be distinguished by two traditional modes of operation, namely active and passive [71, 72, 74] (see *Figure 9*), which differ in the "source of the energy from which information is gathered" [71]. Simply, active sensors generate their own source of energy to "illuminate objects" [72] on Earth's surface, whereas passive sensors utilise ambient energy such as the solar radiation sourced from the Sun, to "illuminate Earth's surface". For many of Earth's observation satellites[12], their onboard sensor instruments operate passively [72], which is also the common operating mode for Sentinel-2's MSI [81], in which collects the "sunlight reflected from the Earth".

[12] Such as Landsat [77], SPOT [78], GeoEye [79], and WorldView [80].

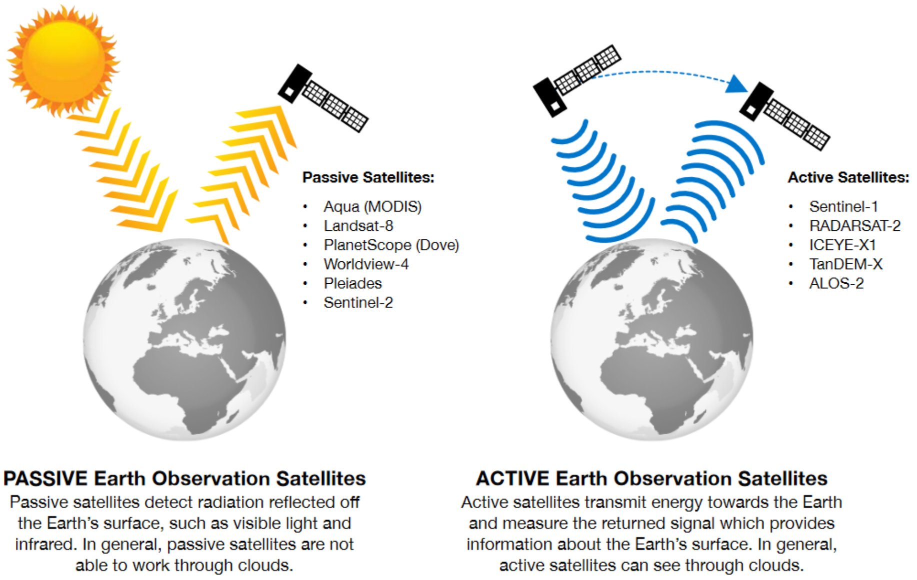


Figure 9: A satellite sensor instrument infographic, highlighting the distinction between passive and active sensor observations [82].

2.1.2 Multispectral Imagery

The anthology of information over a "larger number of wavelength bands" [83][13], is referred to as "multispectral or hyperspectral data" (see *Figure 10*). Optical imaging systems typically utilise the "visible, near-infrared, and shortwave infrared spectrums" [72] (see *Figure 8*) to generate imagery from the aforementioned data types and others alike, including "panchromatic" representations. These imaging systems are commonly onboard many of Earth's contemporary observation satellites [84], such as Landsat [77], WorldView [80], and this study's focus, Sentinel-2 [81][14]. This technology is not new-found however, with the first multispectral satellite, Landsat-1, being launched in 1997 with "four spectral bands" [72].

[13] Otherwise known as spectral bands.
[14] Given that each satellite operates with at least one MSI.

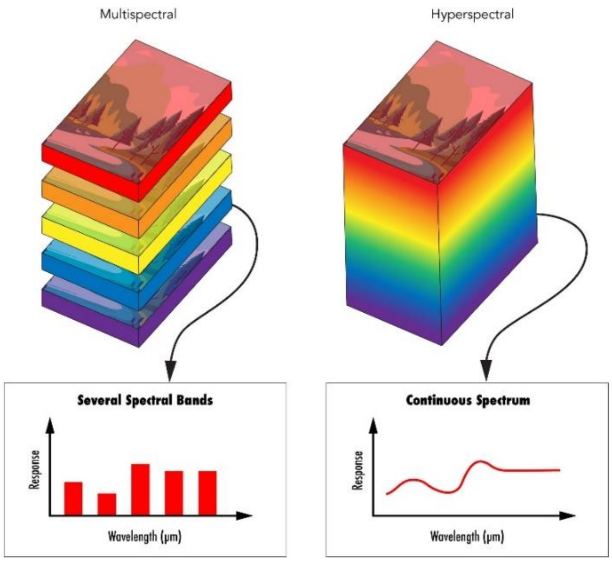


Figure 10: Visualisation of multispectral and hyperspectral imaging data, depicted as comparable image stacks, in which each's images are taken in several and many distinct spectra [85].

Simply, a multispectral image can be characterised as an assembly of "grey-scale images" [86], where each of the images corresponds to a "specific wavelength or wavelength band" in the electromagnetic spectrum. Such that every spectral band comprising the composite can be displayed "one band at a time", as a grey-scale image, or combinatorically as a "colour composite image". The resulting rendition can therefore be understood as a "multilayer image" [72], comprised of both the intensity and spectral (colour) information of the "targets being observed"[15]. Unlike a conventional RGB-colour composite image, the wavelength range of spectral bands can be "extended beyond" [86] the visible spectrum for multispectral images, from Ultra-Violet (UV) to Near Infra-Red (NIR) wavelength ranges, thus allowing "more information" to be propagated in comparison [87]. With respect to Earth observation missions, by manipulating the operational wavelengths of a given MSI, it is possible to "reveal features not otherwise easily discernible" [88] and provide images with "much higher spatial resolution" [89] than their RGB and hyperspectral counterparts.

2.1.2 Deep Learning

Deep learning is recognised as being a "new research direction" [33] in the field of machine learning (ML), which as a subfield, subgroup, and branch of the domain [90, 91, 35], attempts to learn "high-level abstractions in data" [90] through the adoption of "hierarchical structures" [92, 90], namely Artificial Neural Networks (ANNs). Being revolved around the ANN computing paradigm, DL is subsequently inspired by the "functioning of the human brain" [93]. Where like the human brain, DL is orchestrated by a series of "computing cells or neurons" that separately perform a "simple operation", and collectively "interact with each other" to resolve to a verdict or decision. As DL algorithms focus on "learning data representations" [94], as opposed to being purposed as "task-specific algorithms", feature hierarchies of datasets can be learned; with features from "higher levels" of a given hierarchy being formed by the "composition of lower-level features". By high-level features, one refers to a

[15] Or area (FOV) respectively.

feature that "hierarchically depends on other features" [95]. When considering the imaging nature of this study, a hierarchy as such implies that a DL algorithm will "learn its own low-level representations" from a given image, and then be capable of constructing depictions that depend on said low-level representations; this process then recurs in succession through to the "higher levels", or layers, of the network. Therefore, a DL class of algorithm or network can be characterised as a "cascade of multiple layers of nonlinear processing units" [94], or nodes (neurons), purposed to feature extraction and transformation. Where each "successive layer" in the network applies the output data computed by the prior layer, as its input data, for learning "multiple levels of representations" corresponding to the multiple "levels of abstraction" it is configured with.

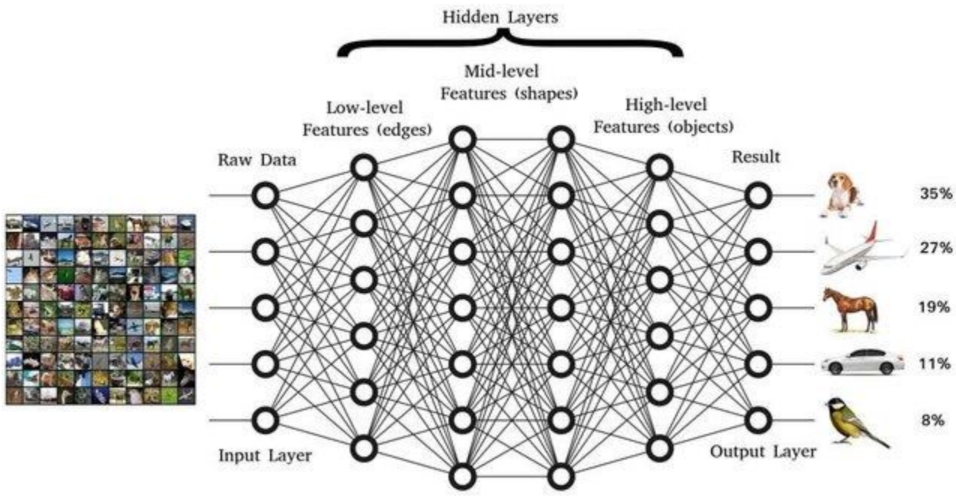


Figure 11: Visualisation of a generic deep convolutional neural network (DCNN) architecture, purposed for image or object recognition tasks. Therein, identifies the levels of feature extraction anticipated by a DCNN [96].

Appearing as a new field of research in 2006 [97], DL was originally received as "hierarchical learning", which anticipated many fields of research correlated with "pattern recognition" applications. Nowadays, DL is an "emerging approach" that has been "widely applied" to many traditional artificial intelligence (AI) domains[16]. Reportedly, there exists three significant reasons for why DL is now a "booming" [90] field of ML today, which are: the dramatically increased computational capability of central processing units (CPU's) [93], the rising affordability of computing hardware for regular consumers, and the contributions that have led to advances in ML algorithms [93, 90], overtime. Such that there is now an "enormous number" [94] of research publications that have been submitted to the field of DL. Aligned with this study's engagement, DCNNs specifically (see *Figure 11*), are actively being purposed for an array of RS applications [45, 46, 47] and have proven to enhance the performance of prior SR methods [50], given their capability to construct HR renditions of LR counterparts [49]. Said capability has led DL to demonstrate "prominent superiority" [35] over other ML algorithms, in various other research domains and application areas also.

[16] Such as natural language processing [98], transfer learning [99], semantic parsing [100], and this study's domain focus of computer vision.

2.1.4 Super-Resolution

Given the widespread availability of "high resolution displays" [101] and imaging instruments that still offer limited spatial resolutions[17], SR has recently received "substantial attention from within the computer vision research community" [102, 103]. Such that the subsequent, "rapid advancements" [101] of DL frameworks purposed to image processing applications, has enabled the procurement of "impressive results" in the SISR domain. Where DL frameworks have exhibited the capability of learning the mapping estimate of a HR image, from its LR counterpart. Which as previously acknowledged, is possible with DNN's, knowing that they are "able to learn abstract feature representations in the input image" that permit some degree of "disambiguation of the fine details" in the HR output. Thereby, in the field of SISR, an "end-to-end-mapping" [18] is typically learned and addressed in a "fully supervised manner" [101][18], via LR-HR image pairs, which is a "more powerful" [102] approach to SR led by learning or example-based methods [18, 33]. Where a DNN, specifically a DCNN [32], is deployed and purposed as an image "upscaling function" [101][19]. Expanding upon the definition provided for SR methods prior, one should acknowledge SISR as an "undetermined inverse problem" [28, 102], that entails the estimation or recovery of a HR image from a LR observation of the same scene; usually in cooperation with "digital image processing and ML techniques" [51]. In which the aforesaid mapping between low- and high-resolution images is learned, to recover (estimate) the "missing high frequency details" [29] of the LR image. Such that the presence of "high frequency components" [104] in a given image, increases, whereas the presence of "degradations" like artifacts[20] [35], decreases (see *Figure 12*).

[17] Due to "several theoretical and practical restrictions" [28].

[18] Otherwise regarded as supervised-learning.

[19] Generally, approaches to SISR can be classified into three classes, namely: interpolation-based methods, reconstruction-based methods, and learning-based methods [33, 51]

[20] Artifacts and other degradations alike, are commonly caused by the imaging process of the respective instrument, to note.

Figure 12: Visualisation of the Deep Sentinel-2 (DSen2) super-resolution model, performing 2× up-sampling on real Sentinel-2 data. From left to right: true scene RGB in 10m ground sampling distance (GSD) (spectral bands B2, B3, B4), native 20m GSD spectral bands, and the super-resolved 20m GSD spectral bands (B12, B8a, and B5 as RGB) up-sampled to 10m GSD [18].

However, as SISR is an "ill-posed problem" [102, 103, 105], existing SR technologies still, cannot always yield "satisfactory reconstruction results", which is why the research area remains to be "notoriously challenging" [35, 102] and of significant interest to both academia and industry [103]. Especially as the quality of a reconstructed SR image also "greatly affects the accuracy of other computer vision tasks" and studies[21]. Nonetheless, learning-based methods have "emerged as an efficient solution" [29] to the spatial resolution enhancement problem [106], given their "strong capacity" [35] in extracting high-level abstractions from images, that "bridge the LR and HR space". Where "several quintessential methods" [28] have failed, due to two recognised drawbacks: the first of two being the "unclear definition of the mapping" [35] operation between the LR and HR spaces, and the other being the "inefficiency of establishing a complex high-dimensional mapping" operation, given the vastness of the data being handled[22]. The first effort contributed to the field of SR can be traced back to as early as 1984 [107], where the term 'super-resolution' was then later coined in 1990 [108]. Today, since the advent of SRCNN [109] in 2014, SISR has begun to "usher in its pioneering work in the field of in-depth learning" and has made substantial improvements both quantitatively and qualitatively.

[21] Concerning image classification, image segmentation, object detection, and this study's focus of Earth observation monitoring [41].
[22] Examples of said methods are non-uniform interpolation, frequency domain, and machine learning-based reconstruction approaches [28].

2.1.5 Data Augmentation

Commonly, DA targets the enlargement of a given dataset "to address gaps in data representation" [110] and to "minimise the problem of overfitting" [111]; where DA can be used to improve "model performance and reduce generalisation error" [112], through enhancing the "ability of the model to generalise". Typically, relatively small datasets are enlarged by "applying transformations to its samples to create new ones" [113], which allows for a broader variation of "image contexts to train the model" with[23]. As an example of instance-based augmentation [114], the transformations typically concern geometric, colouring, noise injection, deformation, translation, brightness, and smoothing operations in conventional SR tasks [115]; with flipping, cropping, rotating, and scaling operations [113] being the most common. Meanwhile, many other strategies for enhancing a models generalisation capability focus on the "model's architecture itself", which has led to the existence of more complex architectures[24].

Figure 13: Depiction of a large collection of data augmentation (DA) techniques supported by 'DeepAugment'; a DA solution, purposed to machine learning (ML) applications [119].

However, DA can also be applied in the predictive phase of a given model, hence Test-Time Augmentation (TTA), for obtaining "greater robustness, improved accuracy, or estimates of uncertainty" [120]. In which, TTA entails "pooling predictions" from several transformed versions of a given input image, in the obtainment of a "smoothed prediction", representing a de-noised output. Given that TTA is "easy to use" and "simple to put into practise", it is regarded as one of the most "practical ways to enhance model performance" [52], hence its popularity in high-level vision tasks [56, 57, 58]. As well, TTA[25] has been present for a "long time" in the field of DL, such that in 2012, a standard "evaluation protocol" [122] was derived from averaging the predictions (ensemble) of an "image classification model over random crops and flips of test data". Though, transforming data before

[23] This helps to improve a model's "invariance to spatial transformations" in the predictive phase.
[24] Such as: AlexNet [116], VGG-16 [117], and ResNet [118], to name a few.
[25] Otherwise regarded as enhanced prediction [121].

inference has "received less attention" [123] than enhancing the diversity of a training dataset, especially for low-level computer vision tasks. Given the SISR emphasis of the study, TTA can be characterised as a technique that utilises an existing network or model, to super-resolve a series of transformations of the same image, before each's transformation can then be "undone" [41], to obtain an ensemble image representing the aggregate of all transformations, as the "final super-resolved image". This technique aims to reduce the presence of "noise patterns and artifacts" [124] unresolved by a baseline model, through the "smoothening process" advocated by the ensemble of the transformations. Thus, maximising the "potential performance" [59] of a related model.

2.2 Related Work

Enhancing the spatial resolution of RS multi-resolution images has been investigated and addressed for an array of image types and sensory instruments already[26]. As one of the most active areas of research since the "seminal work" [104] of [107] in 1984, many SR techniques have been proposed in the "last two decades" [104], as per [50, 132]; featuring approaches that have advanced from the "frequency domain to the spatial domain", and from a "signal processing perspective to a machine learning perspective". In the following passages, an overview of the techniques and innovations contributed to the SR domain are revealed, in the determination of the state-of-the-arts purposed to this study's investigation.

2.2.1 Classical Super-Resolution

Classical approaches to SR[27] aim to generate "high-quality HR images" [133] from a given, single LR input image, through exploiting "certain image priors". Each of which poses as information regarding the prior state of an image. Corresponding to the image priors known, classical approaches to SR can be categorised as prediction-based, patch-based, edge-based, and statistical-based methods [50, 133]. To note, classical SR methods are briefly reviewed to entertain the complete development progression of the SR domain.

2.2.1.1 Prediction-Based Methods

The preliminary efforts in achieving SISR were "based on prediction" [50], where HR images are generated from LR images through "a predefined mathematical formula without training data" [133]. The first of such efforts was recorded in [134] and was based on Lanczos filtering, a "Fourier method of filtering digital data" via "sigma factors" [50][28], to smoothly interpolate data comprising a digital image. In later periods, a similar frequency-domain approach was introduced [107] and purposed to "image resampling" [50], which combined "multiple under-sampled images with sub-pixel displacements to improve the spatial resolution" [135]. Prior to this contribution, a cubic convolution algorithm [136] was also purposed for resampling image data, and the results demonstrated that the

---

[26] Such as Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) [125, 126], Moderate Resolution Imaging Spectroradiometer (MODIS) [127, 128], Visible Infrared Imaging Radiometer Suite (VIIRS) [129, 130], and more recently, MSI's [18, 131].

[27] That are otherwise regarded as conventional methods [50].

[28] Otherwise known as sigma-approximation.

predictive method was "more accurate than the nearest-neighbour algorithm and linear interpolation method" [107, 137]. However, despite being much more efficient, relative to "storage and computation time" [107], and accurate compared to the methods abovementioned, the algorithm was deemed performatively inferior to the cubic spline interpolation method [136]. Expanding upon the works of [107], which did not factor "blur in the imaging process" [50], the authors in [138] used "approximate knowledge of the imaging process", to compute "relative displacements for image interpolation" [50] when a constant sampling rate was configured, to deblur a "single input image" [138]. Leading to the authors' claim that SR "reduces to deblurring", upon their iterative algorithm being applied to individual images without an "increasing sampling rate".

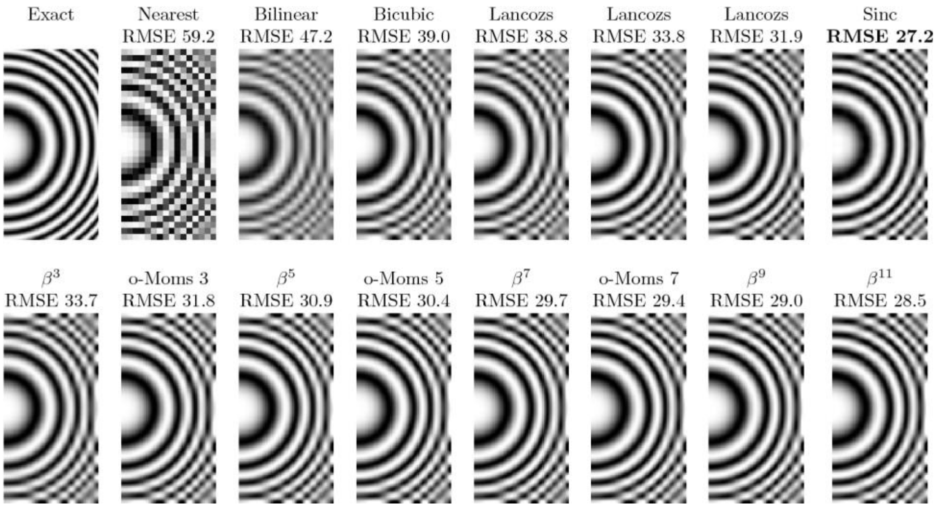


Figure 14: Depiction of interpolation-based approaches to super-resolution, with a smooth data (image) example [139].

Prediction-based methods including bicubic interpolation and non-uniform interpolation [140, 141], are accepted to "estimate the intensity" [103] of a point or pixel in a digital image, using the information of "adjacent pixels". Typically, interpolation-based methods entertain three algorithmic operations, namely registration, interpolation, and deblurring [51, 141, 142]. Although said methods are easily calculated, "very speedy, and straightforward" [103, 105], they are renowned to experience "accuracy shortcomings", from the generation of "jagged artifacts" [103] and "excessive smoothing" of detail (see *Figure 14*).

2.2.1.2 Patch-Based Methods

Patch-based approaches to SR[29] were acknowledged for their "fast computation and outstanding performance" [35] offerings; considered as one of the "more powerful approaches" [102] to SR, patch-based methods aim to establish a "complex mapping between low- and high-resolution image information" [102] from training examples and data[30]. Given a set of LR-HR training image pairs, a series of patches can be "cropped from the training images to learn mapping functions" [133]; many methods based on example-pair images depend on LR patches having a corresponding, HR counterpart

[29] Also referred to as learning- or example-based methods.
[30] Which typically anticipates the utilisation of ML algorithms [105].

[102]. However, exemplar patches can be "generated from external datasets, the input image itself, or combined sources" [133].

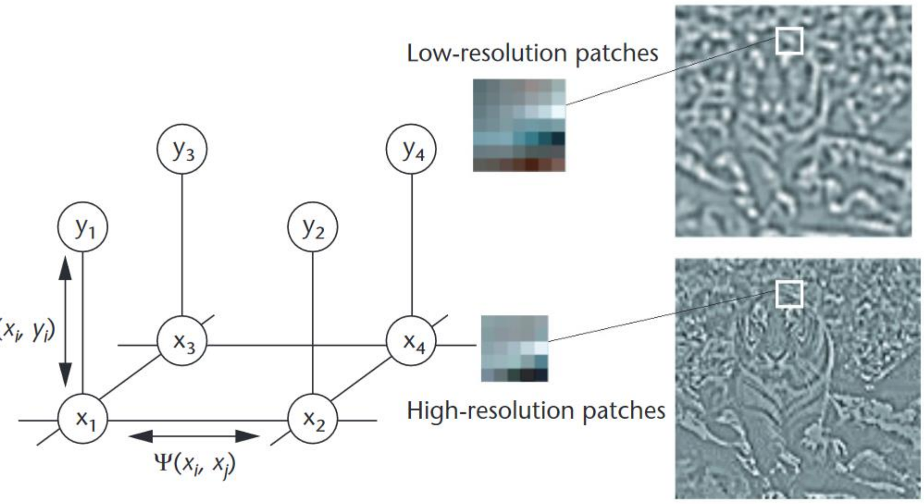


Figure 15: Depiction of the Markov Random Field (MRF) network model. Where the low-resolution image patches at each node y_i, are the observed input. Whereas the high-resolution image patches at each node x_i, correspond to the quantity of closely related input patches to estimate any given high-resolution patch extraction choice [143].

One of the earliest works, namely the Markov Random Field (MRF) approach [143], adopted "various patches within the training set" [50] as training patterns. Which enabled the generation of "detailed high-frequency images", from the utilisation of texture information embedded in each of the extracted image patches (see *Figure 15*). Thereon, in the works of [144] the authors exploited similar "local geometry between LR and HR" [105] patches, for generating HR images derived from "local patch features" [50]. Established as Locally Linear Embedding (LLE) [144], their method was developed upon the assumption of the "closeness of complex constructions between LR and HR peers" [145][31]. Differentiating from this proposal, in [147] the authors exploited "patch redundancies across scales within the image" [102] to approach SR, which anticipates the "reoccurrence of geometrically similar patches in natural images" [50], to identify the elitist pixel values. Meanwhile, in the preceding works of [148], the authors proposed hallucination algorithms[32] that enable the local features within an LR image to be "recognised" and extracted, before being used to "map the HR image" [50].

2.2.1.3 Edge-Based Methods

Edge-based approaches to SR typically utilise "edge smoothness priors" [50] to up-sample LR images, which as "important primitive image structures" [133], pose a significant influence on the visual perception of image quality. Whereby, in [149] a generic image prior described as the "gradient profile prior" is used to smoothen the edges comprising an image, to "achieve super-resolution in natural images" [50]. In which both small- and large-scale detailing is "well recovered in the HR image" [149], and artifact generation is avoided by "working in the gradient domain" (see *Figure 16*). To reconstruct

[31] This method was particularly influential on the existence of the self-similarity paradigm [146].
[32] Which they proceed to regard as "recognition-based reconstruction techniques".

texture detailing of realistic perceptual qualities while "avoiding edge artifacts" [50], the authors of [150] blend an "edge-directed" [50] SR algorithm based upon a gradient profile prior [149], with learning-based techniques that provide the benefits of "detail synthesis". This reports to obtain "good results", both subjectively and quantitively. Advancing from this area of work, in [151] the authors utilise "filter banks" in the search for similar image patches, based on a "local self-similarity observation" [50] which performs fewer "nearest-patch computations" to synthesize an image. Given this relation, the proposed method is capable of reconstructing perceptually convincing edges, efficiently, whilst exhibiting a reduced number of artifacts like "jaggies and ringings" [151]. However, "fine-detailed clustered regions" of a given image are reportedly not reconstructed with a realistic appeal, and instead appear "somewhat faceted". Thus, rendering the performance of the method, poor. Since image priors are "primarily learned" [133] from edge features, HR reconstructions typically exhibit "high-quality edges" and a limited number of artifacts. However, as discovered, edge priors are less successful for modelling "high-frequency structures" such as textures, where the results are seemingly less convincing.

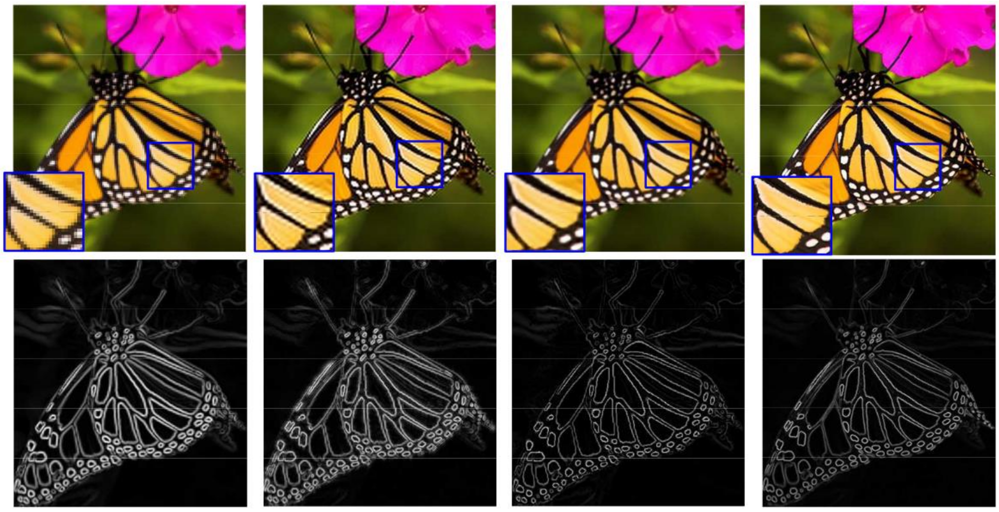


Figure 16: Visualisation of a high-resolution image reconstruction sequence. From left to right: a low-resolution image (using nearest neighbour interpolation) and gradient field of its up-sampled image (bicubic interpolation), the result of back-projection and its gradient field, the result of the gradient profile prior method and its transformed gradient field, and the ground truth image and its gradient field [149].

2.2.1.4 Statistical-Based Methods

Statistical-based methods that exploited the features and properties of images as priors, to predict, restore, or recover HR images from LR counterparts [133], were also proposed as classical SR approaches. In [152] the authors exploit the "sparsity property of large gradients in generic images" [133], to reduce the "time complexity of training and testing" [152] for Kernel Ridge Regression (KRR) [153]. In which the authors propose a "sparse solution" [152] that unites the notions of Kernel Matching Pursuit (KMP) [154] and gradient descent [155], to enable KRR to learn a "mapping function from the image example pairs" [50], more efficiently. Which then as a "regularised solution" [152], allowed KRR to also yield an advanced generalisation capability. Continuing in the direction of regularisation, the authors of [156] combine "adaptive regularisation and learning-based pair matching" techniques, to restore HR web-based images and video sequences from "compressed LR measurements with different content and degradation levels". Which demonstrated the ability to eliminate "compressed artifacts" whilst preserving and enhancing "high-frequency details", post-compression; thus, increasing the "resolution and perceptual quality" of images and videos alike. Meanwhile, in the works of [157] and [158] the authors propose "sparse signal representation" methods to perform ISR, that complies with the concepts surrounding compressed sensing [159]. This demonstrated the "effectiveness of sparsity as a prior" for patch-based SR, in focus of both generic and facially-derived imagery (see *Figure 17*). To note, both methods focus on recovering the SR "version of a given low-resolution image" [157, 158], through their dependency on "patches from example images". Differentiating between each's contribution to the domain, however, the authors of [157] work directly with the LR training patches and "their features", whereas the authors of [158] instead, learn a "compact representation" for the image patch pairs to acquire the "co-occurrence prior", which enables the algorithm to operate more quickly by comparison.

Figure 17: Visualisation of two example images, a flower, and a girl, magnified by a scale factor of three. From left to right: the input low-resolution image, the result of bicubic interpolation, the result of neighbour embedding, the result of the sparse representation method, and the ground truth image [157].

2.2.2 Panchromatic Sharpening Super-Resolution

Panchromatic sharpening or "pan-sharpening" [160] can simply be defined as a "pixel-level fusion" [161] or SR technique, that is used to enhance the "spatial resolution" of multispectral (MS) images [162]. Where it is possible for MS images to be "sharpened" [163] by a higher-resolution PAN (single spectral band) image, in classical pan-sharpening, to enhance both "spatial and spectral resolutions" of the containing image data [164]. This procedure can otherwise be characterised as the "fusion of panchromatic and multispectral images" [165], where pan-sharpening techniques govern the transfer [166] or blend [18] of detail, from a HR panchromatic band to a series of LR bands (see *Figure 18*). As the authors of [167] report, pan-sharpening techniques pose two advantages to ISR, which are their "efficiency and applicability" in resolving HR renders of LR, MS images; hence why the methodology has been regarded as one possible option to "achieve a superior spatial resolution" [17].

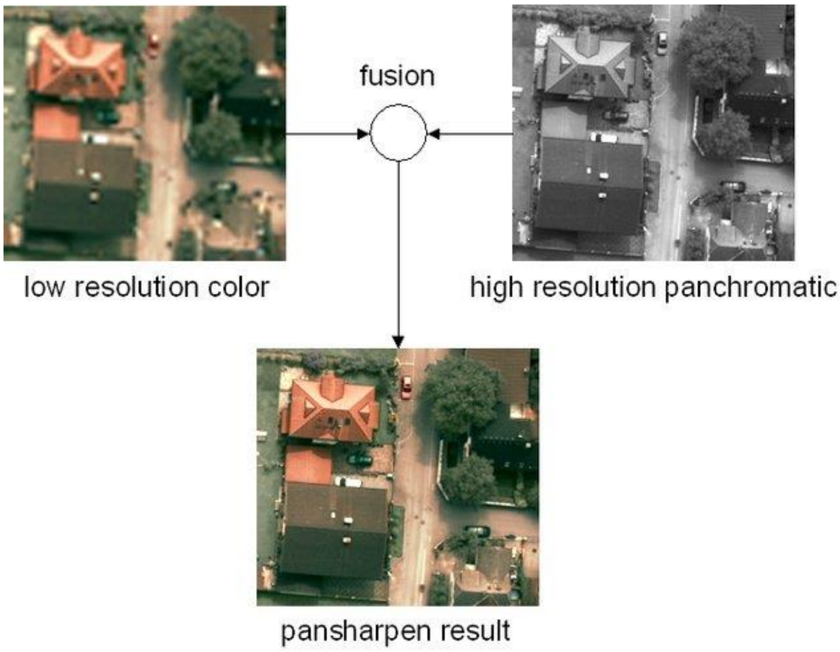


Figure 18: Depiction of the image fusion process entailed in the obtainment of a pansharpened image. Therein, demonstrates the combination or fusion of a low-resolution multispectral image with a high-resolution panchromatic image, to obtain one high-resolution multispectral image [168].

Relevant to this study's investigation, "many" [167] Earth observation satellites natively record panchromatic data, as is entertained by the Landsat [77], SPOT [78], and WorldView [80] series of satellites. Which naturally render pan-sharpening techniques "feasible" [17] for MS ISR. However, as Sentinel-2 [81] does not operate with an onboard panchromatic instrument, other solutions are "needed" [167] instead. Nonetheless, as "a lot of pan-sharpening algorithms" [17] have been proposed and developed for MS and HS images in the "last decades", there exists such solutions to alleviate this shortfall. Given the quantity of algorithms available, it is widely accepted that pan-sharpening methods can be classified as either component substitution (CS) or multi-resolution analysis (MRA) methods [164]. For the fulfilment of this study, these methods alongside those explicitly purposed for the Sentinel-2 mission are explored, where the latter is classified separately from CS and MRA methods to warrant clarity.

2.2.2.1 Component Substitution Methods

CS is renowned as the "most classical" [164] approach to pan-sharpening, that concerns the "projection" [169] or transformation of MS images into a "different colour space" [164], and substitution of one of the components in the LR, MS image, with one from the HR PAN image. This assumes that transformation separates the "spatial structure from the spectral information into different components" [169]. Where the MS image, once transformed, can be "enhanced by replacing the component containing the spatial structure" with the PAN images. Thereon, after "up-sampling" the image's other components, the MS image can then be "back-transformed" [164] to its original colour space; thus, achieving SR. Given the nature of this approach, it is understood that the "spectral fidelity" achieved by a pan-sharpened MS image is "closely related" to the association between the PAN image and the replaced component; where the "greater the correlation", the "lower the distortion" visible in the pan-sharpened image is.

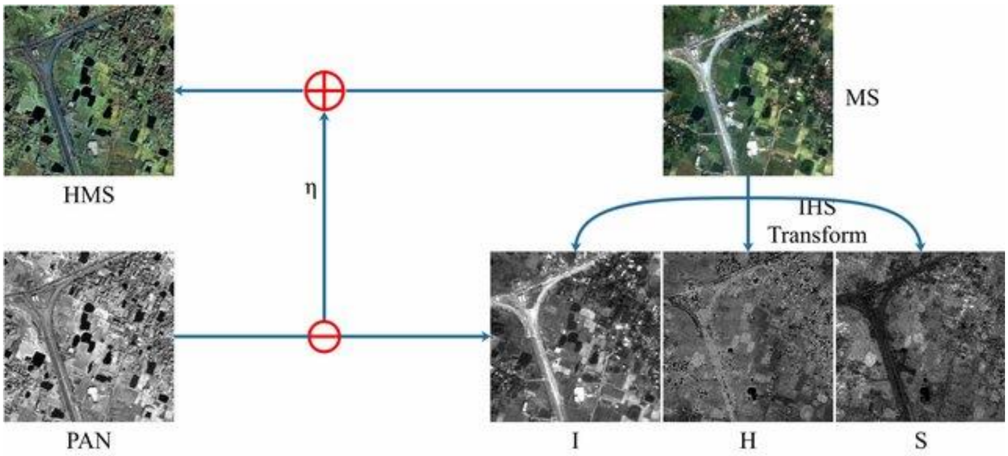


Figure 19: Visualisation of an intensity-hue-saturation (HIS) pan-sharpening method, replacing the intensity (I) component of the low-resolution multispectral image, with the PAN image [170].

Proposed by the authors of [171] was a "fast and simple transformation" based upon intensity-hue-saturation (IHS) image fusion principles. That as an early contribution to the field, was only purposed for "three bands imagery" [164]. Extending from this work, the authors of [172] then proposed a "generalised IHS" [164] method, which incorporated the support for NIR spectral bands, in addition to the PAN and RGB bands supported by IKONOS satellite imaging [173]. Advancing from these preliminary works, in [174] an IHS-based pan-sharpening approach to "spectral fidelity improvement" [164] was proposed, implementing "discrete ripplet transform" and compressed sensing principles, to reconstruct the "intensity component" [174] of the MS image (see *Figure 19*) and acquire "multi-scale sub-images" from the PAN image. As the authors report, this method "outperforms" five competing, state-of-the-art methods on three satellite derived imaging datasets; such that it overcomes spectral distortion whilst improving the spatial resolution of MS images, "significantly". Differing by their approach to colour space transformation, many other "popular" pan-sharpening approaches have been submitted as CS methods as well[33]. However, due to the lack of "unique transforms" [169] for obtaining the optimum component for PAN "substitution", methods focused around its "adaptive estimation", namely adaptive CS [178], have since been proposed to alleviate the presence of "massive spectral distortions" [164] in the results. Even so, CS-based methods consistently "lead to unsatisfied spectral fidelity", which renders them sub-optimal for ISR.

2.2.2.2 Multi-Resolution Analysis Methods

Posed as the second class of pan-sharpening methods, MRA seeks to extract the "high frequency details" [164] from a PAN image, before 'injecting' them into the corresponding "up-sampled MS image". Comparatively, approaches of this nature are provably "less susceptible to spectral

[33] Including principal component analysis (PCA) [175], Gram-Schmidt (GS) [176], and Brovey transform (BT) [177], to name a few.

distortions” than their CS-based counterparts, which determines them to be more optimal for ISR. Derived from the ‘Amélioration de la Résolution Spatiale par Injection de Structures’ (ARSIS) [179] concept, MRA methods are respected as data “preservation” [169] techniques for LR, MS images, that through “spatial filtering”, can transfer the “information obtained from the PAN image” to acquire an improved, pan-sharpened version.

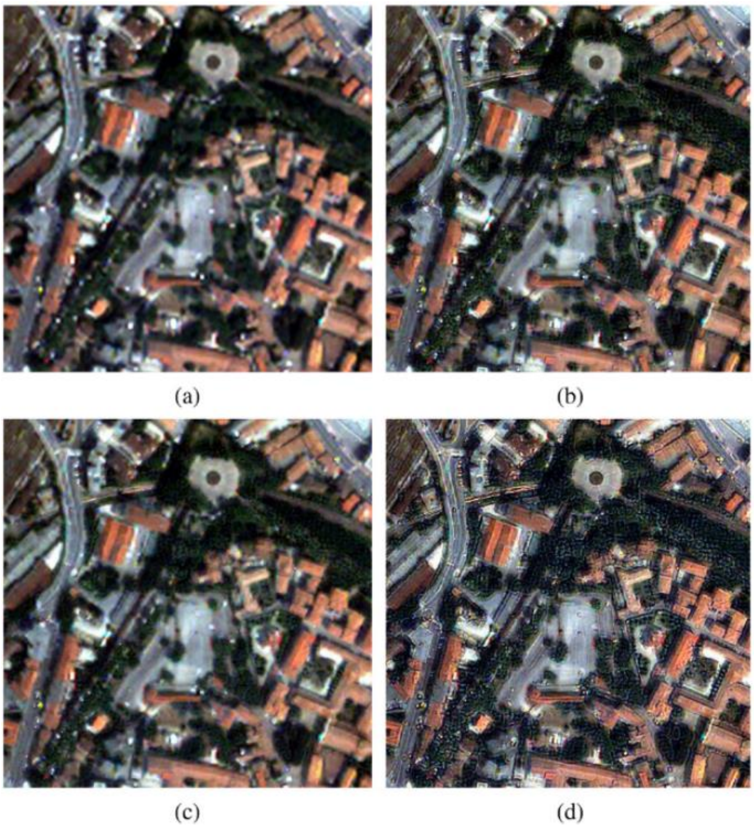


Figure 20: Depiction of a full-scale spatial enhancement of the ‘QuickBird’ image, by a series of image fusion algorithms. Annotated: (a) the original multispectral image bands resampled to the scale of the PAN image, (b) the product of curvelet transform fusion, (c) the product of Gram-Schmidt fusion, and (d) the product of à trous wavelet transform with a spectral distortion minimising model [182].

As a preliminary contribution to the MRA class of methods, in [179] a technique namely à trous wavelet transform (ATWT) was proposed, where a PAN image is firstly “decomposed into some wavelet planes” [164], before the high frequency details are then “extracted and injected” into the intensity or luminance component of the MS image. As the authors conclude, this concept led to the generation of “high spatial resolution multispectral images” [179], which align closely with images that the sensor itself would observe with the highest resolution. Thereon, in [180] a “novel” [164] method that could combine and preserve the “spectral-spatial information” of PAN and MS images well, was proposed, from the combinative use of “multi-resolution wavelet decomposition and the IHS transform” [180]. However, this method and others alike it [181] were acknowledged dependant on “redundant representations” [164], which disappointed MRA-based methods, as “massive spatial distortions” would be produced, depreciating the perceptual quality of pan-sharpened images. Considering this defect, an “improved method” based on non-separable or “curvelet transforms” (CT) [182] was then later proposed, for extracting spatial details with a “directional property” [164]. Which led to the attainment of “superior spatial enhancement” (see *Figure 20*). Beyond this approach, many other MRA-based methods were proposed for “reducing distortions and artifacts”, as in [183], and achieving high accuracy “image registration” [164], as in [184].

2.2.2.3 Sentinel-2 Adapted Methods

As Sentinel-2 does not have a “panchromatic band that covers most of the sensors spectral range” [18], it is deemed “necessary” [166] to purpose S2’s HR spectral bands for “generating super-resolution versions” of its LR counterparts; thus, fabricating a series of panchromatic bands. Which is notably a complex procedure, as is presented by many of the popular works explored in [185]. Entertained by this comparative study, one can understand that many “popular pan-sharpening methods have still been applied to S2 data” [166], and that there exists methods that have been “developed specifically for the super-resolution of S2 images”.

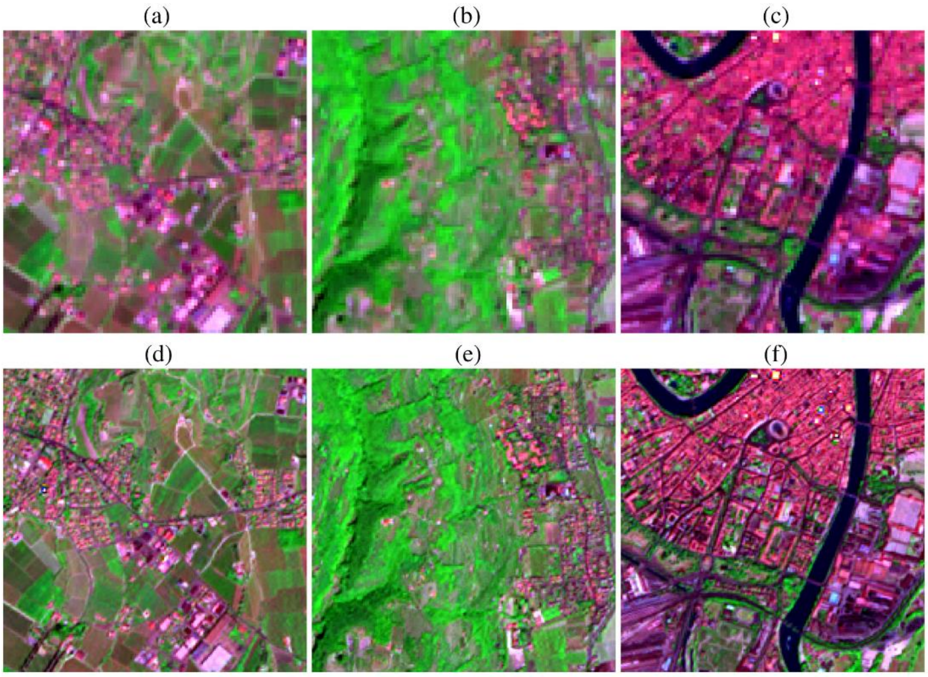


Figure 21: Depiction of the area-to-point regression kriging (ARTPK) results, for a studied Sentinel-2 image (spectral bands B12, B8a, and B5 as RGB) based in three sub-areas of Verona, respectively. Where: (a) – (c) represents the 20m GSD data, and (d) – (f) represents the 10m GSD (downscaled 20m GSD) results [187].

In another comparative study [186], the authors evaluate the performance of “21 different fusion algorithms” across three frameworks, when spatially enhancing the “narrow 20m VNIR and SWIR bands of the Sentinel-2 satellite”. For which, the authors use “heuristic” [18] methods to “synthesize” the panchromatic data from the 10m GSD, HR spectral bands. Despite the “spectral discrepancies” [186] encountered, the authors claim that “most” pan-sharpening methods were able to “spatially enhance the lower resolution data”. Meanwhile, in [187] the authors report some of the “best results in the literature” [18], for their proposed, Area-To-Point Regression Kriging (ARTPK) method (see *Figure 21*). Which in essence, performs “regression analysis” between spectral bands at a lower resolution, before applying the “estimated regression coefficients” to the HR input; this method extends the CS and MRA classes of pan-sharpening, in use of a band synthesis and band selection scheme [187]. Where in [188] the authors propose a series of amendments to “optimise” [18] the schemes aforesaid, that subsequently established compatibility with CS and MRA-based methods. Prior to this contribution, the performance of “four popular” [18] pan-sharpening methods were evaluated in focus of open water body monitoring [189], in attempt to sharpen the B11 SWIR band of the S2 dataset and acquire a HR normalised differential water index (NDWI) image. Further in this direction, in [190] five unique pan-sharpening methods were then utilised for enhancing the “resolution of the 20m bands” [18], to investigate the land-cover classification potential of pansharpened images, compared to the products of “naive neighbour up-sampling” methods. As the authors report, the former dramatically “improved the overall classification accuracy” [190].

2.2.3 Model-Based Super-Resolution

Observational approaches to SR “explicitly define a model of image degradation to be reversed” [166], relative to the targeted system for image acquisition. Where the relationship between an original HR image and observed LR image can be established [142], and then used to describe and negate the “assumed blurring, down-sampling, and noise processes” [18] entertained by the LR observation. Given this relation, model-based methods approach SR as an “inverse imaging problem”, for which many “variational” [166], Bayesian inference frameworks [18] have been proposed to address. As the nature of this problem is ill-posed, methods typically adopt “explicit regularisers” to regulate image priors; enabling HR renders to be resolved by “minimising the residual error” of a given model, or respectively, the “negative log-likelihood” of all spectral band images, “simultaneously”.

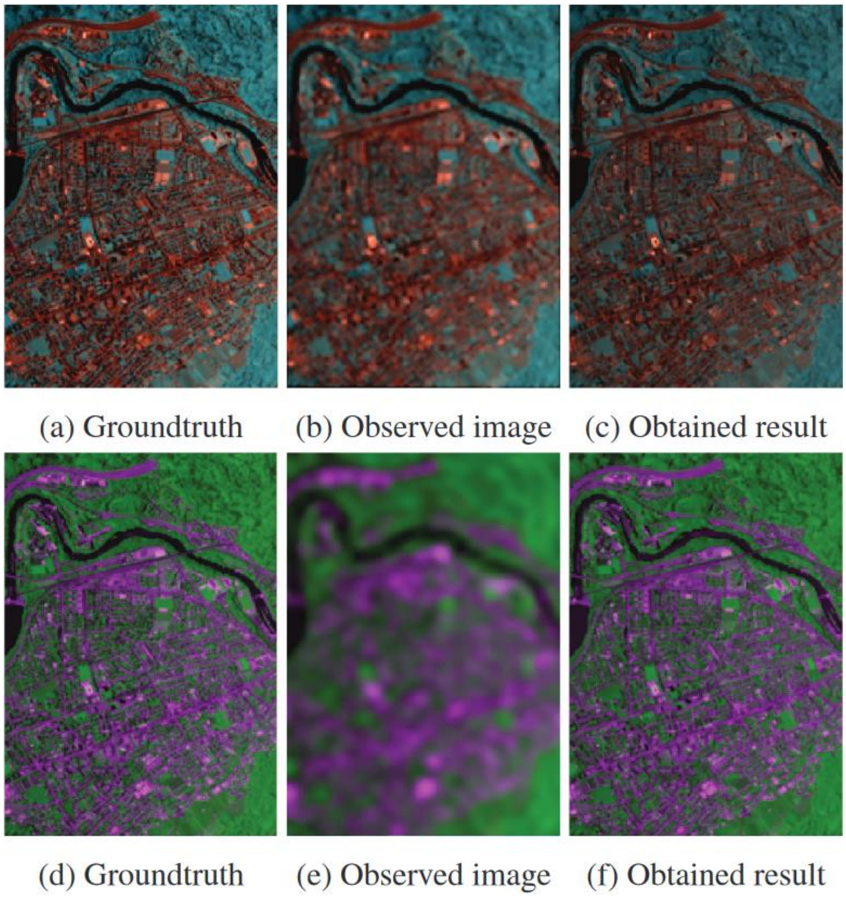


Figure 22: Depiction of the Superresolution MUltiband multireSolution Hierarchical (SMUSH) method results, obtained for simulated Sentinel-2 images. Where: (a) - (c) represents a false colour composite image, comprised of spectral bands B5, B6, and B7(20m GSD), and (d) - (f) represents a false colour composite image, comprised of spectral bands B1 and B9 (60m GSD) [193].

In the work of [191] the author introduced a "resolution enhancement" method[34] purposed for MS and HS images, capable of separating the "band-dependant spectral information" (reflectance data) of Sentinel-2 imagery, from the information branded common between all bands[35]. In which the model can be applied to super-resolve and "unmix" the LR spectral bands, to the extent that the bands are "consistent with those scene elements" [18], whilst preserving their "reflectance" [191] values and "sub-pixel details". Meanwhile, in [192] the authors adopt an observation model, namely SupReME, equipped with "per-band point spread functions" (PSF's) [18], to each cater for "blur and down-sampling per band" [192]; while a spatial regulariser "learns the discontinuities" from the HR bands and "transfers" the detail to the other, lower resolution bands. Which consists of two operations: dimensionality reduction, that infers the "correlation between the bands" [18], and a "contrast-dependant penalisation" of quadratic gradients, used to reduce smoothing across the discontinuities learned from the HR bands [192]. Similarly, the work of [193] also introduced an observation model[36] closely aligned with the SupReME model, except that it utilises an "edge preserving regulariser and a patch-based plug-and play prior", to promote the acquisition of self-similar images. The method entertains a hierarchical process, which sharpens the "medium resolution bands" (20m GSD), before

[34] Namely Superres.
[35] Namely the "geometry of scene elements".
[36] Namely SMUSH.

then the “coarse resolution” bands (60m GSD) of a given S2 image; reportedly, the method can accurately increase the spatial resolution of the lower spectral bands, “without introducing significant distortion and artifacts” (see *Figure 22*). Beyond the scope of the methods explored, exists other state-of-the-art Sentinel-2 frameworks[37]; all which share performance enhancements over classical and other pan-sharpening approaches to SR.

2.2.4 Deep Learning Super-Resolution

Various DL methods have been proposed and “developed over the years” [50] to combat the SR problem[38]; typically, approaches to SR have confronted the “prediction” [18] of HR images as a “supervised machine learning problem”. Where unlike the catalogue of techniques explored prior, the relation or mapping between the LR input to the HR output is not “explicitly specified”, but rather “learned” from exemplar data. Posed as learning-based methods[39], DL models are trained using both “low- and high-resolution images” [50][40], and have received substantial focus because of their “fast computation and outstanding performance” [105] offerings[41]. Providing that they can capture “more complex and general relations” [18] of the features that comprise image data; despite requiring “massive amounts of training data, and large computational resources”. Following the success of [116], CNN architectures pose to be the state-of-the-art for “many computer vision problems” [102][42], which rationalises why the following passages explore CNN contributions to SR, and their deeper, recent advances.

2.2.4.1 Convolutional Neural Network Super-Resolution

Historically, the first CNN[43] was proposed in 1998 [197] for the “classification of handwritten digit recognition” [36], using the profoundly recognised MNIST dataset [198]. Following its debut, the “power of CNNs” was seemingly ignored and neglected by imaging applications until the advent of AlexNet [116][44], in 2012. Which reported a “10% increase in accuracy” [36] over previous non-CNN-based models, for image classification tasks. Ever since, a CNNs purpose has been “extended to a series of problems in computer vision” [34], including SISR, where they have reportedly “excelled” within.

---

[37] Namely S2Sharp [194], MuSA [195], and SSSS [196].
[38] And more so in focus of the SISR problem.
[39] Otherwise regarded as example-based methods.
[40] Referred to as LR-HR pairs.
[41] Where DL-based SISR methods have demonstrated superiority in the domain.
[42] Including SISR.
[43] Namely LeNet-5, a five-layer CNN.
[44] An eight-layer CNN.

Figure 23: Depiction of the super-resolved results for a Sentinel-2 RGB composite. Comparing the products of the bicubic-interpolation model (as the baseline model) and the multispectral instrument Super-Resolution Convolutional Neural Network (msiSRCNN) model [199].

Proposed in [109] was the first SR technology to adopt a neural network (NN), namely SRCNN, which posed as a shallow "three-layer CNN" [33] that bicubically-interpolated an image before performing patch extraction and representation, non-linear mapping, and patch-wise reconstruction operations, to learn an end-to-end mapping between LR-HR pairs. Given the methods ability to "obtain higher PSNR" [33][45], SRCNN was then later applied to satellite imaging in 2016 [199] with "no changes to the networks architecture" [32][46]. Wherein, the model was instead "re-trained using Sentinel-2 images" as described in [109], for achieving a performance superior to the classical, bicubic-interpolation model[47] (see *Figure 23*). Thus, proving that CNN-based SR methods were "equally applicable to satellite imagery". In the same year, the authors of the SRCNN then proposed a new variant, namely "Fast SRCNN" (FSRCNN) [200], to achieve "fast training" [201] whilst maintaining the performance of its predecessor model; this was achieved via substituting the "up-sampling step" [32] with a deconvolution layer at the end of the network. Thereby enabling a LR image to be "inputted directly" [202]. Another variant of SRCNN, namely Multi-Channel SRCNN (MC-SRCNN), was also proposed in 2016 [203]. Which differentiates from SRCNN by accepting "multi-channel input images" [32], as opposed to one single-channel input; the method creates multi-channel input by applying a range of "interpolation algorithms and sharpening filters", during its data pre-processing phase. As the authors reported, multi-channel inputs enabled the model to "reconstruct more proper HR images" [203], that improved upon SRCNN "marginally as a result" [32].

2.2.4.2 Deep Convolutional Neural Network Super-Resolution

Since SRCNN, the importance of SR technology "cooperating with in-depth learning" [33] has been realised, upon establishing that "deeper and more complex architectures can lead to better results" [36]. Providing the theoretical works of [204], the authors identify this relation by increasing the "depth or width" [35] of a DNN's "solution space" and its layers within; allowing for more hierarchical representations to be obtained, "effectively", alongside a generalisation capability beyond "shallow

[45] The acronym of a widely used image quality metric: Peak signal-to-noise ratio.
[46] The SRCNN model then inherits the name msiSRCNN.
[47] This model was purposed as the baseline method of the study.

models" [204]. Irrespective of their "training difficulties", recent DL-based applications have showcased the "great power of very deep" NNs.

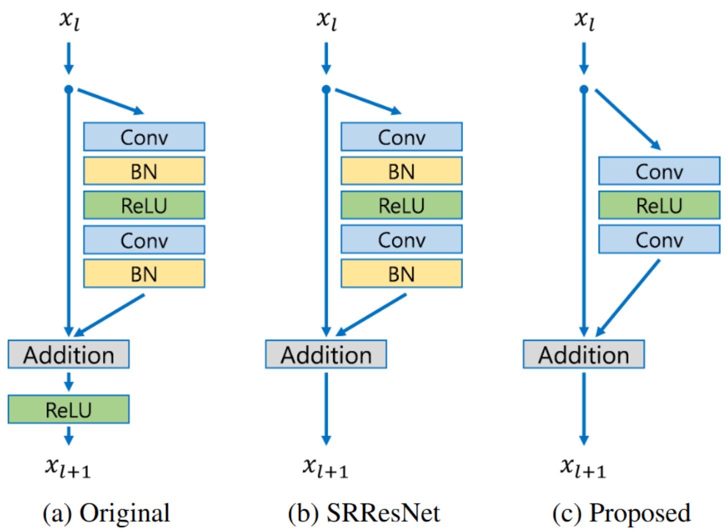


Figure 24: Visualisation of the residual block compositions for the (a) ResNet, (b) SRResNet, and (c) EDSR networks [211].

As the "first very deep model" [35] purposed to combating the SISR problem, VDSR in [205] was introduced as a 'residual network', which was used to train "much deeper network architectures" [59] in modelling the difference between "HR and LR images" [205] and achieving "superior performance" [59] over SRCNN. Inspired by VGG-net [206], VDSR's "20-layers" [36] were found "necessary for good performance" [32], in enabling the model to converge "much faster" than SRCNN and to be applied to "multiple scales" of images, with the help of "residual-learning" [205] concepts. Branching from this work, in [207] the authors "retrain" [166] VDSR[48] to perform well with Sentinel-2 data, which returns "promising results", respective of improving the spatial resolution of RS images "per a factor of four" [207]. Meanwhile, in [208] a very deep encoder-decoder network, namely RED-Net, was then proposed that too "achieves better performance than state-of-the-art methods on image denoising" and SR, with the use of "symmetric skip-connections". Which amalgamate the encoder and decoder sections of the network to help with "recovering clean images", whilst obtaining "performance gains when the network goes deeper"[49]. It is recognised by several other works for models based on skip-connections to have achieved "state-of-the-art performance in many tasks" [35], also. Among them, ResNet [209] is perhaps the most "representative model" [35]. Advancing from this work, in [210] the authors propose a "major architectural" [32] development, namely SRGAN[50], which poses as the first attempt to apply Generative Adversarial Network's (GANs) to the SR problem domain [33]. As such, SRGAN utilises "perceptual loss and adversarial loss" to generate images that appeal more "realistic and natural". Where the former, as a "loss function" [210], is purposed for assessing a solution with "respect to perceptually relevant characteristics", in generating a convincing SRResNet. Whereas the latter, is used within the "discriminator network" to differentiate between the super-resolved and original, photo-realistic images; the former continues to "carry out training" [33] until the discriminator

[48] In becoming the Remote Sensing Very Deep Super-Resolution (RS-VDSR) network.
[49] The deeper, RED-Net-30 network is reportedly the best overall method, performatively.
[50] Which is the acronym for Super-Resolution Generative Adversarial Network.

can be "deceived" by the super-resolved renditions. As the authors report, for large upscaling factors, SRGAN by a "considerable margin" [210], generates more "photo-realistic images" than other state-of-the-art methods. Of such methods is SRResNet, which the authors also propose as an independent network, composed up of "16 residual units" [35][51] or blocks[52] "optimised for MSE" [210][53]; in each ResBlock, SRResNet adopts batch normalisation (BN) to "stabilise the training process" [35]. Relative to the quantitative findings of the study, SRResNet "sets a new state-of-the-art" [210]. However, depite being "very successful" [35], SRResNet faced architectural deficiencies concerning its feature of "redundant modules", which declared it to be "suboptimal" [210] for low-level vision tasks[54]. In resolve of this deficiency, the work of [211] then proposed to optimise the model by "analysing and removing unnecessary modules" from its architecture, and in doing so, the authors established EDSR[55]. Where all BN operations are removed from each residual block [35], to allow information to be affected by fewer "changes" [36], and the networks loss function is altered from the "L2 to L1 norm" [34][56], for achieving "better convergence". Moreover, alongside their single-scale model, the authors also proposed a multi-scale variant, namely MDSR, which is inspired by VDSR [205] to "take advantage of inter-scale correlations" [211]. By implementing "scale-specific processing modules" in the head of the single-scale network, the model became capable of handling SR in "multiple scales". Per the results, the authors report that EDSR "surpasses current models" and achieves state-of-the-art performance, with MDSR competing with a "comparable performance". Later inspired by EDSR, a ResNet variation namely DSen2[57] was proposed [18], that learned to "transfer the high-frequency content" of the HR spectral bands, to the LR spectral bands of Sentinel-2 imagery; contrary to the methods prior, that "hallucinate" HR textures based on previously seen images. Such that the resulting pixels in DSen2's output(s), have "plausible spectra". Due to the methods "global applicability" for Sentinel-2 imagery up-sampling, it is widely respected in literature as the "representative method in the CNN category" [1]; where the methods proposed regulariser, serves as a "big regression engine" [18] from multi-resolution input patches to high-resolution patches, such that it automatically adapts to arbitrary size input data [1]. It is also worthy to mention that the network utilises the full "12/16-bit depth" [32] of S2 images, ensuring spectral information is not lost, as the authors of [212] otherwise discard, in repurposing SRCNN and VDSR for "Pléiades as well as SPOT images" [32]. Providing the "excellent performance" [18] achieved by this model and its very deep counterpart[58], in reducing the Root Mean Square Error (RMSE) of predictions by "50% compared to the best competing methods", the state-of-the-art has since been referred to in deriving many of the most present-day works in the domain [22, 213].

2.2.5 Data Augmentation

DA within the domain of example-based SISR, takes upon a variety of instruction [121] to "effectively improve the performance" of example-based SR models, both architecturally and prophetically. Of these two focuses, the latter has been scarcely investigated [52] despite TTA being present for a "long time in deep-learning research" [122]. As one of the most practical and cost-effective

[51] Otherwise regarded as a "16 blocks deep ResNet" [210].
[52] Referred to as ResBlocks.
[53] Mean Squared Error.
[54] Including SISR.
[55] The Enhanced Deep Super-Resolution network.
[56] From Mean Squared Error (MSE) to Mean Absolute Error (MAE).
[57] The Deep Sentinel-2 network.
[58] Namely, Very Deep Sentinel-2 (VDSen2).

approaches to enhancing “model performance” [52], TTA in more recent works, has “often been used to produce more robust prediction results” [123]. Commonly, TTA can be characterized as the linear process that governs image “augmentation, prediction, dis-augmentation, and merging” [214]; which can be viewed “analogous to ensemble learning techniques in the data space” [115].

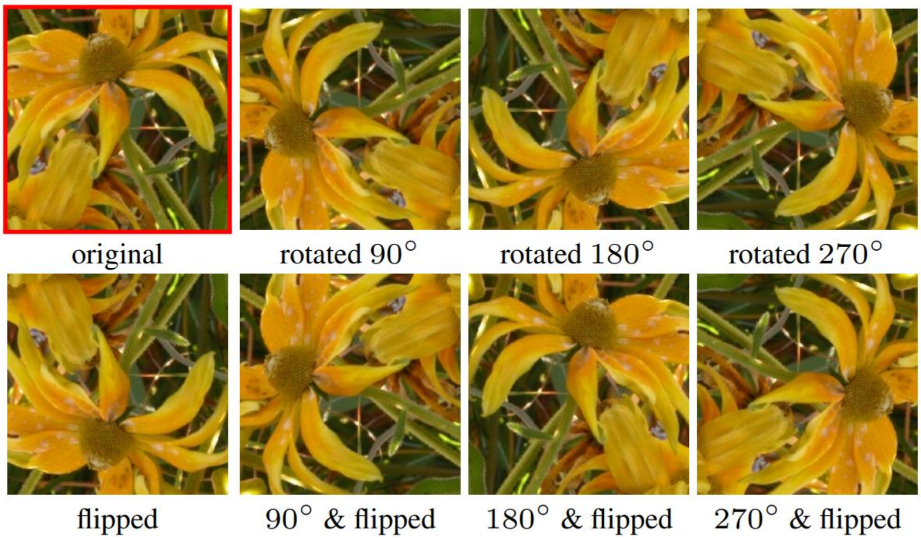


Figure 25: Depiction of seven augmented input images by rotation and flip transformations; the geometric self-ensemble scheme [121].

As the first study into DA techniques for “improving the performance” [52] of example-based SISR, the work of [121] proposed seven “generic” techniques to achieve “substantial improvements”, without incurring changes to the baseline method. Of the seven techniques, the authors approach TTA or “enhanced prediction” as a basic matter of averaging the predictions of a “set of transformed images”, derived from the prediction of an input image. In which the authors address by using “cropping, flipping, and rotating” operations, in achieving “consistent improvements across models and datasets” [52] alike, without significant increases in running time [121]. Prior to this study, in [116] the authors similarly averaged the predictions of AlexNet, an image classification model, over a series of “random crops and flips of test data” [122]. Which then led to becoming a “standard evaluation protocol” in the future works of [117, 209], following its success. Further in this direction, pioneered in [211] the authors derive a technique, namely ‘geometric self-ensemble’, that also standardises the work of [121] as set of “seven augmented inputs” [211], at the time of inference [215] (see *Figure 25*). Which entertains a “performance gain” similar to conventional model ensemble methods, without the requirement of “individually trained models”. Thus, encouraging the techniques presence in numerous other works following [216, 217, 218], wherein [216] the authors alternatively take the “median” of eight outputs as opposed to their “mean”. Deferring from this arrangement of techniques specifically, the authors in [219] report the best performance gains for image classification tasks, in use of “geometric and colour transformations”, combinatorically. Remaining relevant to image classification, the authors of [220] also report the “effectiveness” [115] of using geometric transformations. Beyond this narrative, as a “promising practise” for medical image diagnosis applications, the authors in [221] denote TTA as ‘data distillation’, in which they adopt “scaling and horizontal flipping” transformations to fabricate a “multi-transform inference”, that consistently improves upon the results of state-of-the-art models. In the same year, [61] introduced a novel “random shifting technique” that utilises a translation operation in a "uniformly" random, iterative fashion, to improve the spatial quality of Magnetic Resonance (MR) images. Unlike prior methods, this technique is both iterative and stochastically-driven, which the authors of [221] suggest as being a potential improvement and future work of theirs. Reportedly, the technique obtains "better results", quantitatively, over a state-of-the-art method[59]. Thereon, other similar works have since been proposed [222, 223], none of which offer remarkable advances in the domain, however.

Chapter 3 | Methodology

3.1 Research Approach

In summary of the literature explored for the tackling SR, when taking the "standard metrics" [32] of image quality assessment (IQA) into consideration[60], it is apparent that the "deeper, residual networks deliver the best results". Where identified by the literature as an influential, state-of-the-art method tailored to Sentinel-2 SISR, this study purposes DSen2 [18][61] as both the core baseline and augmented[62] models for the experiments aforementioned[63]. Aside from the model achieving state-of-the-art performance and being publicly available, the globally applicable [18] network also supports input data of arbitrary size[64], which it addresses using "zero-padded convolution", as well as the native colour depth of Sentinel-2 images, that enables the spectral information of input data to be well-persevered. In addition to these features, the model also arrives "pretrained" in lower-evaluation scale settings, 40→20m[65] and 360→60m[66] GDS's respectively [166], aligned with the authors claim regarding "scale-invariance" [18]; where the spatial-spectral mappings between LR-HR image pairs are acknowledged as being "roughly equivalent". Such that the authors train two CNNs with data at the scales listed, to perform 20m→10m and 60m→10m up-sampling, respectively. In knowing of this and by assessing the specifications of the hardware available to the study[67] and of that used in [18], the elected TTA techniques could be trialled at even lower-evaluation scales, respectively, and still be representative of SR for both contemporary and dated[68] MS satellite imaging. That not only addresses modern-day RS interests, but historical ones more so [224]. It is on these grounds that DSen2 is favoured.

Supplementary to the study, the classical bicubic interpolation method [140] is appointed as another reference model, as not only is the method used to synthesise lower-resolution images [18], but it is widely applied as a comparative up-sampling technique in the SISR domain [102, 121, 201]. Which is a convention this study respects also.

Entertained by the review of DA strategies regulated in recent DL applications, it is certain that techniques derived from geometric transformations prove to be the state-of-the-art. In which TTA

[59] Namely SRCNN3D.
[60] They are peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM).
[61] Publicly available from: https://github.com/lanha/DSen2.
[62] Termed DSen2+, in following the field-standard notation of data augmented models [201, 211].
[63] Refer to subsection 1.3.
[64] Reflecting the width and height dimensions of a two-dimensional image.
[65] For 20m to 10m GSD spectral band up-sampling; a scale factor of two.
[66] For 60m to 10m GSD spectral band up-sampling; a scale factor of six.
[67] Refer to subsection 3.4.
[68] Perhaps archived.

specifically, demonstrates consistent enhancements to a model's inference capability, whilst being easy to use and simple to put into practise; this is deemed attractive from a development standpoint. Which is reflected by TTA's recurrence in many works, respective of all other augmentation strategies currently known and trialled for image classification and super-resolution tasks [115]. Among the strategies identified, both the geometric self-ensemble [211] and the stochastic translation scheme [61] methods display state-of-the-art performance gains, relative to the PSNR and SSIM IQA metrics, over various state-of-the-art SR methods. As well, the corresponding authors report that each technique does not require additional training of the affected model(s) and can be addressed using "off-the-shelf libraries" [120]. It is for these reasons that both of the aforesaid methods are adopted.

Given the state-of-the-arts identified, this work proposes that the abovementioned DA techniques are implemented within the predictive phase of the DSen2 model, for acquiring higher, spatially enhanced Sentinel-2 imagery. Optimal configurations for both DA techniques are explored separately, relevant to the number of transformations each technique derives from a given input image, as well as the nature of the transformations themselves. This surpasses the scope of the techniques as they are reported in [18] and [61], in hope of establishing a representative set of DA technique configurations, tailored to RS applications; providing that this nature of study has not been investigated for satellite imaging SR prior [109]. Furthermore, the order in which the DSen2 model crops Sentinel-2 tiles [18] and that the DA techniques are applied, before or after "tilting", is also studied to extend the efforts towards reaching optimal configurations. Like [216], the sampling operation used to obtain the resulting ensemble image is also examined, to further warrant the best possible configurations; this entertains that either a mean or median aggregate is produced from a set of transformed images. As well, a combinative use-case of the techniques is then analysed upon discovering each's optimal state, to identify whether a joint effort can surpass the spatial resolution achieved by the baseline model, and when either technique is applied in isolation.

Of particular focus to this study, only the 20m GSD bands of Sentinel-2 tiles are affected, to address the investigations proposed. Specifically, the RGB composite of spectral bands B5, B6, and B7 is studied, which constitutes to the 'vegetation red edge' frequency wavelength (see *Table 1*) and focuses upon the measure of "vegetation chlorophyll content" [225]; desired for land-cover and climate change monitoring. This is aligned with the limitations posed by the hardware available to the study. Which throughout the development cycle, was noticed to be inadequate for super-resolving the 60m GSD bands, because of the processing expense incurred by DSen2's approach to tile decomposition. Aberrations from the model's original schematic were not anticipated nor of interest to this study's demonstration also. To note, DSen2 defaults to the B5, B6, and B7 band RGB composite for 20m GSD up-sampling, as opposed to the B8A, B11, and B12 band alternative. Nonetheless, the 10m GSD bands do not have any participation in the experiments discerned, providing that they natively support a VHR, and quantitative evaluation on Sentinel-2 images is "only possible at the lower scale at which the models are trained" [18], regardless. Where instead, this study opts for evaluation at lower scales, 80→40m and 40→20m GSD's respectively, based on the model's assumption of scale invariance. In which the results can be reasoned as being representative of the 60m GSD bands also, knowing that lower-resolution images naturally inherit more instances of noise and artifact articulation. Although a limitation of this study, the experimentation that is feasible does not affect the credibility of the discoveries made and the conclusions that are then later drawn.

3.2 Inference Data

Sentinel-2 data is freely available and can be acquired from the Copernicus Open Access Hub [226], in tiles of 110 x 110km^2 (≈0.8GB per tile) [18]. Like [22], this study repurposes the tiles used in [18], to evaluate the performances of the reference and augmented models. Of the fifteen tiles that were originally purposed for testing, ten of those are arbitrarily selected for this study. Which is an amount deemed sufficient for accurately representing the findings of this study, especially as they are sampled "from around the globe".

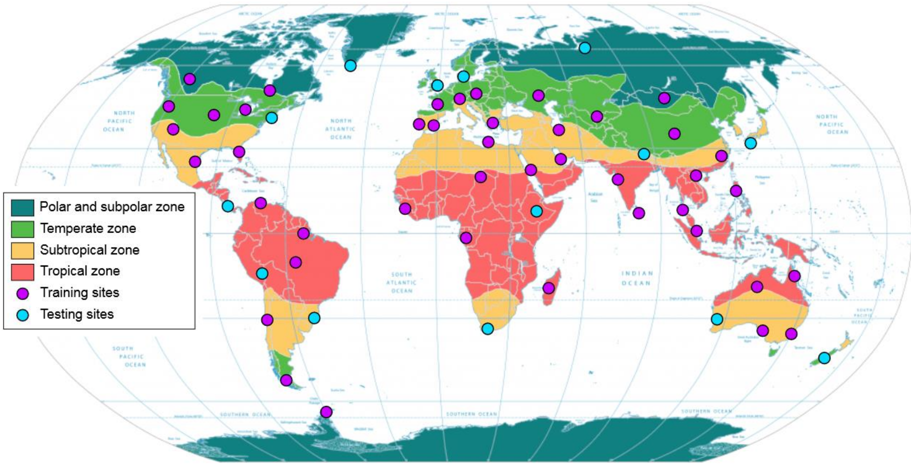


Figure 26: Depiction of a map of the world, illustrating the locations of the Sentinel-2 tiles acquired for training and testing the DSen2 model [18].

Therefore, this study utilises data from both Sentinel-2A and Sentinel-2B satellites, acquired between December 2016 and November 2017, as well as July 2017 and November 2017, respectively. The authors claim to have chosen these tiles "randomly", to obtain an even distribution of "climate zone, land-cover, and biome type" (see *Figure 26*); each with no undefined black backgrounding pixels, to note.

3.3 Development Tools

Supporting the development process of the study, below, identifies the software suites and utilities that were used throughout its undertaking.

3.3.1 Programming Languages

Python 3 – an "interpreted, object-oriented, high-level programming language with dynamic semantics" [227]. Python is simple and easy to learn, that features built-in data structures, dynamic typing, and dynamic binding, making it very appealing for rapid software development. As well, the language supports a wide variety of modules and packages, that "encourages program modularity and code reuse". This is the principal development language of [18].

3.3.2 Integrated Development Environments

Microsoft Visual Studio – a creative suite that one can use to “edit, debug, and build code” [228] with. Visual Studio (VS) includes “compilers, code completion tools, graphical designers, and many more features to ease the software development process”, that as a feature-rich program, can be purposed for the development of many software applications. As a widely used client-side program, VS provides “open-source support for the Python language” [229].

Google Colaboratory – Colaboratory or “Colab for short” [230], is a free-to-use product from Google Research, that allows “anybody to write and execute arbitrary Python code through their browser”. It is well-suited to machine learning, data analysis, and educational interests, that as a server-side suite of tools, provides free-access to computer resources “including GPU’s”. This program is convenient for processing data faster than most local workstations, supporting Python 3, which is accordingly purposed for IQA.

3.3.3 Software Utilities

Imageio – a Python-specific library that provides an easy interface to “read and write a wide range of image data” [231], including data of animated, volumetric, and scientific formats. The utility is “cross-platform”, “easy to install”, and for the nature of this study, is necessary for writing processed image data to an array of external file formats.

OpenCV – an “open-source computer vision and machine learning software library” [232]. OpenCV has a range of programming interfaces, including Python, and provides support for many operating systems also. As one of the libraries focuses, OpenCV supports a suite of image processing tools, including geometric transformations for image data [233], featuring scaling, translation, rotation, affine, and perspective manipulations.

Scikit-image – a free-of-charge and free-of-restriction collection of algorithms, purposed for image processing applications [235]. Aside from the libraries image processing modules, scikit-image also provides separate support via utility methods, to address image data type conversion [236]. Which is principal for preserving the spatial-spectral information of images, pre- and post-processing.

Sewar – a Python-derived package for performing “image quality assessment using different metrics” [237], including MSE, RMSE, PSNR, SSIM, and UQI to name a few. Which is necessary for recording comparative spatial enhancements or degradations, post image super-resolution.

3.3.4 Complementary Platforms

Conda – a “powerful package manager and environment manager” [238], that can be engaged using command-line instruction. Conda allows one to create “separate environments containing files, packages, and their dependencies”, that do not interact with other environments. As a command-line interface, Conda can also be used to execute script files of Python and R language data types, as is entertained by [18].

QGIS – a free and open-source geographic information system (GIS), that enables one to "create, edit, visualise, analyse and publish geospatial information" [239], with a range of supported operating systems and devices. This is purposed as a per-band inspection tool for Sentinel-2 imagery, which supports satellite imaging colour depths.

3.4 Hardware Utilised

Tabularised below, identifies the combination of hardware components and operating system(s) utilised throughout the research and development cycles of the study undertaken.

Table 2: Tabularised feature of the hardware components and operating system(s) utilised throughout the research and development cycles of the study undertaken.

Component	***Description***	***Specification***
Graphics Processing Unit (GPU)	*GeForce GTX 970 GAMING 4G*	*Core Clock Speed* ▪ 1279 MHz / 1140 MHz (OC Mode) ▪ 1253 MHz / 1114 MHz (Gaming Mode) ▪ 1178 MHz / 1051 MHz (Silent Mode) *Memory Clock Speed* ▪ 7010 MHz *Memory Size (VRAM)* ▪ 4096 MB *Memory Type* ▪ GDDR5 *Memory Bus* ▪ 256-bit
Central Processing Unit (CPU)	*Intel Core i7-5820K*	*Core Count* ▪ 6 (hexa-core) *Thread Count* ▪ 12 (logical cores) *Processor Base Frequency* ▪ 3.30 GHz *Processor Maximum Turbo Frequency* ▪ 3.60 GHz *Cache* ▪ 15 MB Intel Smart Cache
Random Access Memory (RAM)	*HyperX FURY DDR4*	*Memory Capacity* ▪ 32 GB (8 GB Single x4) *Memory Speed* ▪ 2666 MHz *Column Address Strobe or Signal (CAS) Latency* ▪ CL16
Operating System (OS)	*Microsoft Windows 10*	*Edition* ▪ Home

Chapter 4 | Implementation

4.1 Methodology

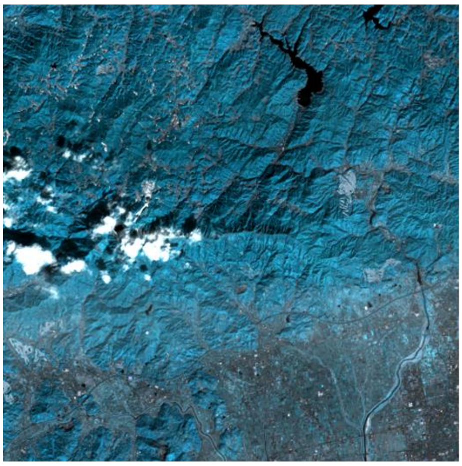
In this section, a description of the study's development cycle is provided, aligned with the model requirements for conducting the abovementioned trials.

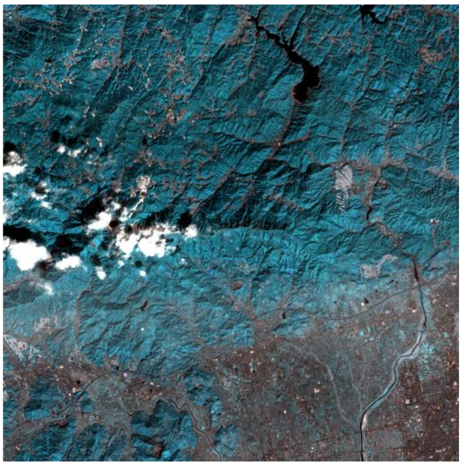
4.1.1 Model Configuration

In following the directives provided by the authors of [18] at [240], one can acquire the DSen2 model, in a readily executable state. However, unspecified by their instruction, a user should consider using Conda [238] that integrates the Python package-management system (PIP) [241], for seamlessly installing all the required utility packages and for then executing the model. Conveniently, this installation procedure can then also be reiterated for the software utilities required for this study's experiments, as they are listed in the section above. To execute the model, one should refer to the command-line instructions provided by [240] as well. However, in the case of a 'cudaNN library' incompatibility error encounter, instead refer to [242] to acquire the version that was originally used to compile the model with[69].

4.1.2 Model Adaptation

Providing the TTA focus of this study, it was only necessary for the 's2_tiles_supres.py' and 'supres.py' [240] Python script files to be modified, aligned with the experiments proposed. As both data structures wholly contain the image pre-processing, super-resolution, and post-processing procedures of the model, respectively. Supplementary to the augmentation of the model, min-max contrast stretching [243] as a default image post-processing technique of QGIS [239], is implemented in both files mentioned to "improve the contrast" of all image data (see *Appendix B*). This poses as a simple method towards the enhancement of an image's perceptual quality, that renders one more colour-rich and naturally appealing, in correspondence to the issues reported in [244]. The enhancement, however, is addressed via remapping the original maximum and minimum pixel values of an image, to a "newly specified set of values that utilize the full range of available brightness values".





[69] The required version is cudaNN 8.1.0, or alternatively a higher minor version variant.

Figure 27: Depiction of a super-resolved, 2,000 x 2,000-pixel crop of a Sentinel-2, 20m GSD tile image, with an RGB colour composite comprised of bands B5, B6, and B7. On the left, illustrates the image in the absence of min-max contrast stretching, whereas the depiction located to the right demonstrates the perceptual quality when min-max stretching is otherwise applied.

Designed for all possible use-cases, min-max stretching is configured as a method, namely 'min_max_contrast_stretch', that arbitrarily handles single-band images and colour composite images, comprised of multiple bands. The operation is contained within a method declaration to adhere to code reuse practises, and can simply be characterised by the following notation [243]:

$$x_{new} = 255 * \frac{x_{input} - x_{minimum}}{x_{maximum} - x_{minimum}} \tag{4.1}$$

Additionally, for the enablement of per-band inspection in QGIS, this study also extends the use of the 'write_band_data' method[70] (see *Appendix C*), for exporting all ground truth, bicubically interpolated, and super-resolved image datasets, as separately generated files of geo-referenced data types[71].

4.1.3 Model Augmentation

Advancing from the customary adaptations imposed on the DSen2 model, below, details the methodologies proposed for augmenting the DSen2 model, at the time of inference.

4.1.3.1 Down-sample Simulation

Aligned with the explanation given in [18], one can generate inference data with a "desired scale ratio s". Of interest to this study, 80→40m and 40→20m up-sampling can be achieved using the original 20m spectral bands of the nominated Sentinel-2 tiles, meaning $s = 4$ and $s = 2$ respectively. Such that ground truth data can either be acquired from the original 20m bands directly (for 40→20m), or when the 20m bands are down-sampled, where $s = 2$ (for 80→40m). To heighten the realism of the down-sampling procedure, as described in [18], this study performs the image degradation process closely aligned with the modulation transfer function (MTF) of Sentinel-2's MSI. Which uses "Gaussian-like low-pass filters" [244] to achieve spatial degradation. As such, the original Sentinel-2 tile data is first blurred via a Gaussian filter, with a standard deviation of $\sigma = 1/s$ pixels, before then being down-sampled by "averaging over s x s windows" (see *Figure 28*).

[70] Located in 's2_tiles_supres.py'.
[71] By default, the DSen2 model generates GeoTIFF data types for this purpose.

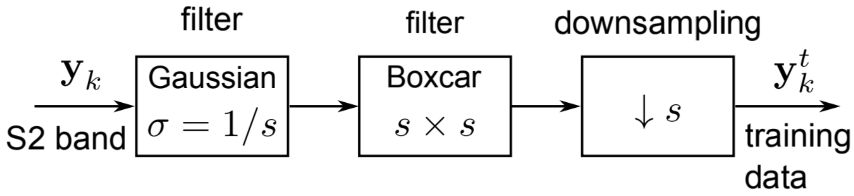


Figure 28: Visualisation of the down-sampling process used for simulating the inference data at lower evaluation scales [18].

From a programmatical standpoint, this work repurposes the 'downPixelAgrr' method[72] to execute the process described above; providing that the authors evaluate the model at a lower scale in their work, where quantitative evaluation is "only possible" [18]. Which requires the relevant image and its desired scale factor, as its functional parameters (see *Appendix D*). As a pre-processing practice, the method is consequently invocated in the 's2_tiles_supres.py' script file, prior to any image data being super-resolved (see *Appendix E*).

4.1.3.2 Geometric Self-ensemble

Corresponding with the geometric transformations featured in the work of [211], this study authors the support for image inversion and rotary operations, to primarily generate seven augmentations of a given Sentinel-2 tile image, at the time of inference (see *Figure 29*). To achieve such, three methods, namely 'rotate_image', 'invert_image', and 'rotate_invert_image' are separately defined[73], to distinguish between the dissimilarities of each transformation policy (see *Appendix F*); syntactically, modular declarations were believed to be better-adapted for maintaining the robustness of the code base and for expediting development. To note, each of the methods listed, requires an image, a transformation constraint, and a Boolean argument as its functional parameters. Firstly, the transformation constraint is resolved by the OpenCV [233] utility, as either a 'flip' or 'rotate' operation, before each of the spectral bands comprising the image can then be transformed, in an iterative manner, for establishing a new inter-band co-registration. Aligned with the tile decomposition feature of the model, the Boolean argument, namely 'tiled', exists to condition the invocation of each methods functionality; dictating whether the transformation constraints are to be applied to a tile image, or alternatively, an image that has been decomposed into a sequence or array of tiles. Thus, to investigate the sequencing hypothesis of this study.

[72] Located in 'patches.py' of the DSen2 repository.
[73] Located in 'supres.py'.

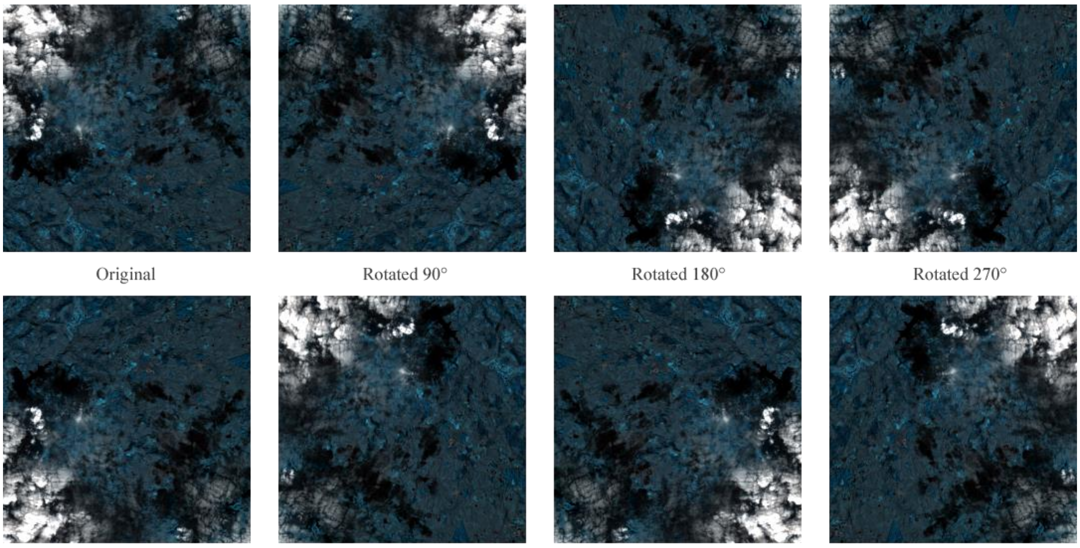


Figure 29: Depiction of augmented 2,000 x 2,000-pixel crops of a Sentinel-2, 20m GSD tile image, with an RGB colour composite comprised of bands B5, B6, and B7. Demonstrating the reproducibility of the geometric self-ensemble scheme, at the time of inference, as is reported in [211].

For simplicity, all transformation constraints are defined within separate array declarations in the 'DSen2_20' method[74] (see *Appendix G*), which has been extended to govern both 20m band super-resolution and TTA procedures. Arrays of transformation constraints were opted for, given the iterative nature of deriving image transforms, that through element-wise indexing, allows each transformation constraint to be accessed and then applied to a new instance of the input image, autonomously. To note, for every array of transformation constraint[75] there exists a corresponding array, purposed to "reversing the transformation on the HR outputs" [121] (see *Appendix G*); this is required for obtaining an ensemble of the predictions, for which the transformed images are restored to their original pose to address. Importantly, this arrangement is not limited to a specified set or number of transformations, as all transformations are appended to an aggregate array, namely 'all_constraints_unapplied', where they are all invocated from. Which allows all possible combinations and quantities of transformations to be explored, for this study's anticipation of achieving optimality.

4.1.3.3 Stochastic Translation

Aligned with the instruction provided in [61], this proposal also adopts the implementation of the 'random shifting model', to generate a series of stochastically translated images at the time of inference. Like the operators constituting geometric self-ensemble, the unary operator of this scheme is also defined as a method, namely 'translate_image', following the regularity of naming conventions (see *Appendix H*). That also requires an image, transformation constraint, and Boolean argument as its functional parameters, for the same conveniences mentioned prior. Wherein, each transformation constraint is received as a set of coordinates, in x and y axis notation respectively, representing the positional or translative offsets in pixel space, that are inflicted on each of the spectral bands comprising an image. Image translation is addressed by the authors [18] existing feature of the numerical Python

[74] Located in 'supres.py'.
[75] Except for inversion operations.

(NumPy) [246] utility package, that can be used to 'roll' or displace an image at a pixel level, to the positions that correspond with the schemes randomly generated offsets. As the data structure representing an image can be interpreted as an array, by 'rolling' the data, an image can be uniformly wrapped around its canvas space, as opposed to being projected beyond it[76]. Such that all image data remains within the original canvas space (see *Figure 30*), as is later required for reversing the transformation(s) proceeding super-resolution. Meanwhile, the Boolean argument 'tiled' is likewise applied to control the methods functional invocation, for investigating the significance of DA sequencing in this study.

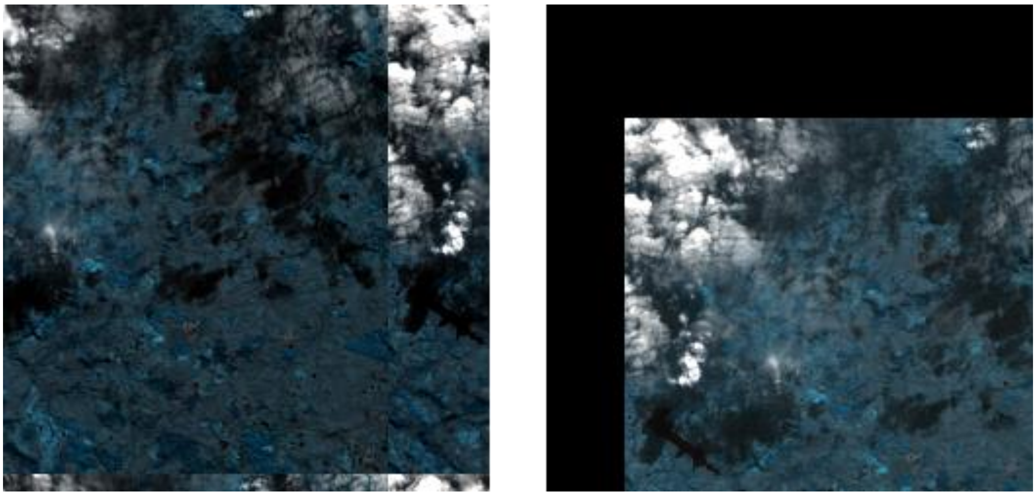


Figure 30: Depiction of augmented 2,000 x 2,000-pixel crops of a Sentinel-2, 20m GSD tile image, with an RGB colour composite comprised of bands B5, B6, and B7. Demonstrating a wrapped mode of pixel translation (left) and a constant mode of pixel translation (right), at the time of inference.

To address the generation of transformation constraints, a separate method is defined, namely 'define_translation_constraints' (see *Appendix I*). Wherein, random integer values representing the positional offsets for both x and y axis modes of translation, are sampled from two separate uniform distributions configured with the range $[-T, T]$, for a desired "number of shifts" [61]. Such that T represents the maximum offset allowed for an image, with respect to the count of pixels along any given axis. Which is derived from the term $T = D * K$, where K represents the "window size" of the operation, and D corresponds to the width or height dimension of the image[77]. Unlike previous method definitions, this method requires an image, a window size, a transformation count, and two array data types in which the compiled constraints populate[78] (see *Appendix J*).

4.1.3.4 Test-time Adaptation

Dissimilar to the implementation of geometric self-ensemble, an independent testing regime correlated with [61] is also featured by this work, in determining an optimal parametric configuration of the stochastic translation scheme. In which this study abandons the configuration it presents for MR

[76] This create black-pixel margins, from where an image has been displaced from.
[77] Refer to line 212 in *Figure 42*.
[78] These variables are passed by reference rather than by value.

imaging, to identify one that is optimal and representative of RS applications, that utilise MS imagery. In doing so, a testbed is also configured within the 'DSen2_20' method, for identifying the optimal window size and the number of shifts or translated images, that should constitute to the ensemble output (see *Appendix K*). To utilise the testbed, one should ensure that the 'translation_testbed' argument of the containing method, is set to 'true'. Following the algorithm as it is proposed in [61], the testbed initially populates a "set" or array of translation vectors, using the 'define_translation_constraints' method, as it is characterised in the section above. Secondly, in an iterative fashion, for every transformation constraint that has been populated in the respective array, a new instance of the input image is then derived from computing the transform that is currently cycled. Which is addressed via the 'translate_image' method by passing a copy of the input image[79] and the corresponding constraint; optionally, the transformation can be applied to the image, before or after it is decomposed into a series of tiles, by regulating the state of the 'self_ensemble_before_tiling' argument[80]. Thereon, the image is then passed to the models CNN for up-sampling, alongside the 10m band data, in producing a HR render of the transformation. Upon which, the images pose is then negated by recycling the 'translate_image' method, that instead passes the inverse constraint for the transformation that was previously applied. The entirety of this procedure results in the obtainment of a single HR restoration, that governed by an iterative statement, proceeds to produce a series of similar images, so that an ensemble of the series can be acquired, representing the "final restored HR image".

Given that the testbed is an encapsulation of all operations described above, per iteration of the series, the model encounters an incremental increase in the number of images to super-resolve, that subsequently comprise the ensemble output. For every ensemble output that is compiled, its contrast is enhanced via min-max stretching, using the 'min_max_contrast_stretch' method. Before its perceptual quality is then compared with the respective ground truth data, using the suite of Sewar [237] IQA methods available. The results obtained from the IQA metrics that have been purposed to this study, are then output to external files of ASCII data types[81], for later review; in determining the optimal configuration of the schematic. In this study, the proposed testbed terminates upon an ensemble of fifty images being produced in a single cycle, instead of the one-hundred that the authors climax to in their work [61]. This is aligned with the developing decline in model performance, that they report to occur after an ensemble of thirteen images has been created, which is inferred from the results of three input images, nominated from three distinct datasets.

Advancing from the testbed's incorporation into the model, the integration of geometric self-ensemble and the stochastic translation scheme adopts a very similar approach (see *Appendix L*). That instead, utilises all transformation methods and constraint types available to the study, to acquire ensemble outputs for either augmentation strategy. Such that the aggregate array, namely 'all_constraints_unapplied', can be populated with all types of transformation constraint, to address combinative and single use-case scenarios of the techniques, as this study aims to investigate the performance implications of. Distinguishing from the testbed's implementation, however, is the absence of the IQA metric calculations and their output, which is deferred to an external process, for simplicity. As well, proceeding from the acquisition of an ensemble output, the baseline model is then executed to obtain the reference image of the study, which is then also subjected to contrast enhancement, for upholding perceptual consistency in the results.

[79] By value and not by reference.
[80] Which is a parameter of the 'DSen2_20' method definition.
[81] This study purposes files of (.txt) formats.

4.1.3.5 *Ensemble Image Sampling*

To acquire an ensemble image, an array, namely ‘super_resolved_transformations’, is defined for containing all the restored HR images, to which they are appended, after each super-resolved images pose is restored (see *Appendix L*). Upon all images being gathered, the array is then exposed to either a mean or median sampling operation, to derive the aggregate result. Which is simply addressed using the corresponding methods provided by the NumPy statistics [247] utility. To expedite the experimental process, the associated NumPy methods can be passed by value to the ‘DSen2_20’ method, alike the ‘self_ensemble_before_tiling’ and ‘translation_testbed’ Boolean arguments. As mentioned previously, support is provided for multiple sampling operations to investigate and warrant the best possible configuration for the model.

4.1.4 Image Quality Assessment

For comparative purposes, as an “elementary baseline” [18] of the study, bicubic interpolation is featured to demonstrate a naïve up-sampling of Sentinel-2 data, that does not preserve the spectral correlations therein. Given the methods simplicity and long-standing support, the OpenCV [233] utility package is used to bicubically up-sample the image data, corresponding to the dimensions of the ensemble output[82].

To obtain a comprehensive quantitative comparison of the predictions acquired by the baseline and augmented models, several popular IQA metrics in the literature [17, 18] were considered for validating this study’s outcomes. They are:

Peak Signal-to-Noise Ratio (PSNR) – a “standard metric” [17] used to evaluate the quality of a reconstructed image, on a “pixel-wise” [102] basis. Derived from Mean Square Error (MSE) [248], PSNR indicates the “ratio of the maximum pixel intensity to the power of the distortion”; higher PSNR generally infers a higher quality image.

$$PSNR = 10\log_{10}\left(\frac{255^2}{MSE}\right) \quad [83] \tag{4.2}$$

Structural Similarity Index (SSIM) – a metric that measures the similarity between two images, considering their “luminance, contrast, and structure” [17]; higher SSIM indicates a more convincing reconstruction, such that a SSIM equal to one, corresponds to “identical images”.

[82] Unlike the ensemble output, bicubic interpolation is performed within the ‘s2_tiles_supres.py’ Python script file.

[83] Where 255 represents the maximum grey-level of an 8-bit, monotonic image.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \text{ [84]} \tag{4.3}$$

Root Mean Square Error (RMSE) – a metric that measures the root "average squared difference between actual and ideal pixel values" [248] of two images; higher RMSE infers lower image qualities, such that the amount of change per pixel is increasingly obvious.

$$RMSE = \sqrt{\frac{1}{n}\sum(\hat{x} - x)^2} \text{ [85]} \tag{4.4}$$

Spectral Angle Mapper (SAM) – a metric that "measures how faithful the relative spectral distribution of a pixel is reconstructed, while ignoring absolute brightness". Calculated as the angular deviation between two images; higher SAM indicates an increasing similarity between two given images.

$$SAM = cos^{-1}\left[\frac{\sum_{i=1}^{n}(T_i \cdot R_i)}{\left(\sum_{i=1}^{n} T_i^2\right)^{\frac{1}{2}} \cdot \left(\sum_{i=1}^{n} R_i^2\right)^{\frac{1}{2}}}\right] \text{ [86]} \tag{4.5}$$

Universal Image Quality index (UIQ) – a "unitless" [18] yet universal measure of the "difference in pixels" [249], between two images, relative to each's "luminance, contrast, and structure" [18]. Higher UIQ supposes superior image quality, such that a UIQ equal to one, corresponds to identical images.

$$Q = \frac{\sigma_{xy}}{\sigma_x\sigma_y} \cdot \frac{2\bar{x}\bar{y}}{(\bar{x})^2 + (\bar{y})^2} \cdot \frac{2\sigma_x\sigma_y}{\sigma_x^2 + \sigma_y^2} \text{ [87]} \tag{4.6}$$

[84] Where: μ_x and μ_y represent the mean values of images x and y, σ_x and σ_y represents the standard deviation of the images, and σ_{xy} represents the covariance of the images [61]. Whereas constants $c_1 = (k_1L)^2$ and $c_2 = (k_2L)^2$ represents the values that "depends on the dynamic range (L) of the pixel values" [17]. Default values were used: $L = 1$, $k_1 = 0.01$ and $k_2 = 0.03$.

[85] Where $\hat{x}$ represents each reconstructed spectral band (vectorised), x represents the vectorised ground truth band, and n represents the number of pixels in x [18].

[86] Where n represents the number of spectral bands comprising images T and R, T_i represents the reflectance value of band i in the test spectra, and R_i represents the reflectance value of band i in the reference spectra.

[87] Where the first component represents the correlation coefficient between images x and y, the second component represents the measure of how close the mean luminance between the images is, and the third component represents the measure of how similar the contrasts of the images are [250].

All metrics identified above are implemented in a Python (Jupyter) notebook file[88], dedicated to IQA, that exists outside of the DSen2 model and its repository (see *Appendix M*). This enables a server-side machine provided by Google Colab [230], to be utilised for conducting IQA routines, whilst the local-machine available to this study can then be committed to generating the results for the experiments proposed. Thus, expediting the study's research cycle. Similar to the testbed proposed, the suite of IQA methods provided by the Sewar [237] utility, are re-adopted, to infer the reconstruction performance of all models bound by this study[89].

Chapter 5 | Evaluation

In this section, the experiments proposed and embarked on for the interests of this study are identified, and a numerical analysis for each experiments outcomes is provided, in resolve of the hypotheses presented for the works undertaking.

As mentioned before, this work aims to identify optimal configurations for both single and combinative use-cases of geometric self-ensemble and the stochastic translation scheme, for the emphasis of RS applications that operate with MS imagery. In doing so, this work proposes to explore each strategies configurative state, beyond the scope of what their authors originally articulate. With the ultimate purpose in mind, which is to inexpensively, surpass the spatial resolution achieved by a state-of-the-art SR model, in supporting more detailed and accurate information extraction, of lower-resolution satellite imaging. To reiterate, only the 20m spectral bands of Sentinel-2 tiles are subjected to augmentation in this work, due to the hardware capacity available; specifically, the RGB composite of bands B5, B6, and B7, are studied, as they are defaulted to by the DSen2 model.

5.1 Experimental Results

With the support of the features introduced to the model, as they are presented in the implementation section of this paper, this study puts forth the following research.

5.1.1 Stochastic Translation

Preliminarily, to ascertain the performance implications of the stochastic translation scheme, the testbed proposed for the technique is firstly engaged with, to establish the schemes optimal parametric configuration. Wherein, for the fifty translations that the testbed endures in every experiment, as mentioned prior, this work trials a window size of: 25, 50, 75, and 100, which extends from the 25 trailed in the original work [61]. As well, for each window size listed, the translations are also trialled before and after tiling the input image. For this series of experiments, only the results for the average ensemble operation are recorded, as the median ensemble operation was consistently observed to incur a computational expense, that would eventualise the termination of the programs runtime session. Which is correlated with the hardware constraints posed on this study. However, for

[88] Namely, 'Super-Resolution: Image Quality Measure.ipynb'.
[89] To note, ensemble images output by model are required to be uploaded to Google Drive, for use with Google Colaboratory.

all the experiments successfully documented, the model processes full-size Sentinel-2 tiles (10,980 x 10,980 pixels) evaluated at the lowest scale that this work examines (80→40m GSD), for an arbitrarily chosen series of images. Like [61], each experiment is repeated for three unique images (see *Figure 31*), in the obtainment of credible results.

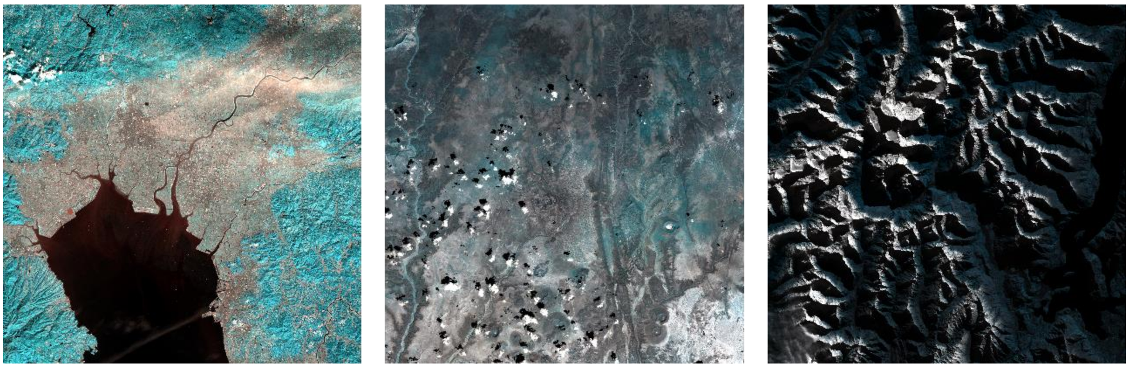


Figure 31: Depiction of the three Sentinel-2 tile images, of size: 10,980 x 10,980 pixels, elected for the parametric configuration of the stochastic translation scheme.

In correspondence with the quantitative evaluation obtained, as is displayed in the table beneath this passage, one can infer that applying the translations before the input image is tiled, yields consistently better results, in comparison to after tiling the image. As is supported by each of the featured IQA metrics. Moreover, it is evident that using smaller window sizes to govern the translative potential of each image, yields better restorations than when opting for greater ones. Such that a window size of 50, is observed optimal. As well, it is also noticed that fewer translated images constituting to the ensemble output, attains consistently better results, in conjunction with the after tiling sequencing pattern. Given these discoveries, the optimal translation count is revealed to be an ensemble of 37 images, derived from the input data. Such that the optimal parametric configuration of the stochastic translation scheme, within the boundaries of the experiments conducted, is determined as the following: window size = 50 and translation count = 37. Refer to *Appendix N*, for the evolution of the IQA metrics, representing the optimum experiment.

Table 3: Quantitative results for 80→40m GSD, representing the performance of the stochastic translation scheme, when subjected to a series of parametric configurations. Relative to an alternating window size and appliance of the translation operation; before and after tiling the input image. The results are averaged across the three images appointed to the experiment; the best are highlighted in bold.

Window Size	*Image Tiling*	*Translation Count*	*PSNR*	*RMSE*	*SSIM*	*UIQ*
25	Before	45	36.112	4.269082	0.986938	0.993857
	After	35	36.18182	4.236816	0.987091	0.993931
50	Before	49	36.11188	4.268699	0.986938	0.993857
	After	**37**	**36.18825**	**4.232218**	**0.987102**	**0.99395**
75	Before	49	36.1124	4.268799	0.986939	0.993861
	After	44	36.17567	4.239603	0.987074	0.993924
100	Before	49	36.11156	4.269173	0.986939	0.99386
	After	45	36.17996	4.237314	0.98709	0.993926

Thereon, in acquirement of the optimal parametric configuration for the stochastic translation scheme, the performance of the schematic is then stressed over the series of ten images purposed to the study (see *Appendix O*), for investigating whether the performance of the baseline model, can be surpassed. Like the previous engagement, however, only the scores for the average ensemble operation are recorded, as once more, the median operation was proven to be conflicting with the hardware available. For this experiment, the model once again processes full-size Sentinel-2 tiles (10,980 x 10,980 pixels) evaluated at the lowest scale (80→40m GSD). In addition to this experiment, the size and scale invariance of this strategy is then also examined, such that each of the ten images purposed, are then processed by the model at half of their original size (5,490 x 5,490 pixels) and evaluated at both 80→40m and 40→20m scales, respectively. To perform 40→20m up-sampling with the hardware constraints imposed, this work appropriates half-size crops of the Sentinel-2 tiles, in response to the hardware failures experienced, upon attempting to process their full-size counterparts.

Table 4: Quantitative results for 80→40m GSD, representing the performance of the optimal configuration for the stochastic translation scheme, when used in isolation. The results are averaged across the ten images appointed to this study, at their full-size; the best are highlighted in bold.

Method	***PSNR***	***RMSE***	***SSIM***	***UIQ***	***SAM***
Bicubic	29.6177	11.6779	0.9188	0.9828	0.0837
DSen2	36.4247	4.2073	0.9867	0.9895	0.0376
DSen2+	**36.6322**	**4.1068**	**0.9874**	**0.9898**	**0.0367**

Aligned with the first of the three experiments mentioned, the results contained by the table above strongly indicate the superiority of using the augmentation strategy, in comparison to the baseline methods. Where for all IQA metrics discerned, the augmented model yields better performance, confirming that an improved reconstruction accuracy can be achieved; despite being an insignificant enhancement. In which the ensemble predictions generated by the augmented model, are determined to resemble the ground truth data more closely, from a quantitative perspective. Thus, in lower-scaled evaluation settings (80→40m GSD), the stochastic translation scheme, in isolation, can be used to yield spatial improvements for MS satellite imaging.

Table 5: Quantitative results for 80→40m GSD, representing the performance of the optimal configuration for the stochastic translation scheme, when used in isolation. The results are averaged across the ten images appointed to this study, at half of their original size; the best are highlighted in bold.

Method	***PSNR***	***RMSE***	***SSIM***	***UIQ***	***SAM***
Bicubic	28.3373	10.2637	0.9181	0.9784	0.1822
DSen2	35.1793	4.6529	0.9865	0.9849	0.0393
DSen2+	**35.3839**	**4.5435**	**0.9869**	**0.9851**	**0.0385**

Corresponding to the second of the three experiments identified, as shown in the table above, a similar correlation in the results is observed. Whereby, the augmented model again, proves to outperform both reference methods, across all featured IQA measures. Respectively, from cross-examining each of the two scenarios results, one can infer that the strategy is in fact size invariant for the same up-sampling factor, given the similar margins of improvement achieved over the baseline model. As such, the following scenarios results can in retrospect, be representative of the performance of this strategy, if the full-size tile images could also be augmented, at the higher evaluation scale studied (40→20m).

Table 6: Quantitative results for 40→20m GSD, representing the performance of the optimal configuration for the stochastic translation scheme, when used in isolation. The results are averaged across the ten images appointed to this study, at half of their original size; the best are highlighted in bold.

Method	***PSNR***	***RMSE***	***SSIM***	***UIQ***	***SAM***
Bicubic	29.3859	9.1134	0.9214	0.9738	0.1702
DSen2	**40.19**	**2.6513**	**0.9907**	**0.9817**	**0.0232**
DSen2+	40.1841	2.6534	**0.9907**	0.9816	**0.0232**

Unlike the results obtained for the two experiments prior, the last of the three experiments conducted, defers from the same narrative. Where in accordance with the results tabularised above, the baseline mode outperforms the proposed variant, in the 40→20m evaluation setting investigated; as is understood from the PSNR, RMSE, and UIQ metrics. Reflected by the IQA measures listed, the proposals inferior performance is assumed to be the cause of an increased articulation of noise and artifact features, in the ensemble reconstructions. Which the authors of [41] explain as being the consequence of averaging "several images", that can in "some cases", blur texture detailing. They too, report "slightly lower metrics" in their work, in compliance with a similar TTA strategy. Thus, one can conclude that the stochastic translation scheme does not yield spatial improvements, when exposed to higher-scaled evaluation settings. That due to the ensemble of images, denies this strategy from being scale invariant.

5.1.2 Geometric Self-ensemble

Advancing from the experimentation led for the stochastic translation scheme, the optimal configuration for the geometric self-ensemble strategy is then studied. Like the experiments conducted for the testbed prior, a series of trials are put forth initially, for examining the implications of transforming images, before and after they are decomposed into smaller tiles. For this collection of experiments, the same three image set is repurposed from a previously led investigation (see *Figure 31*), for acquiring credible results. Also, the model likewise processes full-size tiles (10,980 x 10,980 pixels) evaluated at the lowest scale (80→40m GSD), as was proposed for the preliminary investigations of the stochastic translation scheme. In which, the experiments also neglect considerations for the median ensemble operation, where instead, the results are only shown for the averaged outcome; the optimal ensemble operation is then later studied, in the experiments that follow. For the purpose of these experiments, the geometric self-ensemble strategy adopts the configuration that is articulated with in [211], as its preliminary configurative state, in providing consistent quantitative comparisons (see *Figure 29*).

Table 7: Quantitative results for 80→40m GSD, representing the performance of geometric self-ensemble, when used in isolation and when the input data is augmented before tiling. The results are averaged across the three images appointed to this experiment, at their full-size; the best are highlighted in bold.

Method	***PSNR***	***RMSE***	***SSIM***	***UIQ***	***SAM***

Bicubic	29.13167	9.439333	0.914667	0.986	0.099667
DSen2	35.95233	4.350333	0.986667	0.993667	0.045
DSen2+	**36.28667**	**4.181667**	**0.987333**	**0.994**	**0.043333**

With reference to the results displayed in the tables above and below, firstly, it is evidenced that the geometric self-ensemble strategy provides positive implications on the model's performance, in the 80→40m evaluation setting considered; regardless of the sequence in which the tiling and transformation operations are staged in. Such that for all IQA metrics, the augmented model surpasses both reference methods by a margin greater than what the stochastic translation scheme was able to achieve, for the same series of images (see *Table 3*) and in the same evaluation setting. As well, one can acknowledge that the equivalence of each experiments results, indicates that the order in which the operations are staged in, bares no impact on the peak spatial resolution that the augmented model can attain. Where unlike the stochastic translation scheme, geometric self-ensemble is not determined to have an optimal sequence of operations.

Table 8: Quantitative results for 80→40m GSD, representing the performance of geometric self-ensemble, when used in isolation and when the input data is augmented after tiling. The results are averaged across the three images appointed to this experiment, at their full-size; the best are highlighted in bold.

Method	***PSNR***	***RMSE***	***SSIM***	***UIQ***	***SAM***
Bicubic	29.13167	9.439333	0.914667	0.986	0.099667
DSen2	35.95233	4.350333	0.986667	0.993667	0.045
DSen2+	**36.28667**	**4.181667**	**0.987333**	**0.994**	**0.043333**

Thereon, the performance of geometric self-ensemble is then examined for a broad series of compositional alterations, dictating the amount and types of transformation constraints, that the strategy is deployed with. In doing so, this work explores configurative states beyond of what is formulated in the current literature [211]. To accommodate this investigation, a series of experiments are proposed to address the following sequence of transformation composites: original + seven augmentations (as seen in [211]), original + three rotations, original + horizontal and vertical flips, and the original + a blend of the three rotations and the two flips (in order). For the transformation types listed, this work refers to use of the constraints featured in [211] (see *Figure 29*), that cater for all combinations of geometric transformation, whilst enabling all pixel data to remain within the boundaries of an image's canvas space. To attain reliable results, this series of experiments repurposes the same ten image collection as they shown in *Figure 31*, which are processed by the model as full-size tiles (10,980 x 10,980 pixels), evaluated at the lowest scale (80→40m GSD). Then, for the configuration presented in [211] only, the operation used to derive the ensemble output of the model is studied, where the optimum of the two in the experiment, is then applied to the following configurations tested. As well, the scale invariance of this configuration is then also examined, such that each of the ten images purposed, are then processed by the model at half of their original size (5,490 x 5,490 pixels) and evaluated in the 40→20m setting, respectively.

Table 9: Quantitative results for 80→40m GSD, representing the performance of geometric self-ensemble, in isolation, when the 'original + seven augmentation' configuration is applied, and the ensemble output is obtained via the average. The results are averaged across the ten images appointed to this study, at their full-size; the best are highlighted in bold.

Method	*PSNR*	*RMSE*	*SSIM*	*UIQ*	*SAM*
Bicubic	29.6177	9.292	0.9188	0.9828	0.0837
DSen2	36.4247	4.2073	0.9867	0.9895	0.0376
DSen2+	**36.7528**	**4.0455**	**0.9876**	**0.9898**	**0.0361**

Corresponding to the results displayed in the table above, one can infer from all IQA metrics that the augmented model outperforms the baseline and reference models, drastically, in the 80→40m setting. As was similarly recorded in the experiments prior, where the strategy adopts the same 'original + seven augmentation' configuration. Undoubtedly, geometric self-ensemble proves to yield superior performance gains, compared to the stochastic translation scheme. As this experiment and the two that precede it, authenticates.

Table 10: Quantitative results for 80→40m GSD, representing the performance of geometric self-ensemble, in isolation, when the 'original + seven augmentation' configuration is applied, and the ensemble output is obtained via the median. The results are averaged across the ten images appointed to this study, at their full-size; the best are highlighted in bold.

Method	*PSNR*	*RMSE*	*SSIM*	*UIQ*	*SAM*
Bicubic	29.6177	9.292	0.9188	0.9828	0.0837
DSen2	36.4247	4.2073	0.9867	0.9895	0.0376
DSen2+	**36.6995**	**4.0725**	**0.9875**	**0.9898**	**0.0365**

Thereon, for the same composite of transformations deployed by the strategy, the results, as they are populated in *Table 10*, demonstrate the performance gains achieved when ensemble images are derived from averaging operations, compared to their median-derived counterparts. In which all IQA metrics, except UIQ, support this observation in lower-scaled evaluation settings (80→40m GSD). As a result of this discovery, each configuration deployed by the strategy in the following experiments, produces an averaged aggregate of its image transforms.

Table 11: Quantitative results for 40→20m GSD, representing the performance of geometric self-ensemble, in isolation, when the 'original + seven augmentation' configuration is applied, and the ensemble output is obtained via the average. The results are averaged across the ten images appointed to this study, at half of their original size; the best are highlighted in bold.

Method	*PSNR*	*RMSE*	*SSIM*	*UIQ*	*SAM*
Bicubic	29.3859	9.1134	0.9214	0.9738	0.0784
DSen2	40.19	2.6513	**0.9907**	0.9817	**0.0232**
DSen2+	**40.2002**	**2.6491**	**0.9907**	**0.9815**	**0.0232**

Following from the configuration settled in the experiment prior, the results obtained in the highest evaluation setting (40→20m), inform that the augmented model is still, able to surpass the performance of both reference methods. Where aligned with the IQA metrics featured above, one can appreciate that the strategy enriches the spatial-spectral information of the ensemble images, relative to the RMSE, PSNR, and UIQ assessments; unlike the stochastic translation scheme, which instead, negated the outcomes of this same experiment previously. Providing that the performance gains of the strategy are trivial in comparison to the results acquired in the 80→40m setting, this experiment renders geometric self-ensemble scale variant, alike the stochastic translation scheme.

Table 12: Quantitative results for 80→40m GSD, representing the performance of geometric self-ensemble, in isolation, when the 'original + three rotations' configuration is applied, and the ensemble output is obtained via the average. The results are averaged across the ten images appointed to this study, at their full-size; the best are highlighted in bold.

Method	***PSNR***	***RMSE***	***SSIM***	***UIQ***	***SAM***
Bicubic	29.6177	9.292	0.9188	0.9828	0.0837
DSen2	36.4247	4.2073	0.9867	0.9895	0.0376
DSen2+	**36.6628**	**4.089**	**0.9875**	**0.9898**	**0.0366**

Aligned with the results obtained for the 'original + three rotations' composite, one can easily identify the performative degradation of the strategy, compared to its predecessor configuration. Despite this defect however, in the lowest evaluation setting, the proposed model remains to demonstrate superior performance gains over the baseline method. As is validated by all IQA metrics.

Table 13: Quantitative results for 80→40m GSD, representing the performance of geometric self-ensemble, in isolation, when the 'original + horizontal and vertical flips' configuration is applied, and the ensemble output is obtained via the average. The results are averaged across the ten images appointed to this study, at their full-size; the best are highlighted in bold.

Method	***PSNR***	***RMSE***	***SSIM***	***UIQ***	***SAM***
Bicubic	29.6177	9.292	0.9188	0.9828	0.0837
DSen2	36.4247	4.2073	0.9867	0.9895	0.0376
DSen2+	**36.6587**	**4.0909**	**0.9874**	**0.9898**	**0.0366**

Similarly, for the 'original + horizontal and vertical flips' composite proposed, the results indicate that the strategy experiences further degradations to its performance, upon being deployed in the 40→20m evaluation setting. Nevertheless, the augmented model proves to surpass the performance of the reference methods, yet again. As is showcased by the IQA metrics, presented in the table above.

Table 14: Quantitative results for 80→40m GSD, representing the performance of geometric self-ensemble, in isolation, when the 'original + a blend of the three rotations and the two flips' configuration is applied, and the ensemble output is obtained via the average. The results are averaged across the ten images appointed to this study, at their full-size; the best are highlighted in bold.

Method	***PSNR***	***RMSE***	***SSIM***	***UIQ***	***SAM***
Bicubic	29.6177	9.292	0.9188	0.9828	0.0837
DSen2	36.4247	4.2073	0.9867	0.9895	0.0376
DSen2+	**36.7081**	**4.0684**	**0.9875**	**0.9898**	**0.0364**

Unlike the previous two transformation composites deployed in this investigation, the 'original + a blend of the three rotations and the two flips' complex, yields performance enhancements over both reference models, that are more closely aligned with the original configuration trialled. In which, the strategy once again proves to dominate all IQA's, in the lowest evaluation setting, as can be inferred from the results recorded in the table above. As the last configurative state investigated for the series proposed, this work identifies the original schematic, proposed in [211], as the optimum composite for geometric self-ensemble. Which is provably superior to the stochastic translation scheme, in isolation, for spatially enhancing MS satellite imaging, beyond the capability of the baseline model.

5.1.3 Policy Amalgamation

Lastly, progressing on from the isolated studies of geometric self-ensemble and the stochastic translation scheme, this work then proposes to identify the performance implications of a consolidated transformation policy, from discovering each strategies optimal configuration. For this investigation, two experiments are put forth to examine the scale invariance of the strategy proposed, in both 80→40m and 40→20m evaluation settings, respectively. As previously recycled in other experiments, the ten-image series used to derive the optimum configuration of each strategy, is retargeted for this investigations purposes as well. Given that quantitative comparisons between the strategies proposed for the study, are necessary, in determination of the superior configuration, that this work submits.

Table 15: Quantitative results for 80→40m GSD, representing the performance of the consolidated transformation policy. The results are averaged across the ten images appointed to this this, at their full-size; the best are highlighted in bold.

Method	***PSNR***	***RMSE***	***SSIM***	***UIQ***	***SAM***
Bicubic	29.6177	9.292	0.9188	0.9828	0.0837
DSen2	36.4247	4.2073	0.9867	0.9895	0.0376
DSen2+	**36.7137**	**4.0674**	**0.9875**	**0.9898**	**0.0365**

With reference to the results obtained, as they are displayed in the tables above and below, it is proven that the consolidated strategy provides positive implications on the model's performance, in both evaluation settings studied. Such that the strategy improves upon both reference methods, in correspondence with the IQA metrics compiled. However, in the 80→40m setting, the performance of the policy advances significantly, from its performance in the higher, 40→20m setting. As within *Table 16*, the baseline model shares the same reconstruction accuracy as the proposed method, relative to an image's structural similarity. Whereas within *Table 15*, the proposed method instead yields a noticeably higher reconstruction accuracy by comparison. This correlation is assumed to be the result of detail smoothing in the higher setting, which as previously mentioned, can occur from deriving the ensemble output for several images. As such, the consolidated policy is declared scale variant. Moreover, in comparison to the results populated in *Tables 4* and *9* for the 80→40m setting, one can conclude that geometric self-ensemble, when used in isolation, is the performatively optimal strategy. Meanwhile, in the 40→20m setting, the results shown in *Tables 6* and *11* identify that the consolidated policy, is in fact, the optimal strategy. Where in either evaluation setting, the stochastic translation scheme is performatively inferior, when used in isolation.

Table 16: Quantitative results for 80→40m GSD, representing the performance of the consolidated transformation policy. The results are averaged across the ten images appointed to this this, at their full-size; the best are highlighted in bold.

Method	***PSNR***	***RMSE***	***SSIM***	***UIQ***	***SAM***
Bicubic	29.3859	9.1134	0.9214	0.9738	0.0784
DSen2	40.19	2.6513	**0.9907**	0.9817	**0.0232**
DSen2+	**40.2008**	**2.6489**	**0.9907**	**0.9815**	**0.0232**

5.2 Discussion

Advancing from the quantitative analysis submitted for the results obtained throughout the study, the discoveries of this work are then evaluated against the hypotheses originally put forth for its commission. To reiterate, this work proposes to answer the following matters:

- Can state-of-the-art data augmentation techniques be applied to state-of-the-art deep learning super-resolution models, to further advance the spatial resolution exhibited by Sentinel-2 satellite imagery?
- To what extent, if any, does a combinative use-case of state-of-the-art data augmentation techniques have on enhancing the spatial resolution of Sentinel-2 satellite imagery?
- If a state-of-the-art deep learning model incorporates image granulation into its approach to super-resolution, does the sequence in which data augmentation techniques are applied and that the images are decomposed, impact the peak spatial resolution attainable by the super-resolution model?

In summary of the quantitative observations obtained, this work strongly evidences that the reconstruction accuracy of DL SR models can be enhanced, using DA techniques at the time of inference. Such that the two state-of-the-arts identified in the literature review, demonstrate consistent performance gains over the baseline and reference methods purposed. Validated by the series of IQA measures featured in this study, the optimal configurations established for the techniques prove to consistently yield spatial enhancements for MS satellite image restorations; especially in the lowest evaluation setting studied. In which, geometric self-ensemble is identified as the candidate technique, relative to the greatest performance gains observed, closely followed by the combined strategy that this study engineers, and then the stochastic translation scheme thereafter. As such, the combined use-case of the two strategies studied, demonstrates a sub-optimal performance in comparison to geometric self-ensemble, when it used in isolation. However, the performance gains reported by the amalgamation of the two techniques, are well-respected, when considering the fidelity of the image series utilised. Thereon, when articulating the optimal configuration for either strategy, it was then observed that the order in which the image decomposition and the geometric transformation operations are staged in, posed no effect on the peak spatial resolution that the augmented model could attain, when geometric self-ensemble was invocated. Whereas for the stochastic translation scheme, this narrative was not upheld, as translating the input image data before it was tiled, was proven to yield better reconstruction accuracies in comparison to the alternate sequence of operations.

Given the performance implications of the strategies identified, this work submits three possible solutions for inexpensively enhancing the reconstruction accuracy of example-based DL models, tailored to the SR of MS satellite imaging. As such, this work puts forth geometric self-ensemble as the superior augmentation strategy, as it is articulated in [211], for use with other RS applications that operate with MS imagery. However, hindering the confidence of this suggestion, is this limited scope of experimentation conducted, for the means of hardware available to the study. In which, the results are only representative of 2x up-sampling, for the B5, B6, and B7 band composites of Sentinel-2 imagery, in the 80→40m and 40→20m evaluation settings synthesised. To eliminate this burden, a series of future work would be necessary to trial the strategies recommended by this study, with 60m band composites, alternate 20m band composites, and various other scales and ranges of evaluation setting. Though, works of this kind go beyond the scope of this research. Which aims to reveal the feasibility of achieving spatial enhancements, cheaply, in support of more detailed and accurate information extraction, of lower-resolution satellite imaging. As the discoveries of this work are believed to demonstrate.

Chapter 6 | Conclusion

6.1 Research Summary

In this paper, multiple strategies are presented for spatially enhancing the lower-resolution band composites, of Sentinel-2 imagery. Of the three strategies that this work presents, geometric self-ensemble, on average, is performatively superior to both the stochastic translation scheme, adopted from [61], and the consolidation of the two strategies, that this work establishes. Wherein, the study also identifies optimal configurations for either strategy, in isolation, that are proposed to be used with other RS applications that regulate MS imagery. Relevant to this study's objectives, this research identifies that example-based SR models, specifically, can adopt the use of DA techniques at test-time, to advance the spatial resolution that they can achieve, in their LR-HR mapping reconstructions. As well, this work illustrates that a combinative use-case of the two strategies identified, promises performance gains over the baseline and reference methods purposed to the study. Such that it outperforms the stochastic translation scheme, in isolation, as well. Lastly, the configurations submitted for each of the strategies, identify that the sequence in which data augmentation techniques are applied and that the input image data is decomposed, can impact the peak spatial resolution attainable by an augmented model. As is presented for the stochastic translation scheme, only. With the limited experimentation conducted for this study aside, from the quantitative results that were obtained, this work proves to provide a novel contribution to the SISR domain, respective of RS interests.

6.4 Future Work

Suggested as future work for this study, it would be beneficial to investigate the lowest resolution band composites of Sentinel-2 data, which are sampled from a 60m GSD, in acquirement of more credible outcomes for the strategies and their configurations, like they are presented here. Moreover, with regards to the augmentation strategies themselves, the fidelity of the geometric self-ensemble trials could be investigated further, to consider additional transformation counts and constraints, that were not considered in the respective literature. As such, one would be investigating the performance implications of other existing, geometric transformations. Expanding from the investigations already led, per-band analysis could also be considered, in identifying the spectral correlations between the strategies deployed. And lastly, the performance implications of the strategies could be investigated for the lowest resolution satellite images, publicly available, as well. In more accurately acknowledging each strategies contribution to historical data analysis.

Chapter 7 | Critical Appraisal

7.1 Development Approach

Aligned with this works adherence to the hypotheses formulated, it is without doubt, that an adequate number of experiments were formulated, to enable each study aim to be acknowledged, in confidence. However, as originally proposed for this study, both lower-resolution band composites were intended to be exposed to augmentation, at the 20m and 60m GSD's. Such that the evaluation of the augmented strategies would have been conducted in the 40→20m and 360→60m settings, respectively, aligned with [18]. Given the hardware constraints posed on the study, however, the 60m bands could not be super-resolved due to the processing expense incurred by DSen2's approach to tile decomposition. Where instead, this study only investigated the augmentation of the B5, B6, and B7, 20m band composite, to address the hypotheses put forth. In which, evaluation was performed in both 80→40m and 40→20m settings, alternatively, as the DSen2 model was already trained at this scale, for the purposes of the authors own investigations. Thus, it is believed that this issue was alleviated accordingly. However, the results obtained for the experiments conducted, could then only be indicative of the lower-resolution bands, absent from the study; as this work makes references to, throughout. Nonetheless, this deviation from the original plan, led to the addressal of the hypotheses in question.

7.2 Development Adherence

Aligned with the study's development plan, disguised as the projects management schematic, the adherence to the plan is upon itself, deserving of recognition. As outlined in the proposal for the work presented, it was expected that a literature study was conducted, a model was configured and then extended, and a quantitative analysis of the data accumulated was delivered, in a timely fashion. Each of which, alongside this very document, comprises the underlying work packages of this study. In correspondence with the projects management schematic (see *Figure 32*), as it was recorded live throughout the study's undertaking, one can acknowledge that all work packages were delivered within an appropriate duration. As well, one can infer that the distribution of time was appropriately conceived.

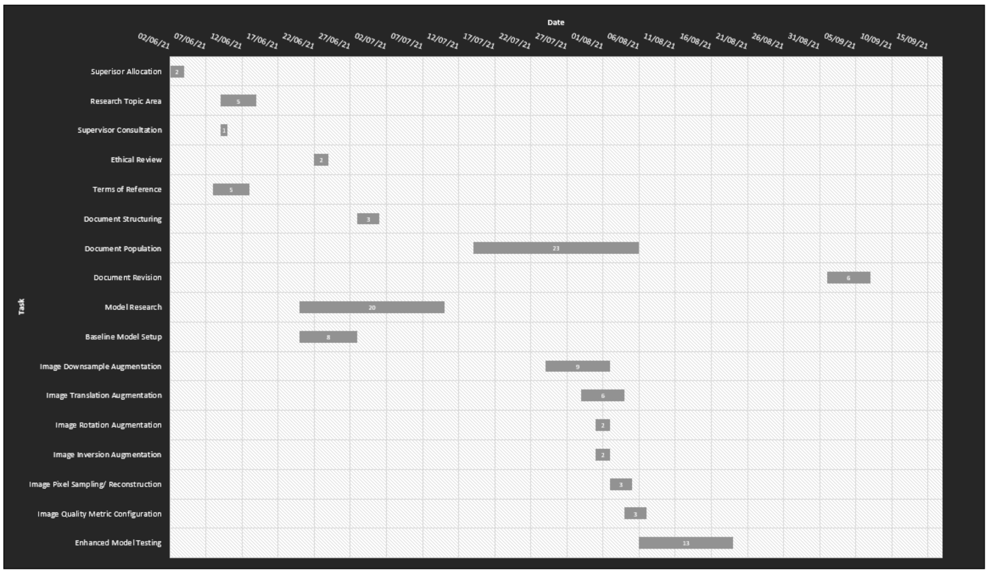


Figure 32: Visualisation of the projects management schematic, recorded live, in the format of a Gantt chart.

7.3 Academic Advancement

From undertaking this study, a plethora of knowledge has been acquired, over the course of the works research and development cycles. Most of which has surfaced from the research conducted into the concepts surrounding remote sensing, multispectral imagery, data augmentation, deep learning, and super resolution. As well as when conducting the literature study, in identifying the state-of-the-arts purposed for the investigations of this work. Aside from the accumulation of knowledge, time-keeping has also been exercised as an interpersonal competence, throughout this development, and as such, has evolved to a stricter adherence.

Bibliography

[1] Lin, C. and Bioucas-Dias, J.M. (2020) An Explicit and Scene-Adapted Definition of Convex Self-Similarity Prior With Application to Unsupervised Sentinel-2 Super-Resolution. In: *IEEE Transactions on Geoscience and Remote Sensing, March 2020*. New York: IEEE, pp. 3352-3365.

[2] Hoffmann, M., Butenko, Y. and Traore, S. (2018) Evaluation of Satellite Imagery to Increase Crop Yield in Irrigated Agriculture. *Agris on-line Papers in Economics and Informatics*. [Online] 10. Available from: https://www.researchgate.net/publication/327975697_Evaluation_of_Satellite_Imagery_to_Increase_Crop_Yield_in_Irrigated_Agriculture [Accessed: 21/08/21].

[3] Zhao, S., Wang, Q., Li, Y., Liu, S., Wang, Z., Zhu, L. and Wang, Z. (2017) An overview of satellite remote sensing technology used in China's environmental protection. *Earth Science Informatics*. [Online] Available from: https://www.researchgate.net/publication/312672567_An_overview_of_satellite_remote_sensing_technology_used_in_China%27s_environmental_protection [Accessed: 21/08/21].

[4] Joyce, K., Belliss, S.E., Samsonov, S., McNeill, S. and Glassey, P. (2009) A review of the status of satellite remote sensing and image processing techniques for mapping natural hazards and disasters. *Progress in Physical Geography*. [Online] 33. Available from: https://www.researchgate.net/publication/233921123_A_review_of_the_status_of_satellite_remote_sensing_and_image_processing_techniques_for_mapping_natural_hazards_and_disasters [Accessed: 21/08/21].

[5] Fonji, S.F. and Taff, G.N. (2014) Using satellite data to monitor land-use land-cover change in North-eastern Latvia. *SpringerPlus 3*. [Online] 61. Available from: https://springerplus.springeropen.com/articles/10.1186/2193-1801-3-61 [Accessed: 21/08/21].

[6] Sunar, F., Ulubay, A., Maktav, D. and Yagiz, E. (2005) The use of satellite image maps for urban planning in Turkey. *International Journal of Remote Sensing - INT J REMOTE SENS*. [Online] 26. Available from: https://www.researchgate.net/publication/240527278_The_use_of_satellite_image_maps_for_urban_planning_in_Turkey [Accessed: 21/08/21].

[7] Yang, J., Gong, P., Fu, R., Zhang, M., Chen, J., Liang, S., Xu, B., Shi, J. and Dickinson, R. (2013) The role of satellite remote sensing in climate change studies. *Nature Climate Change*. [Online] 3. Available from: https://www.researchgate.net/publication/259841888_The_role_of_satellite_remote_sensing_in_climate_change_studies [Accessed: 21/08/21].

[8] Kaku, K. (2018) Satellite remote sensing for disaster management support: A holistic and staged approach based on case studies in Sentinel Asia. *International Journal of Disaster Risk Reduction*. [Online] 33. Available from: https://www.researchgate.net/publication/328009106_Satellite_remote_sensing_for_disaster_management_support_A_holistic_and_staged_approach_based_on_case_studies_in_Sentinel_Asia [Accessed: 21/08/21].

[9] Gianinetto, M., Giussani, A., Lechi, G. and Scaioni, M. (2004) Fast mapping from high resolution satellite images: a sustainable approach to provide maps for developing countries. [Online]. Available from: https://www.researchgate.net/publication/233966477_Fast_mapping_from_high_resolution_satellite_images_a_sustainable_approach_to_provide_maps_for_developing_countries [Accessed: 21/08/21].

[10] Pettorelli, N., Safi, K. and Turner, W. (2014) Satellite remote sensing, biodiversity research and conservation of the future. *Philosophical transactions of the Royal Society of London, Series B, Biological sciences*. [Online] 369. Available from: https://www.researchgate.net/publication/261737513_Satellite_remote_sensing_biodiversity_research_and_conservation_of_the_future [Accessed: 21/08/21].

[11] Abdullah, A., Nassr, S. and Ghaleeb, A. (2013) Remote Sensing and Geographic Information System for Fault Segments Mapping a Study from Taiz Area, Yemen. *Journal of Geological Research*. [Online] 2013. Available from: https://www.hindawi.com/journals/jgr/2013/201757/ [Accessed: 21/08/21].

[12] Ma, J., Sun, W., Yang, G. and Zhang, D. (2018) Hydrological Analysis using Satellite Remote Sensing Big Data and CREST Model. *IEEE Access*. [Online]. Available from: https://www.researchgate.net/publication/323460536_Hydrological_Analysis_using_Satellite_Remote_Sensing_Big_Data_and_CREST_Model [Accessed: 21/08/21].

[13] Rajeesh, R. and Dwarakish, G.S. (2015) Satellite Oceanography – A review. *Aquatic Procedia*. [Online] 4. Available from: https://www.researchgate.net/publication/273792010_Satellite_Oceanography-_A_review [Accessed: 21/08/21].

[14] Curzi, G., Modenini, D. and Tortora, P. (2020) Large Constellations of Small Satellites: A Survey of Near Future Challenges and Missions. *Aerospace*. [Online] 7. Available from: https://www.researchgate.net/publication/344683515_Large_Constellations_of_Small_Satellites_A_Survey_of_Near_Future_Challenges_and_Missions [Accessed: 21/08/21].

[15] Froment, R., van Loenhout, J., Vanwambeke, S. and Guha-Sapir, D. (2020) STENTOR: Use of Earth Observation Satellites to Improve Effectiveness of Humanitarian Operations. [Online]. Available

from:
https://www.researchgate.net/publication/346679476_STENTOR_Use_of_Earth_Observation_Satellites_to_Improve_Effectiveness_of_Humanitarian_Operations [Accessed: 21/08/21].

[16] Brodu, N. (2017) Super-Resolving Multiresolution Images With Band-Independent Geometry of Multispectral Pixels. In: *IEEE Transactions on Geoscience and Remote Sensing, August 2017*. New York: IEEE, pp. 4610-4617.

[17] Romero, L.F.S., Marcello, J. and Vilaplana, V. (2020) Super-Resolution of Sentinel-2 Imagery Using Generative Adversarial Networks. *Remote Sensing*. [Online] 12. Available from: https://www.researchgate.net/publication/343302472_Super-Resolution_of_Sentinel-2_Imagery_Using_Generative_Adversarial_Networks [Accessed: 21/08/21].

[18] Lanaras, C., Bioucas-Dias, J., Galliani, S., Baltsavias, E. and Schindler, K. (2018) Super-resolution of Sentinel-2 images: Learning a globally applicable deep neural network. *ISPRS Journal of Photogrammetry and Remote Sensing*. [Online] 146. Available from: https://arxiv.org/abs/1803.04271 [Accessed: 21/08/21].

[19] Zhang, K., Sumbul, G. and Demir, B. (2020) An Approach To Super-Resolution Of Sentinel-2 Images Based On Generative Adversarial Networks. In: *2020 Mediterranean and Middle-East Geoscience and Remote Sensing Symposium (M2GARSS), June 2020*. New York: IEEE, pp. 69-72.

[20] eoPortal Directory (2021) *RADARSAT-2*. [Online] eoPortal Directory. Available from: https://earth.esa.int/web/eoportal/satellite-missions/r/radarsat-2 [Accessed: 22/08/21].

[21] Ose, K., Corpetti, T. and Demagistri, L. (2016) 2 - Multispectral Satellite Image Processing. In: BAGHDADI, N. and ZRIBI, M. (eds) *Optical Remote Sensing of Land Surface*. Amsterdam: Elsevier, pp 57-124.

[22] Zhang, R., Cavallaro, G. and Jitsev, J. (2020) Super-Resolution of Large Volumes of Sentinel-2 Images with High Performance Distributed Deep Learning. *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*. [Online]. Available from: https://ieeexplore.ieee.org/abstract/document/9323734 [Accessed: 22/08/21].

[23] Li, Y. and Li, B. (2021) Super-Resolution of Sentinel-2 Images at 10m Resolution without Reference Images. [Online]. Available from: https://www.researchgate.net/publication/351039119_Super-Resolution_of_Sentinel-2_Images_at_10m_Resolution_without_Reference_Images [Accessed: 22/08/21].

[24] Farsiu, S., Robinson, D., Elad, M. and Milanfar, P. (2004) Advances and challenges in Super-Resolution. *International Journal of Imaging Systems and Technology*. [Online] 14. Available from: https://www.researchgate.net/publication/228986240_Advances_and_challenges_in_Super-Resolution [Accessed: 22/08/21].

[25] Boyle, S., Kennedy, C., Torres, J., Colman, K., Pérez-Estigarribia, P.E. and De La Sancha, N. (2014) High-Resolution Satellite Imagery Is an Important yet Underutilized Resource in Conservation Biology. *PLOS ONE*. [Online] 9. Available from:

https://www.researchgate.net/publication/259920718_High-Resolution_Satellite_Imagery_Is_an_Important_yet_Underutilized_Resource_in_Conservation_Biology [Accessed: 22/08/21].

[26] NASA (2021) *Instruments*. [Online] NASA. Available from: https://aqua.nasa.gov/content/instruments [Accessed: 23/08/21].

[27] Kaku, K. (2019) Satellite remote sensing for disaster management support: A holistic and staged approach based on case studies in Sentinel Asia. *International Journal of Disaster Risk Reduction*. [Online] 33. Available from: https://www.sciencedirect.com/science/article/pii/S2212420918304801 [Accessed: 23/08/21].

[28] Xiong, Z., Lin, M., Lin, Z., Sun, T., Yang, G. and Wang, Z. (2020) Single image super-resolution via Image Quality Assessment-Guided Deep Learning Network. *PLOS ONE*. [Online] 15 (10). Available from: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0241313 [Accessed: 23/08/21].

[29] Isaac, J.S. and Kulkarni, R. (2015) Super resolution techniques for medical image processing. In: *2015 International Conference on Technologies for Sustainable Development (ICTSD), February 2015*. New York: IEEE, pp. 1-6.

[30] Armanious, K., Abdulatif, S., Aziz, F., Schneider, U. and Yang, B. (2019) An Adversarial Super-Resolution Remedy for Radar Design Trade-offs. In: *2019 27th European Signal Processing Conference (EUSIPCO), September 2019*. New York: IEEE, pp. 1-5,

[31] UI Hoque, M.R., Burks, R., Kwan, C. and Li, J. (2019) Deep Learning for Remote Sensing Image Super-Resolution. In: *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), October 2019*. New York: IEEE, pp. 0286-0292.

[32] Müller, M.U., Ekhtiari, N., Almeida, R.M. and Rieke, C. (2020). Super-resolution of multispectral satellite images using convolutional neural networks. [Online]. Available from: https://arxiv.org/abs/2002.00580 [Accessed: 23/08/21].

[33] Zhou, L. and Feng, S. (2019) A Review of Deep Learning for Single Image Super-Resolution. In: *2019 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), February 2019*. New York: IEEE, pp. 139-142.

[34] Galar, M., Sesma, R., Ayala, C., Albizua, L. and Aranda, C. (2020) LEARNING SUPER-RESOLUTION FOR SENTINEL-2 IMAGES WITH REAL GROUND TRUTH DATA FROM A REFERENCE SATELLITE. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*. [Online]. Available from: https://www.researchgate.net/publication/343402401_LEARNING_SUPER-RESOLUTION_FOR_SENTINEL-2_IMAGES_WITH_REAL_GROUND_TRUTH_DATA_FROM_A_REFERENCE_SATELLITE [Accessed: 23/08/21].

[35] Yang, W., Zhang, X., Tian, Y., Wang, W., Xue, J. and Liao, Q. (2019) Deep Learning for Single Image Super-Resolution: A Brief Review. In: *IEEE Transactions on Multimedia, May 2019*. New York: IEEE, pp. 3106-3121.

[36] Galar, M., Sesma, R., Ayala, C. and Aranda, C. (2019) SUPER-RESOLUTION FOR SENTINEL-2 IMAGES. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. [Online]. Available from: https://www.researchgate.net/publication/335874581_SUPER-RESOLUTION_FOR_SENTINEL-2_IMAGES [Accessed: 23/08/21].

[37] Gargiulo, M., Dell' Aglio, D.A.G., Lodice, A., Riccio, D. and Ruello, G. (2019) A CNN-based Super-resolution Technique for Active Fire Detection on Sentinel-2 Data. In: *2019 PhotonIcs & Electromagnetics Research Symposium - Spring (PIERS-Spring), March 2019*. New York: IEEE, pp. 418-426.

[38] Xiao, A., Wang, Z., Wang, L. and Ren, Y. (2018) Super-Resolution for "Jilin-1" Satellite Video Imagery via a Convolutional Network. *Sensors*. [Online] 18 (4). Available from: https://www.mdpi.com/1424-8220/18/4/1194/htm# [Accessed: 23/08/21].

[39] Goodfellow, I., Bengio, Y. and Courville, A. (2016) Deep Learning. *MIT Press*. Available from: http://www.deeplearningbook.org [Accessed: 23/08/21].

[40] Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2017) ImageNet classification with deep convolutional neural networks. *Communications of the ACM*. [Online] 60 (6). Available from: https://dl.acm.org/doi/10.1145/3065386 [Accessed: 23/08/21].

[41] Galar, M., Sesma, R., Ayala, C., Albizua, L. and Aranda, C. (2020) Super-Resolution of Sentinel-2 Images Using Convolutional Neural Networks and Real Ground Truth Data. *Remote Sensing*. [Online] 12 (18). Available from: https://www.mdpi.com/2072-4292/12/18/2941 [Accessed: 23/08/21].

[42] He, K., Zhang, X., Ren, S. and Sun, J. (2016) Deep Residual Learning for Image Recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), December 2016*. New York: IEEE, pp. 770-778.

[43] Ronneberger O. and Fischer P. and Brox T. (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N., Hornegger J., Wells W. and Frangi A. (eds) *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Basel: Springer, pp. 234-241.

[44] Deng, J., Guo, J., and Zafeiriou, S. (2019) ArcFace: Additive Angular Margin Loss for Deep Face Recognition. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), January 2020*. New York: IEEE, pp. 4685-4694.

[45] Scarpa, G., Gargiulo, M., Mazza, A. and Gaetano, R. (2018). A CNN-Based Fusion Method for Feature Extraction from Sentinel Data. *Remote Sensing*. [Online] 10 (2). Available from: https://www.researchgate.net/publication/322934928_A_CNN-Based_Fusion_Method_for_Feature_Extraction_from_Sentinel_Data [Accessed: 23/08/21].

[46] Kussul, N., Lavreniuk, M., Skakun, S., and Shelestov, A. (2017) Deep Learning Classification of Land Cover and Crop Types Using Remote Sensing Data. In: *IEEE Geoscience and Remote Sensing Letters*, *March 2017*. New York: IEEE, pp. 778-782.

[47] Stoian, A., Poulain, V., Inglada, J., Poughon, V. and Derksen, D. (2019) Land Cover Maps Production with High Resolution Satellite Image Time Series and Convolutional Neural Networks: Adaptations and Limits for Operational Systems. *Remote Sensing*. [Online] 11 (17). Available from: https://www.researchgate.net/publication/335400675_Land_Cover_Maps_Production_with_High_Resolution_Satellite_Image_Time_Series_and_Convolutional_Neural_Networks_Adaptations_and_Limits_for_Operational_Systems [Accessed: 23/08/21].

[48] Gargiulo, M., Dell' Aglio, D.A.G., Lodice, A., Riccio, D. and Ruello, G. (2020). Integration of Sentinel-1 and Sentinel-2 Data for Land Cover Mapping Using W-Net. *Sensors*. [Online] 20 (10). Available from: https://www.semanticscholar.org/paper/Integration-of-Sentinel-1-and-Sentinel-2-Data-for-Gargiulo-Dell%27Aglio/90402c8d85a2b83a49afa0b1eb321f1f8c0c9e11 [Accessed: 23/08/21].

[49] Li, Y. and Li, B. (2021) Super-Resolution of Sentinel-2 Images at 10m Resolution without Reference Images. [Online]. Available from: https://www.researchgate.net/publication/351039119_Super-Resolution_of_Sentinel-2_Images_at_10m_Resolution_without_Reference_Images [Accessed: 23/08/21].

[50] Bashir, S.M.A., Wang, Y., Khan, M. and Niu, Y. (2021) A comprehensive review of deep learning-based single image super-resolution. *PeerJ Computer Science*. [Online] 7 (9). Available from: https://arxiv.org/abs/2102.09351 [Accessed: 23/08/21].

[51] Hayat, K. (2017) Super-Resolution via Deep Learning. [Online]. Available from: https://arxiv.org/abs/1706.09077 [Accessed: 24/08/21].

[52] Yoo, J., Ahn, N. and Sohn, K. (2020) Rethinking Data Augmentation for Image Super-resolution: A Comprehensive Analysis and a New Strategy. In: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020*. New York: IEEE, pp. 8372-8381.

[53] Devries, T., and Taylor, G.W. (2017) Improved Regularization of Convolutional Neural Networks with Cutout. [Online]. Available from: https://arxiv.org/abs/1708.04552 [Accessed: 25/08/21].

[54] Hendrycks, D., and Dietterich, T.G. (2019) Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. [Online]. Available from: https://arxiv.org/abs/1903.12261 [Accessed: 25/08/21].

[55] Zhang, H., Cissé, M., Dauphin, Y., and Lopez-Paz, D. (2018). mixup: Beyond Empirical Risk Minimization. [Online]. Available from: https://arxiv.org/abs/1710.09412 [Accessed: 25/08/21].

[56] Zhao, Z., Zheng, P., Xu, S., and Wu, X. (2019) Object Detection With Deep Learning: A Review. IEEE Transactions on Neural Networks and Learning Systems. [Online] 30 (11). Available from: https://arxiv.org/abs/1807.05511 [Accessed: 25/08/21].

[57] Ciaparrone, G., Sánchez, F.L., Tabik, S., Troiano, L., Tagliaferri, R., and Herrera, F. (2020) Deep Learning in Video Multi-Object Tracking: A Survey. *Neurocomputing*. [Online] 381. Available from: https://arxiv.org/abs/1907.12740 [Accessed: 25/08/21].

[58] Zheng, C., Wu, W., Yang, T., Zhu, S., Chen, C., Liu, R., Shen, J., Kehtarnavaz, N., and Shah, M. (2020) Deep Learning-Based Human Pose Estimation: A Survey. [Online]. Available from: https://arxiv.org/abs/2012.13392 [Accessed: 25/08/21].

[59] Lim, B., Son, S., Kim, H., Nah, S., and Lee, K.M. (2017) Enhanced Deep Residual Networks for Single Image Super-Resolution. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), August 2017*. New York: IEEE, pp. 1132-1140.

[60] Du, C., Zewei, H., Anshun, S., Jiangxin, Y., Yanlong, C., Yanpeng, C., Siliang, T. and Yang, M.Y. (2019) Orientation-Aware Deep Neural Network for Real Image Super-Resolution. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), April 2019*. New York: IEEE, pp. 1944-1953.

[61] Thurnhofer-Hemsi, K., López-Rubio, E., Roé-Vellvé, N., Domínguez, E., and Molina-Cabello, M.A. (2018) Super-resolution of 3D Magnetic Resonance Images by Random Shifting and Convolutional Neural Networks. In: *2018 International Joint Conference on Neural Networks (IJCNN), October 2018*. New York: IEEE, pp. 1-8.

[62] The European Space Agency (2021) *Sentinel-2*. [Online] The European Space Agency. Available from: https://sentinel.esa.int/web/sentinel/missions/sentinel-2 [Accessed: 25/08/21].

[63] The European Space Agency (2021) *The European Space Agency*. [Online] The European Space Agency. Available from: https://www.esa.int/ [Accessed: 25/08/21].

[64] Copernicus (2021) *Copernicus Europe's Eyes on Earth*. [Online] Copernicus. Available from: https://www.copernicus.eu/en [Accessed: 25/08/21].

[65] The European Space Agency (2015) *Sentinel-1 Copernicus*. [Online] The European Space Agency. Available from: https://www.d-copernicus.de/fileadmin/Content/pdf/Sentinels_update_170510_final_printed.pdf [Accessed: 25/08/21].

[66] The European Space Agency (2021) *Mission Objectives*. [Online] The European Space Agency. Available from: https://sentinel.esa.int/web/sentinel/missions/sentinel-2/mission-objectives [Accessed: 25/08/21].

[67] The European Space Agency (2010) *GMES Sentinel-2 Mission Requirements Document*. [Online] The European Space Agency. Available from: https://esamultimedia.esa.int/docs/GMES/Sentinel-2_MRD.pdf [Accessed: 25/08/21].

[68] The European Space Agency (2021) *Resolution and Swath*. [Online] The European Space Agency. Available from: https://sentinel.esa.int/web/sentinel/missions/sentinel-2/instrument-payload/resolution-and-swath [Accessed: 26/08/21].

[69] NASA (2021) *TERRA The EOS Flagship*. [Online] NASA. Available from: https://terra.nasa.gov/data/modis-data [Accessed: 26/08/21].

[70] NASA (2021) *Aqua Project Science*. [Online] NASA. Available from: https://aqua.nasa.gov/content/images-data [Accessed:26/08/21].

[71] Hay, S.I. (2000) An overview of remote sensing and geodesy for epidemiology and public health application. *Advances in parasitology*. [Online] 47. Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3164799/ [Accessed: 26/08/21].

[72] Zhu, L., Suomalainen, J., Liu, J., Hyyppä, J., Kaartinen, H. and Haggrén, H. (2018) A Review: Remote Sensing Sensors. In: *Multi-purposeful Application of Geospatial Data*. London: IntechOpen, pp. 19-42.

[73] Waghmare, B. and Suryawanshi, M. (2017) A Review- Remote Sensing. *International Journal of Engineering Research and Applications*. [Online] 7 (6). Available from: https://www.researchgate.net/publication/317633797_A_Review-_Remote_Sensing [Accessed: 26/08/21].

[74] Navalgund, R., V, J. and Roy, P. (2007) Remote sensing applications: An overview. *Current science*. [Online] 93 (2). Available from: https://www.researchgate.net/publication/255616101_Remote_sensing_applications_An_overview [Accessed: 26/08/21].

[75] EARTHDATA (2021) *What is Remote Sensing?* [Online] EARTHDATA. Available from: https://earthdata.nasa.gov/learn/backgrounders/remote-sensing [Accessed: 26/08/21].

[76] Madl, P. and Egot-Lemaire, S. (2015) The field and the photon from a physical point of view. In: FELS, D., CIFRA, M. and SCHOLKMANN, F. (eds.) *The field and the photon from a physical point of view*. 1st Edition. Berlin: ResearchGate, pp. 25-53.

[77] USGS (2021) *Landsat Missions*. [Online] USGS. Available from: https://www.usgs.gov/core-science-systems/nli/landsat/landsat-satellite-missions?qt-science_support_page_related_con=0#qt-science_support_page_related_con [Accessed: 26/08/21].

[78] The European Space Agency (2021) *SPOT*. [Online] The European Space Agency. Available from: https://earth.esa.int/eogateway/missions/spot [Accessed: 26/08/21].

[79] The European Space Agency (2021) *GeoEye-1*. [Online] The European Space Agency. Available from: https://earth.esa.int/eogateway/missions/geoeye-1 [Accessed: 26/08/21].

[80] The European Space Agency (2021) *WorldView Series*. [Online] The European Space Agency. Available from: https://earth.esa.int/eogateway/missions/worldview [Accessed: 26/08/21].

[81] The European Space Agency (2015) *Sentinel-2 User Handbook*. [Online] The European Space Agency. Available from: https://sentinel.esa.int/documents/247904/685211/Sentinel-2_User_Handbook [Accessed: 26/08/21].

[82] Radiant Earth Foundation (2017) *The View From Above*. [Online] Radiant Earth Foundation. Available from: https://www.radiant.earth/infographic/the-view-from-above/ [Accessed: 26/08/21].

[83] Kairu, E. (1982). An introduction to remote sensing. *GeoJournal*. [Online] 6 (3). Available from: https://www.researchgate.net/publication/226233143_An_introduction_to_remote_sensing/ [Accessed: 26/08/21].

[84] Kwan, C. (2019) Methods and Challenges Using Multispectral and Hyperspectral Images for Practical Change Detection Applications. *Information*. [Online] 10 (11). Available from: https://www.mdpi.com/2078-2489/10/11/353 [Accessed: 26/08/21].

[85] Edmund Optics (2021) *Hyperspectral and Multispectral Imaging*. [Online] Edmund Optics. Available from: https://www.edmundoptics.co.uk/knowledge-center/application-notes/imaging/hyperspectral-and-multispectral-imaging/ [Accessed: 26/08/21].

[86] Bandara, W.G.C., Prabhath, G.W.K., Dissanayake, D.W.S.C.B., Herath, H.M.V.R., Godaliyadda, G.M.R.I., Ekanayake, M.P.B., Vithana, S.S.P., Demini, S.M.D. and Madhunjith, T. (2018) A Multispectral Imaging System to Assess Meat Quality. In: *2018 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), December 2018*. New York: IEEE, pp. 1-6.

[87] Khodabakhshian, R., Emadi, B., Khojastehpour, M. and Golzarian, M.R. (2017) Determining quality and maturity of pomegranates using multispectral imaging. *Journal of the Saudi Society of Agricultural Sciences*. [Online] 16 (4). Available from: https://www.sciencedirect.com/science/article/pii/S1658077X15300758 [Accessed: 26/08/21].

[88] Historic England (2021) *Satellite Imagery, Muli and Hyper-Spectral Data*. [Online] Historic England. Available from: https://historicengland.org.uk/research/methods/airborne-remote-sensing/satellite-and-ms-imagery/ [Accessed: 26/08/21].

[89] Borsoi, R.A., Imbiriba, T. and Bermudez, J.C.M. (2019) Super-Resolution for Hyperspectral and Multispectral Image Fusion Accounting for Seasonal Spectral Variability. In: *IEEE Transactions on Image Processing, July 2019*. New York: IEEE, pp. 116-127.

[90] Jia, X. (2017) Image recognition method based on deep learning. In: *2017 29th Chinese Control And Decision Conference (CCDC), May 2017*. New York: IEEE, pp. 4730-4735.

[91] Vaishnnave, M.P., Devi, S.K. and Srinivasan, P. (2019) A Study on Deep Learning Models for Satellite Imagery. *International Journal of Applied Engineering Research*. [Online] 14 (4). Available from: https://www.ripublication.com/ijaer19/ijaerv14n4_06.pdf [Accessed: 04/09/21].

[92] Du, X., Cai, Y., Wang, S. and Zhang, L. (2016) Overview of deep learning. In: 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), January 2016. New York: IEEE, pp. 159-164.

[93] Mahony, N.O., Campbell, S., Carvalho, A., Harapanahalli, S., Velasco-Hernández, G., Krpalkova, L., Riordan, D. and Walsh, J. (2019). Deep Learning vs. Traditional Computer Vision. *Computer Vision and Pattern Recognition*. [Online]. Available from: https://www.semanticscholar.org/paper/Deep-Learning-vs.-Traditional-Computer-Vision-Mahony-Campbell/dea1903672cc2c548ed7053845991f403478e8be [Accessed: 04/09/21].

[94] Zhang, W.J., Yang, G., Lin, Y., Ji, Ci. and Gupta, M.M. (2018) On Definition of Deep Learning. In: *2018 World Automation Congress (WAC), June 2018*. New York: IEEE, pp. 1-5.

[95] Lauzon, F.Q. (2012) An introduction to deep learning. In: *2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA), July 2012*. New York: IEEE, pp. 1438-1439.

[96] Sugiarto, I. and Pasila, F. (2018) Understanding a Deep Learning Technique through a Neuromorphic System a Case Study with SpiNNaker Neuromorphic Platform. *MATEC Web of Conferences*. [Online] 164 (1). Available from: https://www.researchgate.net/publication/324700732_Understanding_a_Deep_Learning_Technique_through_a_Neuromorphic_System_a_Case_Study_with_SpiNNaker_Neuromorphic_Platform [Accessed: 04/09/21].

[97] Vargas, R., Mosavi, A. and Ruiz, R. (2018). Deep Learning: A Review. [Online]. Available from: https://www.semanticscholar.org/paper/Deep-Learning%3A-A-Review-Vargas-Mosavi/33de241c5afe0941650a43620aee84ea7a75377f [Accessed: 04/09/21].

[98] Otter, D.W., Medina, J.R. and Kalita, J.K. (2021) A Survey of the Usages of Deep Learning for Natural Language Processing. *IEEE Transactions on Neural Networks and Learning Systems*. [Online] 32 (2). Available from: https://ieeexplore.ieee.org/abstract/document/9075398 [Accessed: 04/09/21].

[99] Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H. and He, Q. (2021) A Comprehensive Survey on Transfer Learning. In: *Proceedings of the IEEE, vol. 109, no. 1, January 2021*. New York: IEEE, pp. 43-76.

[100] Grefenstette, E., Blunsom, P., Freitas, N.D. and Hermann, K. (2014) A Deep Architecture for Semantic Parsing. *Computer Science*. [Online]. Available from: https://www.semanticscholar.org/paper/A-Deep-Architecture-for-Semantic-Parsing-Grefenstette-Blunsom/2c19e3f01ed899d166d771d2593bb3cb20740681 [Accessed: 04/09/21].

[101] Wang, Y., Perazzi, F., Mcwilliams, B., Sorkine-H, A., Sorkine-H, O., and Schroers, C. (2018) A Fully Progressive Approach to Single-Image Super-Resolution. [Online]. Available from: https://arxiv.org/abs/1804.02900 [Accessed: 04/09/21].

[102] Ledig, C., Theis, L., Huszár, F., Caballero, J., Aitken, A.P., Tejani, A., Totz, J., Wang, Z. and Shi, W. (2017) Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017*. New York: IEEE, pp. 105-114.

[103] Xu, Y., Li, J., Song, H. and Du, L. (2021) Single-Image Super-Resolution Using Panchromatic Gradient Prior and Variational Model. *Mathematical Problems in Engineering*. [Online] 2021. Available from: https://www.semanticscholar.org/paper/Single-Image-Super-Resolution-Using-Panchromatic-Xu-Li/8dd714813e647b52217d2372e8bcd9d7627f79ae [Accessed: 04/09/21].

[104] Yang, J. and Huang, T. (2017) Image Super-Resolution: Historical Overview and Future Challenges. [Online]. Available from: https://www.researchgate.net/publication/333960480_Image_Super-Resolution_Historical_Overview_and_Future_Challenges [Accessed: 05/09/21].

[105] Yang, W., Zhang, X., Tian, Y., Wang, W., Xue, J. and Liao, Q. (2019) Deep Learning for Single Image Super-Resolution: A Brief Review. *IEEE Transactions on Multimedia*. [Online] 21 (12). Available from: https://ieeexplore.ieee.org/document/8723565 [Accessed: 05/09/21].

[106] Chaudhuri, S. (2001) Super-Resolution Imaging. [Online]. Available from: https://www.semanticscholar.org/paper/Super-Resolution-Imaging-Chaudhuri/d9b72a6014927cc5e36bf71f14cab9e9d70e888f [Accessed: 05/09/21].

[107] Tsai, R. and Huang, T.S. (1984) Multiframe image restoration and registration. [Online]. Available from: https://www.semanticscholar.org/paper/Multiframe-image-restoration-and-registration-Tsai-Huang/0b98e71885239ee1eed204009502bb047ae2c7ce [Accessed: 05/09/21].

[108] Irani, M. and Peleg, S. (1990). Super resolution from image Sequences. [Online]. Available from: https://www.researchgate.net/publication/3505000_Super_resolution_from_image_Sequences [Accessed: 05/09/21].

[109] Dong, C., Loy, C.C., He, K. and Tang, X. (2016) Image Super-Resolution Using Deep Convolutional Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. [Online] 38 (2). Available from: https://ieeexplore.ieee.org/document/7115171?denied= [Accessed: 05/09/21].

[110] Ghaffar, M.A., McKinstry, A., Maul, T. and Vu, T. (2019) Data Augmentation Approaches for Satellite Image Super-Resolution. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. [Online]. Available from: https://www.semanticscholar.org/paper/Data-Augmentation-Approaches-for-Satellite-Image-Ghaffar-McKinstry/088e13393198e43575a5d8d28abe4acb196467f7 [Accessed: 05/09/21].

[111] Ying, X. (2019) An Overview of Overfitting and its Solutions. *Journal of Physics: Conference Series*. [Online]. Available from: https://www.researchgate.net/publication/331677125_An_Overview_of_Overfitting_and_its_Solutions [Accessed: 05/09/21].

[112] Das, N., Sharma, N., Shebare V., Dawda, P., Gourkhede, P. and Gandhewar, N. (2021) Deep Learning Based Image Enhancement using Super Resolution. *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)*. [Online] 4 (3). Available from: http://ijarsct.co.in/ICISET031.pdf [Accessed: 05/09/21].

[113] Wang, G., Li, W., Aertsen, M., Deprest, J., Ourselin, S. and Vercauteren, T. (2018) Test-time augmentation with uncertainty estimation for deep learning-based medical image segmentation. [Online]. Available from: https://www.researchgate.net/publication/326505635_Test-time_augmentation_with_uncertainty_estimation_for_deep_learning-based_medical_image_segmentation [Accessed: 05/09/21].

[114] Ghaffar, M.A., McKinstry, A., Maul, T. and Vu, T. (2019). Data Augmentation Approaches for Satellite Image Super-Resolution. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. [Online]. Available from: https://www.semanticscholar.org/paper/Data-Augmentation-Approaches-for-Satellite-Image-Ghaffar-McKinstry/088e13393198e43575a5d8d28abe4acb196467f7 [Accessed: 05/09/21].

[115] Shorten, C. and Khoshgoftaar, T. (2019) A survey on Image Data Augmentation for Deep Learning. Journal of Big Data. [Online] 6. Available from: https://www.semanticscholar.org/paper/A-survey-on-Image-Data-Augmentation-for-Deep-Shorten-Khoshgoftaar/3813b88a4ec3c63919df47e9694b577f4691f7e5 [Accessed: 05/09/21].

[116] Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012) ImageNet classification with deep convolutional neural networks. *Communications of the ACM*. [Online] 60. Available from: https://www.semanticscholar.org/paper/ImageNet-classification-with-deep-convolutional-Krizhevsky-Sutskever/abd1c342495432171beb7ca8fd9551ef13cbd0ff [Accessed: 05/09/21].

[117] Simonyan, K. and Zisserman, A. (2015) Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*. [Online]. Available from: https://www.semanticscholar.org/paper/Very-Deep-Convolutional-Networks-for-Large-Scale-Simonyan-Zisserman/eb42cf88027de515750f230b23b1a057dc782108 [Accessed: 12/09/21].

[118] He, K., Zhang, X., Ren, S. and Sun, J. (2016) Deep Residual Learning for Image Recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018*. New York: IEEE, pp. 770-778.

[119] Özmen, B. (2019) AutoML for Data Augmentation. [Weblog] *Insight*. 27th May. Available from: https://blog.insightdatascience.com/automl-for-data-augmentation-e87cf692c366 [Accessed: 12/09/21].

[120] Shanmugam, D., Blalock, D.W., Balakrishnan, G. and Guttag, J. (2020) When and Why Test-Time Augmentation Works. [Online]. Available from: https://www.semanticscholar.org/paper/When-and-Why-Test-Time-Augmentation-Works-Shanmugam-Blalock/7edf8a4592b8c6b6cf0f9dbfdb3c844da3568c65 [Accessed: 12/09/21].

[121] Timofte, R., Rothe, R. and Gool, L. (2016) Seven Ways to Improve Example-Based Single Image Super Resolution. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016*. New York: IEEE, pp. 1865-1873.

[122] Molchanov, D., Lyzhov, A., Molchanova, Y., Ashukha, A. and Vetrov, D. (2020) Greedy Policy Search: A Simple Baseline for Learnable Test-Time Augmentation. *UAI*. [Online]. Available from:

https://www.semanticscholar.org/paper/Greedy-Policy-Search%3A-A-Simple-Baseline-for-Molchanov-Lyzhov/8286fbad6d4712d950689930720134bc8828963f [Accessed: 12/09/21].

[123] Kim, I., Kim, Y., & Kim, S. (2020). Learning Loss for Test-Time Augmentation. [Online]. Available from: https://www.semanticscholar.org/paper/Learning-Loss-for-Test-Time-Augmentation-Kim-Kim/ed6f5f1061fc39897b024167fb863e57a5be5037 [Accessed: 12/09/21].

[124] Araújo, V.C. (2021) Ensembles of Single Image Super-Resolution Generative Adversarial Networks, Dissertation. [Online]. Available from: http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1531779&dswid=9100 [Accessed: 12/09/21].

[125] Tonooka, H. (2005) Resolution enhancement of ASTER shortwave and thermal infrared bands based on spectral similarity. *SPIE Asia-Pacific Remote Sensing*. [Online]. Available from: https://www.semanticscholar.org/paper/Resolution-enhancement-of-ASTER-shortwave-and-bands-Tonooka/f8bd4b1a42e1563e4b34bacb8942e4962d5497eb [Accessed: 12/09/21].

[126] Fasbender, D., Tuia, D., Bogaert, P. and Kanevski, M. (2008) Support-Based Implementation of Bayesian Data Fusion for Spatial Enhancement: Applications to ASTER Thermal Images. *IEEE Geoscience and Remote Sensing Letters*. [Online] 5 (4). Available from: https://ieeexplore.ieee.org/document/4656478 [Accessed: 12/09/21].

[127] Trishchenko, A., Luo, Y. and Khlopenkov, K. (2006) A method for downscaling MODIS land channels to 250-m spatial resolution using adaptive regression and normalization. *SPIE Remote Sensing*. [Online]. Available from: https://www.semanticscholar.org/paper/A-method-for-downscaling-MODIS-land-channels-to-and-Trishchenko-Luo/e06c03dafafd2a36cd91fb8a3eb4fa3dc4c0c7bd [Accessed: 12/09/21].

[128] Sirguey, P., Mathieu, R., Arnaud, Y., Khan, M.M. and Chanussot, J. (2008) Improving MODIS Spatial Resolution for Snow Mapping Using Wavelet Fusion and ARSIS Concept. *IEEE Geoscience and Remote Sensing Letters*. [Online] 5 (1). Available from: https://ieeexplore.ieee.org/document/4382933 [Accessed: 12/09/21].

[129] Picaro, G., Addesso, P., Restaino, R., Vivone, G., Picone, D. and Dalla Mura, M. (2016) Thermal sharpening of VIIRS data. In: *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), July 2016*. New York: IEEE, pp. 7260-7263.

[130] Liu, X. and Wang, M. (2021) Super-Resolution of VIIRS-Measured Ocean Color Products Using Deep Convolutional Neural Network. *IEEE Transactions on Geoscience and Remote Sensing*. [Online] 59 (1). Available from: https://ieeexplore.ieee.org/document/9097401 [Accessed: 12/09/21].

[131] Lanaras, C., Bioucas-D, J., Baltsavias, E. and Schindler, K. (2017) Super-Resolution of Multispectral Multiresolution Images from a Single Sensor. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), July 2017*. New York: IEEE, pp. 1505-1513.

[132] Yang, D., Li, Z., Xia, Y. and Chen, Z. (2015) Remote sensing image super-resolution: Challenges and approaches. In: *2015 IEEE International Conference on Digital Signal Processing (DSP), July 2015*. New York: IEEE, pp. 196-200.

[133] Yang, Chih-Y., Ma, C. and Yang, Ming-H. (2014) Single-Image Super-Resolution: A Benchmark. [Online]. Available from: https://www.researchgate.net/publication/278693824_Single-Image_Super-Resolution_A_Benchmark [Accessed: 12/09/21].

[134] Duchon, C. (1979) Lanczos Filtering in One and Two Dimensions. *Journal of Applied Meteorology - J APPL METEOROL*. [Online] 18 (8). Available from: https://www.researchgate.net/publication/252898828_Lanczos_Filtering_in_One_and_Two_Dimensions [Accessed: 12/09/21].

[135] Yang, Y., Li, Q., Yang, C., Fu, Y., Feng, H., Xu, Z. and Chen, Y. (2020) Deep Networks With Detail Enhancement for Infrared Image Super-Resolution. *IEEE Access*. [Online] 8. Available from: https://ieeexplore.ieee.org/document/9171268 [Accessed: 12/09/21].

[136] Keys, R. (1981) Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*. [Online] 29 (6). Available from: https://ieeexplore.ieee.org/document/1163711 [Accessed: 12/09/21].

[137] Rifman, S. (1973) Digital rectification of ERTS multispectral imagery. [Online]. Available from: https://www.semanticscholar.org/paper/Digital-rectification-of-ERTS-multispectral-imagery-Rifman/843946db06e9dcaf918ab9e7f33b57b521b35bda [Accessed: 12/09/21].

[138] Irani, M. and Peleg, S. (1991) Improving resolution by image registration. CVGIP: Graphical Models and Image Processing. [Online]. Available from: https://www.researchgate.net/publication/222444142_Improving_resolution_by_image_registration [Accessed: 12/09/21].

[139] Getreuer, P. (2011) Linear Methods for Image Interpolation. *Image Processing On Line*. [Online]. Available from: https://www.semanticscholar.org/paper/Linear-Methods-for-Image-Interpolation-Getreuer/9ac1e42439d4532409233c69fdd9b98ae21d3f0e [Accessed: 12/09/21].

[140] Hou, H. and Andrews, H. (1978) Cubic splines for image interpolation and digital filtering. *IEEE Transactions on Acoustics, Speech, and Signal Processing*. [Online] 26 (6). Available from: https://ieeexplore.ieee.org/document/1163154 [Accessed: 12/09/21].

[141] Gaidhani, P. (2020) Super-resolution. *Education Information UK*. [Online]. Available from: https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/AV1011/Super_Resolution_CVonline.pdf [Accessed: 12/09/21].

[142] Srinivasan, K. and Kanakaraj, J. (2014) An Overview of SR Techniques Applied to Images, Videos and Magnetic Resonance Images. *Smart Computing Review*. [Online] 4 (3). Available from: https://www.researchgate.net/publication/301771127_An_Overview_of_SR_Techniques_Applied_to_Images_Videos_and_Magnetic_Resonance_Images [Accessed: 12/09/21].

[143] Freeman, W.T., Jones, T.R. and Pasztor, E.C. (2002) Example-based super-resolution. *IEEE Computer Graphics and Applications*. [Online] 22 (2). Available from: https://ieeexplore.ieee.org/abstract/document/988747 [Accessed: 12/09/21].

[144] Chang, H., Yeung, Dit-Y. and Xiong, Y. (2004) Super-resolution through neighbor embedding. In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, July 2004. New York: IEEE, pp. I-I.

[145] Keshk, H.M. and Yin, X. (2017) Satellite super-resolution images depending on deep learning methods: A comparative study. In: *2017 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), October 2017*. New York: IEEE, pp. 1-7.

[146] Suetake, N., Sakano, M. and Uchino, E. (2008) Image super-resolution based on local self-similarity. *Optical Review*. [Online] 15 (1). Available from: https://www.researchgate.net/publication/225968775_Image_super-resolution_based_on_local_self-similarity [Accessed: 12/09/21].

[147] Glasner, D., Bagon, S. and Irani, M. (2009) Super-resolution from a single image. In: *2009 IEEE 12th International Conference on Computer Vision*, October 2009. New York: IEEE, pp. 349-356.

[148] Baker, S. and Kanade, T. (2002) Limits on super-resolution and how to break them. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. [Online] 24 (9). Available from: https://ieeexplore.ieee.org/document/1033210 [Accessed: 12/09/21].

[149] Sun, J., Xu, Z. and Shum, Heung-Y. (2008) Image super-resolution using gradient profile prior. In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*, *June 2008*. New York: IEEE, pp. 1-8.

[150] Tai, Y., Liu, S., Brown, M.S. and Lin, S. (2010) Super resolution using edge prior and single image detail synthesis. In: *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 2400-2407.

[151] Freedman, G. and Fattal, R. (2011) Image and video upscaling from local self-examples. *ACM Trans. Graph*. [Online] 30. Available from: https://www.semanticscholar.org/paper/Image-and-video-upscaling-from-local-self-examples-Freedman-Fattal/95064fdabc04e1621ec1e46ff85f74163ba2f766 [Accessed: 12/09/21].

[152] Kim, K.I. and Kwon, Y. (2010) Single-Image Super-Resolution Using Sparse Regression and Natural Image Prior. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*. [Online] 32 (6). Available from: https://ieeexplore.ieee.org/document/5396341 [Accessed: 12/09/21].

[153] Kim, K.I. and Kwon, Y. (2008) Example-Based Learning for Single-Image Super-Resolution. In: Rigoll G. (eds) *Pattern Recognition*. DAGM 2008. Lecture Notes in Computer Science, volume 5096. Springer: Berlin, Heidelberg.

[154] Vincent, P. and Bengio, Y. (2002) Kernel Matching Pursuit. *Machine Learning*. [Online] 48 (1). Available from: https://www.researchgate.net/publication/226966322_Kernel_Matching_Pursuit [Accessed: 12/09/21].

[155] Ruder, S. (2016) An overview of gradient descent optimization algorithms. [Online]. Available from: https://www.semanticscholar.org/paper/An-overview-of-gradient-descent-optimization-Ruder/769ef3d5021cd71c37d2c403f231a53d1accf786 [Accessed: 12/09/21].

[156] Xiong, Z., Sun, X. and Wu, F. (2010) Robust Web Image/Video Super-Resolution. IEEE transactions on image processing: a publication of the IEEE Signal Processing Society. [Online] 19 (8). Available from: https://www.researchgate.net/publication/42254951_Robust_Web_ImageVideo_Super-Resolution [Accessed: 12/09/21].

[157] Yang, J., Wright, J., Huang, T. and Ma, Yi. (2008) Image super-resolution as sparse representation of raw image patches. In: *2008 IEEE Conference on Computer Vision and Pattern Recognition*, June 2008. New York: IEEE, pp. 1-8.

[158] Yang, J., Wright, J., Huang, T.S. and Ma, Y. (2010) Image Super-Resolution Via Sparse Representation. *IEEE Transactions on Image Processing*. [Online] 19 (11). Available from: https://ieeexplore.ieee.org/document/5466111 [Accessed: 12/09/21].

[159] Donoho, D.L. (2006) Compressed sensing. *IEEE Transactions on Information Theory*. [Online] 52 (4). Available from: https://ieeexplore.ieee.org/document/1614066 [Accessed: 12/09/21].

[160] Nikolakopoulos, K. (2008) Comparison of Nine Fusion Techniques for Very High Resolution Data. *Photogrammetric Engineering and Remote Sensing*. [Online] 74. Available from: https://www.semanticscholar.org/paper/Comparison-of-Nine-Fusion-Techniques-for-Very-High-Nikolakopoulos/70451ca0a9a33cb8eb3eb4af91d54efcbca993f7 [Accessed: 12/09/21].

[161] Amro, I., Mateos, J., Vega, M., Molina, R. and Katsaggelos, A. (2011) A survey of classical methods and new trends in pansharpening of multispectral images. *EURASIP Journal on Advances in Signal Processing*. [Online] 2011 (1). Available from: https://www.researchgate.net/publication/252063308_A_survey_of_classical_methods_and_new_trends_in_pansharpening_of_multispectral_images [Accessed: 12/09/21].

[162] Xu, Y., Li, J., Song, H. and Du., L. (2021) Single-Image Super-Resolution Using Panchromatic Gradient Prior and Variational Model. *Mathematical Problems in Engineering*. [Online] 2021. Available from: https://www.hindawi.com/journals/mpe/2021/9944385/ [Accessed: 12/09/21].

[163] Garzelli, A., Nencini, F., Alparone, L., Aiazzi, B. and Baronti, S. (2004) Pan-sharpening of multispectral images: a critical review and comparison. In: *IGARSS 2004, 2004 IEEE International Geoscience and Remote Sensing Symposium, September 2004*. New York: IEEE, pp. 84.

[164] Li, X., Xu, F., Lyu, X., Tong, Y., Chen, Z., Li, Shengyang and Liu, D. (2020) A Remote-Sensing Image Pan-Sharpening Method Based on Multi-Scale Channel Attention Residual Network. *IEEE Access*. [Online] 8. Available from: https://ieeexplore.ieee.org/document/8981952 [Accessed: 12/09/21].

[165] Zhang, M., Su, W., Fu, Y., Zhu, D., Xue, Jing-H., Huang, J., Wang, W., Wu, J. and Yao, C. (2019) Super-resolution enhancement of Sentinel-2 image for retrieving LAI and chlorophyll content of

summer corn. *European Journal of Agronomy*. [Online] 111. Available from: https://www.researchgate.net/publication/335684145_Super-resolution_enhancement_of_Sentinel-2_image_for_retrieving_LAI_and_chlorophyll_content_of_summer_corn [Accessed: 12/09/21].

[166] Armannsson, S., Ulfarsson, M., Sigurdsson, J., Han, N.V. and Sveinsson, J. (2021) A Comparison of Optimized Sentinel-2 Super-Resolution Methods Using Wald's Protocol and Bayesian Optimization. *Remote Sensing*. [Online] 13 (11). Available from: https://www.researchgate.net/publication/352169492_A_Comparison_of_Optimized_Sentinel-2_Super-Resolution_Methods_Using_Wald%27s_Protocol_and_Bayesian_Optimization [Accessed: 12/09/21].

[167] Brodu, N. (2017) Super-Resolving Multiresolution Images With Band-Independent Geometry of Multispectral Pixels. *IEEE Transactions on Geoscience and Remote Sensing*. [Online] 55. Available from: https://www.semanticscholar.org/paper/Super-Resolving-Multiresolution-Images-With-of-Brodu/75dc0712f41ac708c5382cb6cb1d58d42bbf5043 [Accessed: 12/09/21].

[168] Perko, R. (2004) Computer Vision for Large Format Digital Aerial Cameras. [Online]. Available from: https://www.researchgate.net/publication/308121983_Computer_Vision_for_Large_Format_Digital_Aerial_Cameras [Accessed: 12/09/21].

[169] Vivone, G., Alparone, L., Chanussot, J., Mura, M., Garzelli, A., Licciardi, G., Restaino, R. and Wald, L. (2015) A Critical Comparison Among Pansharpening Algorithms. *IEEE Transactions on Geoscience and Remote Sensing*. [Online] 53. Available from: https://www.semanticscholar.org/paper/A-Critical-Comparison-Among-Pansharpening-Vivone-Alparone/38328711715c407b300abdb2a9b173c600e96b0b [Accessed: 12/09/21].

[170] Li, J., Yuan, G. and Fan, H. (2019) Multispectral Image Fusion Using Fractional-Order Differential and Guided Filtering. *IEEE Photonics Journal*. [Online] 99. Available from: https://www.researchgate.net/publication/336067525_Multispectral_Image_Fusion_Using_Fractional-Order_Differential_and_Guided_Filtering [Accessed: 12/09/21].

[171] Tu, Te-M., Su, Shun-C., Shyu, Hsuen-C. and Huang, P.S. (2001) A new look at IHS-like image fusion methods. *Information Fusion*. [Online] 2 (3). Available from: https://www.sciencedirect.com/science/article/abs/pii/S1566253501000367 [Accessed: 12/09/21].

[172] Tu, Te-M., Huang, P.S., Hung, Chung-L. and Chang, Chien-P. (2004) A fast intensity-hue-saturation fusion technique with spectral adjustment for IKONOS imagery. *IEEE Geoscience and Remote Sensing Letters*. [Online] 1 (4). Available from: https://ieeexplore.ieee.org/document/1347129 [Accessed: 12/09/21].

[173] GISGeography (2021) *IKONOS Satellite Imagery: First Commercial Space-based Imaging*. [Online] GISGeography. Available from: https://gisgeography.com/ikonos-satellite/ [Accessed: 12/09/21].

[174] Yang, C., Zhan, Q., Liu, H. and Ma, R. (2018) An IHS-Based Pan-Sharpening Method for Spectral Fidelity Improvement Using Ripplet Transform and Compressed Sensing. *Sensors*. [Online]

18 (11). Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6263882/ [Accessed: 12/09/21].

[175] Jr, P.S.C. and Kwarteng, A.Y. (1989) Extracting spectral contrast in Landsat Thematic Mapper image data using selective principal component analysis. Photogrammetric Engineering and Remote Sensing. [Online] 55 (3). Available from: https://www.researchgate.net/publication/262301549_Extracting_spectral_contrast_in_Landsat_Thematic_Mapper_image_data_using_selective_principal_component_analysis [Accessed: 12/09/21].

[176] Laben, C.A. and Brower, B.V. (1998) Process for enhancing the spatial resolution of multispectral imagery using pan-sharpening. [Online]. Available from: https://patents.google.com/patent/US6011875A/en [Accessed: 13/09/21].

[177] Tu, Te-M., Hsu, Ching-L., Tu, Pin-Y. and Lee, Ching-H. (2012) An Adjustable Pan-Sharpening Approach for IKONOS/QuickBird/GeoEye-1/WorldView-2 Imagery. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. [Online] 5 (1). Available from: https://ieeexplore.ieee.org/document/6125998 [Accessed: 13/09/21].

[178] Choi, J., Yu, K. and Kim, Y. (2010) A New Adaptive Component-Substitution-Based Satellite Image Fusion by Using Partial Replacement. *IEEE Transactions on Geoscience and Remote Sensing*. [Online] 49 (1). Available from: https://ieeexplore.ieee.org/document/5523973 [Accessed: 13/09/21].

[179] Ranchin, T. and Wald, L. (2000). Fusion of high spatial and spectral resolution images: The ARSIS concept and its implementation. *Photogrammetric Engineering and Remote Sensing*. [Online] 66. Available from: https://www.researchgate.net/publication/29612628_Fusion_of_high_spatial_and_spectral_resolution_images_The_ARSIS_concept_and_its_implementation [Accessed: 13/09/21].

[180] Yan, Wu., Ming, L. and Guisheng, L. (2004) Multiresolution wavelet decomposition to merge Landsat TM and SPOT panchromatic data. In: *Proceedings. ICCEA 2004. 2004 3rd International Conference on Computational Electromagnetics and Its Applications, November 2004*. New York: IEEE, pp. 521-524.

[181] Jr, P.S.C., Sides, S.C. and Anderson, J.A. (1991) Comparison of Three Different Methods to Merge Multiresolution and Multispectral Data: Landsat TM and SPOT Panchromatic. Photogrammetric Engineering and Remote Sensing. [Online] 57. Available from: https://www.researchgate.net/publication/200458993_Comparison_of_Three_Different_Methods_to_Merge_Multiresolution_and_Multispectral_Data_Landsat_TM_and_SPOT_Panchromatic [Accessed: 13/09/21].

[182] Nencini, F., Garzelli, A., Baronti, S. and Alparone, L. (2007) Remote sensing image fusion using the curvelet transform. *Information Fusion*. [Online] 8 (2). Available from: https://www.sciencedirect.com/science/article/pii/S1566253506000340 [Accessed: 13/09/21].

[183] Wald, L. and Ranchin, T. (2002) Liu 'Smoothing filter-based intensity modulation: A spectral preserve image fusion technique for improving spatial details'. *International Journal of Remote Sensing*. [Online] 23. Available from:

https://www.researchgate.net/publication/43236957_Liu_%27Smoothing_filter-based_intensity_modulation_A_spectral_preserve_image_fusion_technique_for_improving_spatial_details%27 [Accessed: 13/09/21].

[184] Khan, M., Chanussot, J., Condat, L. and Montanvert, A. (2008) Indusion: Fusion of Multispectral and Panchromatic Images Using the Induction Scaling Technique. *IEEE Geoscience and Remote Sensing Letters*. [Online] 5. Available from: https://www.semanticscholar.org/paper/Indusion%3A-Fusion-of-Multispectral-and-Panchromatic-Khan-Chanussot/c08c5595b09caf621a2b3d4b86eae34ab5972ed1 [Accessed: 13/09/21].

[185] Kaplan, G. and Avdan, U. (2018) Sentinel-2 Pan Sharpening – Comparative Analysis. [Online]. Available from: https://www.researchgate.net/publication/323933204_Sentinel-2_Pan_Sharpening_-_Comparative_Analysis [Accessed: 13/09/21].

[186] Vaiopoulos, A.D. and Karantzalos, K. (2016) PANSHARPENING ON THE NARROW VNIR AND SWIR SPECTRAL BANDS OF SENTINEL-2. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. [Online]. Available from: https://www.researchgate.net/publication/304337752_PANSHARPENING_ON_THE_NARROW_VNIR_AND_SWIR_SPECTRAL_BANDS_OF_SENTINEL-2 [Accessed: 13/09/21].

[187] Wang, Q., Shi, W., Li, Z. and Atkinson, P. (2016) Fusion of Sentinel-2 images. *Remote Sensing of Environment*. [Online] 187. Available from: https://www.researchgate.net/publication/309308481_Fusion_of_Sentinel-2_images [Accessed: 13/09/21].

[188] Park, H., Choi, J., Park, N., & Choi, S. (2017) Sharpening the VNIR and SWIR Bands of Sentinel-2A Imagery through Modified Selected and Synthesized Band Schemes. *Remote Sensing*. [Online] 9 (10). Available from: https://www.semanticscholar.org/paper/Sharpening-the-VNIR-and-SWIR-Bands-of-Sentinel-2A-Park-Choi/86d336f4a5bf8341856d307e6276c8bee7b9ea11 [Accessed: 13/09/21].

[189] Du, Y., Zhang, Y., Ling, F., Wang, Q., Li, W. and Li, X. (2016) Water Bodies' Mapping from Sentinel-2 Imagery with Modified Normalized Difference Water Index at 10-m Spatial Resolution Produced by Sharpening the SWIR Band. *Remote Sensing*. [Online] 354 (8). Available from: https://www.researchgate.net/publication/301565305_Water_Bodies%27_Mapping_from_Sentinel-2_Imagery_with_Modified_Normalized_Difference_Water_Index_at_10-m_Spatial_Resolution_Produced_by_Sharpening_the_SWIR_Band [Accessed: 13/09/21].

[190] Gasparovic, M. and Jogun, T. (2018) The effect of fusing Sentinel-2 bands on land-cover classification. *International Journal of Remote Sensing*. [Online] 39. Available from: https://www.researchgate.net/publication/320575826_The_effect_of_fusing_Sentinel-2_bands_on_land-cover_classification [Accessed: 13/09/21].

[191] Brodu, N. (2017) Super-Resolving Multiresolution Images With Band-Independent Geometry of Multispectral Pixels. *IEEE Transactions on Geoscience and Remote Sensing*. [Online] 55. Available from: https://www.semanticscholar.org/paper/Super-Resolving-Multiresolution-Images-With-of-Brodu/75dc0712f41ac708c5382cb6cb1d58d42bbf5043 [Accessed: 13/09/21].

[192] Lanaras, C., Bioucas-D, J., Baltsavias, E. and Schindler, K. (2017) Super-Resolution of Multispectral Multiresolution Images from a Single Sensor. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), July 2017*. New York: IEEE, pp. 1505-1513.

[193] Paris, C., Bioucas-D., J. and Bruzzone, L. (2017) A hierarchical approach to superresolution of multispectral images with different spatial resolutions. In: *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, July 2017. New York: IEEE, pp. 2589-2592.

[194] Ulfarsson, M.O., Palsson, F., Mura, M.D. and Sveinsson, J.R. (2019) Sentinel-2 Sharpening Using a Reduced-Rank Method. *IEEE Transactions on Geoscience and Remote Sensing*. [Online] 57 (9). Available from: https://ieeexplore.ieee.org/document/8694937 [Accessed: 13/09/21].

[195] Paris, C., Bioucas-D., J. and Bruzzone, L. (2019) A Novel Sharpening Approach for Superresolving Multiresolution Optical Images. *IEEE Transactions on Geoscience and Remote Sensing*. [Online] 57 (3). Available from: https://ieeexplore.ieee.org/document/8472286 [Accessed: 13/09/21].

[196] Lin, C. and Bioucas-D, J. (2020) An Explicit and Scene-Adapted Definition of Convex Self-Similarity Prior With Application to Unsupervised Sentinel-2 Super-Resolution. *IEEE Transactions on Geoscience and Remote Sensing*. [Online] 58 (5). Available from: https://ieeexplore.ieee.org/document/8931230 [Accessed: 13/09/21].

[197] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. (1998) Gradient-based learning applied to document recognition. *Proceedings of the IEEE*. [Online] 86 (11). Available from: https://ieeexplore.ieee.org/document/726791 [Accessed: 13/09/21].

[198] LeCun, Y. (1998) *The MNIST Database*. [Online] LeCun, Y. Available from: http://yann.lecun.com/exdb/mnist/ [Accessed: 13/09/21].

[199] Liebel, L. and Körner, M. (2016) Single-Image Super Resolution for Multispectral Remote Sensing Data Using Convolutional Neural Networks. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. [Online]. Available from: https://www.researchgate.net/publication/303899950_Single-Image_Super_Resolution_for_Multispectral_Remote_Sensing_Data_Using_Convolutional_Neural_Networks [Accessed: 13/09/21].

[200] Dong, C., Loy, C.C. and Tang, X. (2016) Accelerating the Super-Resolution Convolutional Neural Network. *ECCV*. [Online]. Available from: https://www.semanticscholar.org/paper/Accelerating-the-Super-Resolution-Convolutional-Dong-Loy/79da740db9006b2aa3e7b571d038ec895e323121 [Accessed: 13/09/21].

[201] Cheng, G., Matsune, A., Li, Q., Zhu, L., Zang, H. and Zhan, S. (2019) Encoder-Decoder Residual Network for Real Super-Resolution. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), June 2019*. New York: IEEE, pp. 2169-2178.

[202] Wang, H., Ma, C. and Shen, C. (2021) Kernel Agnostic Real-world Image Super-resolution. [Online]. Available from: https://www.semanticscholar.org/paper/Kernel-Agnostic-Real-world-Image-Super-resolution-Wang-Ma/43472b27b2b19fc82df32f47513963a8a21f11cc [Accessed: 13/09/21].

[203] Youm, G., Bae, S. and Kim, M. (2016) Image super-resolution based on convolution neural networks using multi-channel input. In: *2016 IEEE 12th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP), July 2016*. New York: IEEE, pp. 1-5.

[204] Montufar, G., Pascanu, R., Cho, K. and Bengio, Y. (2014) On the Number of Linear Regions of Deep Neural Networks. *NIPS*. [Online]. Available from: https://www.researchgate.net/publication/260126867_On_the_Number_of_Linear_Regions_of_Deep_Neural_Networks [Accessed: 13/09/21].

[205] Kim, J., Lee, J.K. and Lee, K.M. (2016) Accurate Image Super-Resolution Using Very Deep Convolutional Networks. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016*. New York: IEEE, 1646-1654.

[206] Simonyan, K., and Zisserman, A. (2015) Very Deep Convolutional Networks for Large-Scale Image Recognition. CoRR. [Online]. Available from: https://www.semanticscholar.org/paper/Very-Deep-Convolutional-Networks-for-Large-Scale-Simonyan-Zisserman/eb42cf88027de515750f230b23b1a057dc782108 [Accessed: 13/09/21].

[207] Panagiotopoulou, A., Grammatikopoulos, L., Charou, E., Bratsolis, E., Madamopoulos, N. and Petrogonas, J. (2020) Very Deep Super-Resolution of Remotely Sensed Images with Mean Square Error and Var-norm Estimators as Loss Functions. [Online]. Available from: https://www.semanticscholar.org/paper/Very-Deep-Super-Resolution-of-Remotely-Sensed-with-Panagiotopoulou-Grammatikopoulos/513b86d9d9258460e0debb83f8c224832e1d818b [Accessed: 13/09/21].

[208] Mao, X., Shen, C. and Yang, Y. (2016) Image Denoising Using Very Deep Fully Convolutional Encoder-Decoder Networks with Symmetric Skip Connections. [Online]. Available from: https://www.semanticscholar.org/paper/Image-Denoising-Using-Very-Deep-Fully-Convolutional-Mao-Shen/d9330eea16c5af3ae32040af1b6eb3265f28deda [Accessed: 13/09/21].

[209] He, K., Zhang, X., Ren, S. and Sun, J. (2016) Deep Residual Learning for Image Recognition. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016*. New York: IEEE, pp. 770-778.

[210] Ledig, C., Theis, L., Huszár, F., Caballero, J., Aitken, A.P., Tejani, A., Totz, J., Wang, Z. and Shi, W. (2017) Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017*. New York: IEEE, 105-114.

[211] Lim, B., Son, S., Kim, H., Nah, S. and Lee, K.M. (2017) Enhanced Deep Residual Networks for Single Image Super-Resolution. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), July 2017*. New York: IEEE, pp. 1132-1140.

[212] Tuna, C., Unal, G. and Sertel, E. (2018) Single-frame super resolution of remote-sensing images by convolutional neural networks. *International Journal of Remote Sensing*. [Online] 39 (8). Available from: https://www.tandfonline.com/doi/abs/10.1080/01431161.2018.1425561 [Accessed: 13/09/21].

[213] Calm, O.E. (2020) Super-resolution of Sentinel-2 images a single network for all-bands up-sampling, thesis. [Online]. Available from: https://upcommons.upc.edu/handle/2117/329711 [Accessed: 13/09/21].

[214] Moshkov, N., Mathe, B., Kertész-Farkas, A., Hollandi, R. and Horváth, P. (2019) Test-time augmentation for deep learning-based cell segmentation on microscopy images. [Online]. Available from: https://www.semanticscholar.org/paper/Test-time-augmentation-for-deep-learning-based-cell-Moshkov-Mathe/084d3d1f2e8284f75aa6a5d4b550ba191ad12084 [Accessed: 13/09/21].

[215] Wei, P., Lu, H., Timofte, R., Lin, L., Zuo, W., Pan, Z., Li, B., Xi, T., Fan, Y., Zhang, G., Liu, J., Han, J., Ding, E., Xie, T., Cao, L., Zou, Y., Shen, Y., Zhang, J., Jia, Y., Debeir, O. (2020) AIM 2020 Challenge on Real Image Super-Resolution: Methods and Results. [Online]. Available from: https://www.researchgate.net/publication/344400340_AIM_2020_Challenge_on_Real_Image_Super-Resolution_Methods_and_Results [Accessed: 13/09/21].

[216] Shocher, A., Cohen, N. and Irani, M. (2018) Zero-Shot Super-Resolution Using Deep Internal Learning. [Online]. Available from: https://www.researchgate.net/publication/329748512_Zero-Shot_Super-Resolution_Using_Deep_Internal_Learning [Accessed: 13/09/21].

[217] Du, C., Zewei, H., Anshun, S., Jiangxin, Y., Yanlong, C., Yanpeng, C., Siliang, T. and Yang, M.Y. (2019) Orientation-Aware Deep Neural Network for Real Image Super-Resolution. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), June 2019*. New York: IEEE, pp. 1944-1953.

[218] Li, C., He, D., Liu, X., Ding, Y. and Wen, S. (2019) Adapting Image Super-Resolution State-Of-The-Arts and Learning Multi-Model Ensemble for Video Super-Resolution. In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), June 2019*. New York: IEEE, 2033-2040.

[219] Perez, F., Vasconcelos, C., Avila, S. and Valle, E. (2018) Data Augmentation for Skin Lesion Analysis. [Online]. Available from: https://www.semanticscholar.org/paper/Data-Augmentation-for-Skin-Lesion-Analysis-Perez-Vasconcelos/d5f0f8544ed011cca4a0865c48c88c4b5304cc57 [Accessed: 13/09/21].

[220] Matsunaga, K., Hamada, A., Minagawa, A. and Koga, H. (2017) Image Classification of Melanoma, Nevus and Seborrheic Keratosis by Deep Neural Network Ensemble. [Online]. Available from: https://www.semanticscholar.org/paper/Image-Classification-of-Melanoma%2C-Nevus-and-by-Deep-Matsunaga-Hamada/34eec4cb7b71b8533b2fe1e34c65786b0c7c0e4b [Accessed: 13/09/21].

[221] Radosavovic, I., Dollár, P., Girshick, R.B., Gkioxari, G. and He, K. (2018) Data Distillation: Towards Omni-Supervised Learning. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, June 2018*. New York: IEEE, 4119-4128.

[222] Pang, T., Xu, K. and Zhu, J. (2020) Mixup Inference: Better Exploiting Mixup to Defend Adversarial Attacks. [Online]. Available from: https://www.semanticscholar.org/paper/Mixup-Inference%3A-Better-Exploiting-Mixup-to-Defend-Pang-Xu/e1b9d5d4cf17f5ceaf2d6798ffd594fd1d017eaa [Accessed: 13/09/21].

[223] Wang, G., Li, W., Aertsen, M., Deprest, J., Ourselin, S. and Vercauteren, T.K. (2019) Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks. *Neurocomputing*. [Online] 335. Available from: https://www.semanticscholar.org/paper/Aleatoric-uncertainty-estimation-with-test-time-for-Wang-Li/41c9bfb05a7c3eeafe3f749eae1acf6afd873194 [Accessed: 13/09/21].

[224] The European Space Agency (2019) Historical Satellite Images: Get Insights From The Past. [Weblog] *The European Space Agency*. 3rd May. Available from: https://sentinel.esa.int/web/sentinel/missions/sentinel-2 [Accessed: 25/08/21].

[225] Frampton, W.J., Dash, J., Watmough, G. and Milton, E.J. (2013) Evaluating the capabilities of Sentinel-2 for quantitative estimation of biophysical variables in vegetation. *ISPRS Journal of Photogrammetry and Remote Sensing*. [Online] 82. Available from: https://www.sciencedirect.com/science/article/pii/S092427161300107X [Accessed: 13/09/21].

[226] The European Space Agency (2021) *Copernicus Open Access Hub*. [Online] The European Space Agency. Available from: https://scihub.copernicus.eu/dhus/#/home [Accessed: 13/09/21].

[227] Python (2021) *What is Python? Executive Summary*. [Online] Python. Available from: https://www.python.org/doc/essays/blurb/ [Accessed: 13/09/21].

[228] Microsoft (2021) *Welcome to the Visual Studio IDE*. [Online] Microsoft. Available from: https://docs.microsoft.com/en-us/visualstudio/get-started/visual-studio-ide?view=vs-2019 [Accessed: 13/09/21].

[229] Microsoft (2019) *Work with Python in Visual Studio on Windows*. [Online] Microsoft. Available from: https://docs.microsoft.com/en-us/visualstudio/python/overview-of-python-tools-for-visual-studio?view=vs-2019 [Accessed: 13/09/21].

[230] Google (2019) *Colaboratory*. [Online] Google. Available from: https://research.google.com/colaboratory/faq.html [Accessed: 13/09/21].

[231] Imageio (2020) *Welcome to imageio's documentation!* [Online] Imageio. Available from: https://imageio.readthedocs.io/en/stable/ [Accessed: 13/09/21].

[232] OpenCV (2021) *About*. [Online] OpenCV. Available from: https://opencv.org/about/ [Accessed: 13/09/21].

[234] OpenCV (2021) *Geometric Transformations of Images*. [Online] OpenCV. Available from: https://docs.opencv.org/master/da/d6e/tutorial_py_geometric_transformations.html [Accessed: 13/09/21].

[235] Scikit-image (2021) *Image processing in Python*. [Online] Scikit-image. Available from: https://scikit-image.org/ [Accessed: 13/09/21].

[236] Scikit-image (2021) *Image processing for Python*. [Online] Scikit-image. Available from: https://scikit-image.org/docs/dev/api/skimage.html [Accessed: 13/09/21].

[237] Khalel, A. (2021) *All image quality metrics you need in one package*. [Online] Pypi. Available from: https://pypi.org/project/sewar/ [Accessed: 13/09/21].

[238] Conda (2017) *Getting started with conda*. [Online] Conda. Available from: https://conda.io/projects/conda/en/latest/user-guide/getting-started.html [Accessed: 13/09/21].

[239] QGIS (2021) *A Free and Open Source Geographic Information System*. [Online] QGIS. Available from: https://www.qgis.org/en/site/ [Accessed: 13/09/21].

[240] GitHub (2021) *DSen2*. [Online] GitHub. Available from: https://github.com/lanha/DSen2 [Accessed: 13/09/21].

[241] Pip (2021) *pip*. [Online] pip. Available from: https://pip.pypa.io/en/stable/ [Accessed: 13/09/21].

[242] NVIDIA (2021) *CUDA Toolkit Archive*. [Online] NVIDIA. Available from: https://developer.nvidia.com/cuda-toolkit-archive [Accessed: 13/09/21].

[243] TheAILearner (2021) *Contrast Stretching*. [Online] TheAILeaner. Available from: https://theailearner.com/tag/min-max-stretching/ [Accessed: 13/09/21].

[244] Jensen, J.R. and Schill, S.R. (2005) Contrast Enhancement. [Online]. Available from: http://knightlab.org/rscc/legacy/RSCC_Contrast_Enhancement.pdf [Accessed: 13/09/21].

[245] Wang, J., Huang, B., Zhang, H.K. and Ma, P. (2019) Sentinel-2A Image Fusion Using a Machine Learning Approach. In: *IEEE Transactions on Geoscience and Remote Sensing*. [Online] 57 (12). Available from: https://ieeexplore.ieee.org/abstract/document/8809414 [Accessed: 13/09/21].

[246] NumPy (2021) *The fundamental package for scientific computing with Python*. [Online] NumPy. Available from: https://numpy.org/ [Accessed: 14/09/21].

[247] NumPy (2021) *Statistics*. [Online] NumPy. Available from: https://numpy.org/doc/stable/reference/routines.statistics.html [Accessed: 15/09/21].

[248] MathWorks (2021) *Image Quality Metrics*. [Online] MathWorks. Available from: https://uk.mathworks.com/help/images/image-quality-metrics.html [Accessed: 15/09/21].

[249] Asamoah, D., Oppong, E., Oppong, S. and Danso, J. (2018) Measuring the Performance of Image Contrast Enhancement Technique. *International Journal of Computer Applications*. [Online] 181 (22). Available from: https://www.researchgate.net/publication/328358793_Measuring_the_Performance_of_Image_Contrast_Enhancement_Technique [Accessed: 15/09/21].

[250] Wang, Z. and Bovik, A.C. (2002) A universal image quality index. *IEEE Signal Processing Letters*. [Online] 9 (3). Available from: https://ieeexplore.ieee.org/document/995823 [Accessed: 15/09/21].

Appendices

Appendix A:

DE MONTFORT UNIVERSITY LEICESTER

For official use
Tracking No:
Date approved:
Initials:

Faculty of Computing, Engineering and Media (CEM)
Application to Gain Ethical Approval
Taught Masters' Degree Student

NOTE: If your research involves using human tissue or fluid samples or animals please DO NOT use this form. You should seek guidance from the Chair of the Faculty Research Ethics Committee (FREC) before starting your project.

All Taught Masters that include a research project or dissertation require ethical approval. Students must complete this form and discuss the likely outcome with their project supervisor. There are four possible outcomes:

1. No interaction with human beings is planned and no identifiable data on or from individuals is used.
2. Students interview individuals, carry out surveys, observe, and participate with adults who understand the research and are aware they can withdraw their participation at any time. Supervisors must ensure that the appropriate boxes in section 2 are ticked and that the student knows how to address the ethical concerns.

   For projects which fall under outcome 1 or 2, this ethical application form should be signed by the student and the project supervisor. Nothing further is required; the form does not need to go to the FREC.
3. The research is with vulnerable people who may not understand the research and their role (eg children, hospital patients, people with mental health issues, subordinates in power relationships, etc). This also applies to research into illegal activities or research that could produce a risk of injury to anybody. The student / researcher must find ways to address these problems and the supervisor must be confident that these have been addressed satisfactorily.

   For projects which fall under outcome 3, the ethical application form should be signed by the student and project supervisor and a copy of the review form sent to the FREC (via amsmith@dmu.ac.uk). Once the FREC accepts the review form, the student and supervisor will be notified and the student may start work on the project.
4. The research is ethically problematic.

   For projects which fall under outcome 4, this ethical application form should be signed by the student and project supervisor and a copy of the review form submitted to the FREC (via amsmith@dmu.ac.uk) for resolution. Once resolved, the student and supervisor will be notified and the student may start work on the project.

All outcomes

Once approved, the form should be submitted by the student to the relevant Blackboard Dissertation shell. A copy of the form and, *where relevant*, the following supporting documents, must be included in the project report (dissertation) as appendices when it is submitted for assessment.

Supporting documents (may apply to outcome 2, 3 or 4):

- Information that will be provided to the study participants
- Participant consent form
- Other documentation as advised by the supervisory team

1. Applicant			
Surname	Hubble	First Name	Adam
DMU Email Address	p17175774@my365.dmu.ac.uk	Student ID Number	P17175774
Working title of the proposed investigation: *(Abbreviations must not be used)*: Single-Image Super-Resolution: Towards the Enhancement of Sentinel-2 Satellite Imagery			

2. Delete 'Yes' or 'No as appropriate in table below. If you answer any of the following questions with 'Yes', then specific ethical issues WILL be raised that MUST be addressed. You will need to explain in detail in section 3 how you will address these ethical issues, and consult your supervisor.

Has your research proposal identified any of the following research activities?

Gathering information from or/and about human beings through: interviewing, surveying, questionnaires, observation of human behaviour	~~Yes~~ / No
Using archived data in which individuals are identifiable	~~Yes~~ / No
Researching into illegal activities, or activities at the margins of the law	~~Yes~~ / No
Researching into activities that have a risk of personal injury anybody.	~~Yes~~ / No
Research that might impact on human behaviour, for example on autonomous vehicles.	~~Yes~~ / No
Researching topics that are concerned with the following 'sensitive research' areas: access to web sites normally prohibited on university servers, or extremism and radicalisation, criminal activities, etc.	~~Yes~~ / No

For more information about whether your research should be classified as sensitive see: http://www.dmu.ac.uk/research/ethics-and-governance/sensitive-research.aspx).

Are there additional factors that could give rise to ethical concerns eg communication difficulties?

No additional factors of the research proposed could give rise to ethical concerns.

3. How will the issues you have raised in response to questions 2 be addressed?

No issues are realised, concerning the subject area of the research proposed.

Note: you should consider the following:

- *Providing participants with the full details of the objectives of the research*
- *Providing information appropriate for those whose first language is not English*
- *Voluntary participation with informed consent (through the provision of a consent form)*
- *Written description of involvement*
- *Freedom to withdraw*
- *Keeping appropriate records*
- *Signed acknowledgement and understanding by participants*
- *Relevant codes of conduct / guidelines*

4. To which ethical codes of conduct have you referred?

Codes referred to, are of the following:

- GDPR Guidance for Researchers
- Guidelines for Good Research Practice
- Misconduct in Research - Investigation Procedure
- Open Access (DORA)

Note: For the CEM Faculty these codes typically include those published by the BCS, ACM, IEEE or other applicable codes such as the code of the Social Research Association or specific funding bodies, such as the ESRC. Links to some of these codes are available on the CEM Faculty FHREC website. http://www.dmu.ac.uk/research/ethics-and-governance/dmu-policies-and-external-requirements-.aspx

Please note, if the methodology changes in relation to ethical considerations after submission, you can submit a new form, following the same procedure.

AUTHORISATION

Signature of Applicant			
Signed	Adam Leonard Hubble	Date	24/06/21

Approval signature of Supervisor			
Signed	Dr. Linika Deka	Date	30/06/2021

Outcome [circle number] (1 ~~2 3 4~~)

Name of Supervisor ____________________

Where necessary, **authorising signature (FREC Chair)**			
Signed		Date	

Outcome [circle number] (1 2 3 4)

Name of FREC chair ____________________

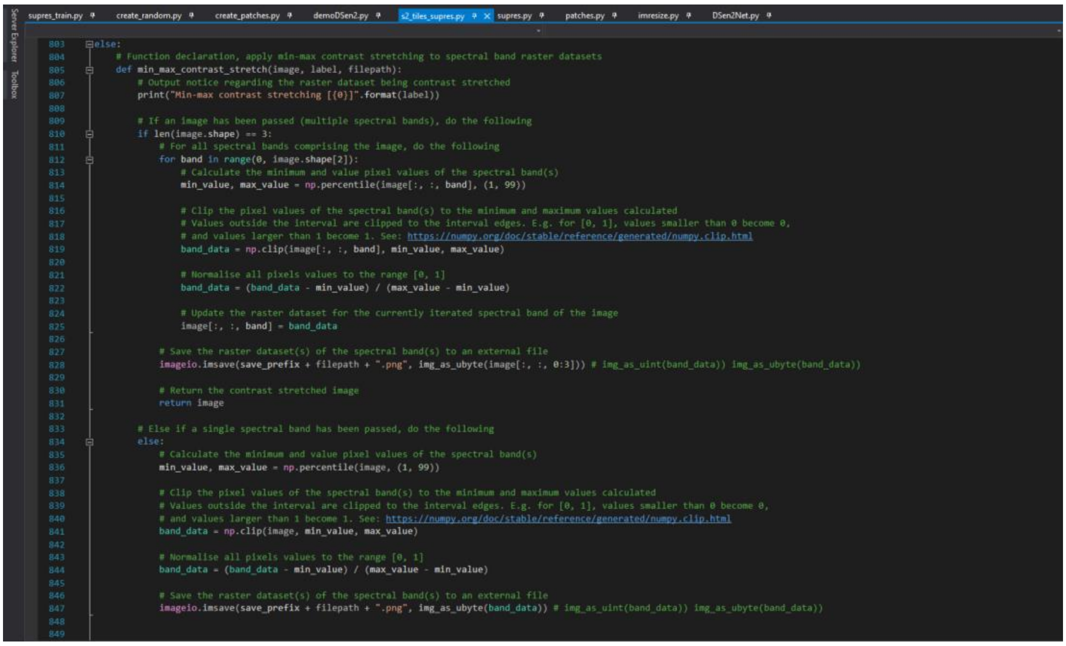
Conditions

Where necessary, **full approval - authorising signature (FREC Chair)**			
Signed	____________________	Date	____________

NOTES FOR GUIDANCE:

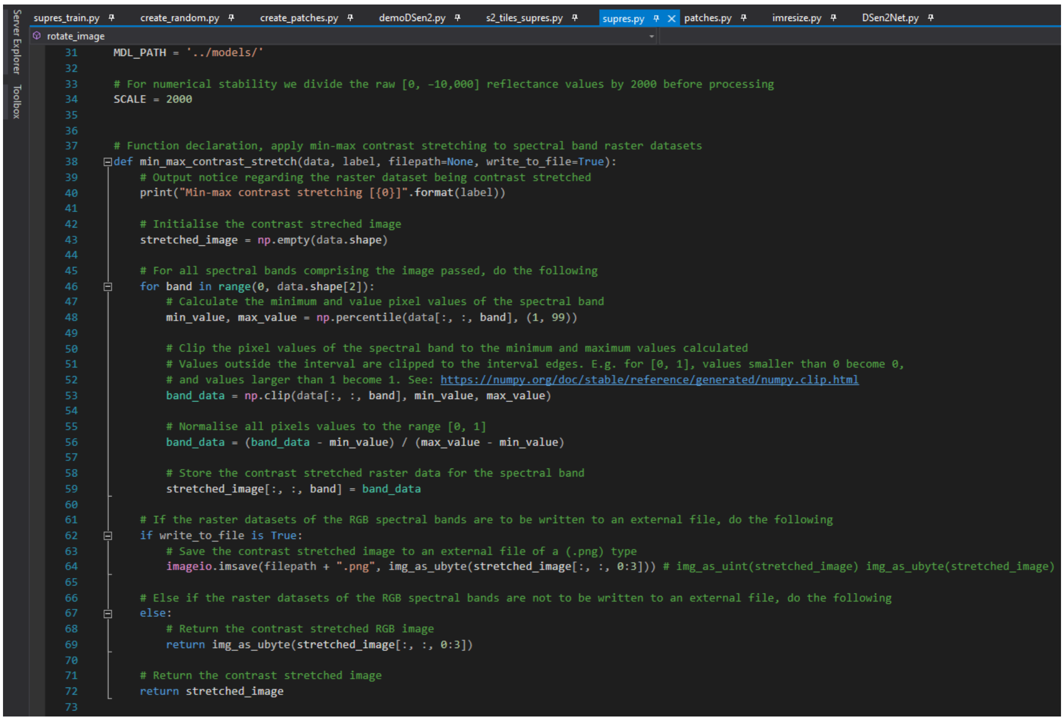
1. Participants cooperation in a research project is entirely voluntary at all stages. They must not be misled when being asked for co-operation.
2. Participant anonymity must be strictly preserved. If the Participant, on request from the Researcher, has given permission for data to be passed on in a form which allows that Participant to be personally identified:
   (a) the Participant must first have been told to whom the information would be supplied and the purpose for which it will be used
   (b) the Researcher must ensure that the information will not be used for any non-research purpose and that the recipient of the information has agreed to conform to the requirements of any relevant Code of Practice.
3. The Researcher must take all reasonable precautions to ensure that the Participant is in no way directly harmed or adversely affected as a result of their involvement in a research project.
4. The Researcher must take special care when interviewing vulnerable people – for example children or the elderly. The Faculty ethics representative will give advice on gaining consent for studies involving vulnerable people.
5. Participants must be told (normally at the beginning of the interview) if observation techniques and/or recording equipment are used, except where these are used in a public place. If a respondent so wishes, the record or relevant section of it must be destroyed or deleted. Participant anonymity must not be infringed by the use of such methods.
6. Participants must be enabled to check, without difficulty, the identity and bona fides of the Researcher.
7. Researchers should ensure data is held securely, and follow university and legal requirements regarding data protection and data storage.

Appendix B:



```python
else:
    # Function declaration, apply min-max contrast stretching to spectral band raster datasets
    def min_max_contrast_stretch(image, label, filepath):
        # Output notice regarding the raster dataset being contrast stretched
        print("Min-max contrast stretching [{0}]".format(label))

        # If an image has been passed (multiple spectral bands), do the following
        if len(image.shape) == 3:
            # For all spectral bands comprising the image, do the following
            for band in range(0, image.shape[2]):
                # Calculate the minimum and value pixel values of the spectral band(s)
                min_value, max_value = np.percentile(image[:, :, band], (1, 99))

                # Clip the pixel values of the spectral band(s) to the minimum and maximum values calculated
                # Values outside the interval are clipped to the interval edges. E.g. for [0, 1], values smaller than 0 become 0,
                # and values larger than 1 become 1. See: https://numpy.org/doc/stable/reference/generated/numpy.clip.html
                band_data = np.clip(image[:, :, band], min_value, max_value)

                # Normalise all pixels values to the range [0, 1]
                band_data = (band_data - min_value) / (max_value - min_value)

                # Update the raster dataset for the currently iterated spectral band of the image
                image[:, :, band] = band_data

            # Save the raster dataset(s) of the spectral band(s) to an external file
            imageio.imsave(save_prefix + filepath + ".png", img_as_ubyte(image[:, :, 0:3])) # img_as_uint(band_data)) img_as_ubyte(band_data))

            # Return the contrast stretched image
            return image

        # Else if a single spectral band has been passed, do the following
        else:
            # Calculate the minimum and value pixel values of the spectral band(s)
            min_value, max_value = np.percentile(image, (1, 99))

            # Clip the pixel values of the spectral band(s) to the minimum and maximum values calculated
            # Values outside the interval are clipped to the interval edges. E.g. for [0, 1], values smaller than 0 become 0,
            # and values larger than 1 become 1. See: https://numpy.org/doc/stable/reference/generated/numpy.clip.html
            band_data = np.clip(image, min_value, max_value)

            # Normalise all pixels values to the range [0, 1]
            band_data = (band_data - min_value) / (max_value - min_value)

            # Save the raster dataset(s) of the spectral band(s) to an external file
            imageio.imsave(save_prefix + filepath + ".png", img_as_ubyte(band_data)) # img_as_uint(band_data)) img_as_ubyte(band_data))
```

Figure 33: Depiction of the min-max contrast stretching method implemented within the 's2_tiles_supres.py' Python script file, devoted to both single-band and multi-band image contrast stretching.



```python
MDL_PATH = '../models/'

# For numerical stability we divide the raw [0, -10,000] reflectance values by 2000 before processing
SCALE = 2000


# Function declaration, apply min-max contrast stretching to spectral band raster datasets
def min_max_contrast_stretch(data, label, filepath=None, write_to_file=True):
    # Output notice regarding the raster dataset being contrast stretched
    print("Min-max contrast stretching [{0}]".format(label))

    # Initialise the contrast streched image
    stretched_image = np.empty(data.shape)

    # For all spectral bands comprising the image passed, do the following
    for band in range(0, data.shape[2]):
        # Calculate the minimum and value pixel values of the spectral band
        min_value, max_value = np.percentile(data[:, :, band], (1, 99))

        # Clip the pixel values of the spectral band to the minimum and maximum values calculated
        # Values outside the interval are clipped to the interval edges. E.g. for [0, 1], values smaller than 0 become 0,
        # and values larger than 1 become 1. See: https://numpy.org/doc/stable/reference/generated/numpy.clip.html
        band_data = np.clip(data[:, :, band], min_value, max_value)

        # Normalise all pixels values to the range [0, 1]
        band_data = (band_data - min_value) / (max_value - min_value)

        # Store the contrast stretched raster data for the spectral band
        stretched_image[:, :, band] = band_data

    # If the raster datasets of the RGB spectral bands are to be written to an external file, do the following
    if write_to_file is True:
        # Save the contrast stretched image to an external file of a (.png) type
        imageio.imsave(filepath + ".png", img_as_ubyte(stretched_image[:, :, 0:3])) # img_as_uint(stretched_image) img_as_ubyte(stretched_image)

    # Else if the raster datasets of the RGB spectral bands are not to be written to an external file, do the following
    else:
        # Return the contrast stretched RGB image
        return img_as_ubyte(stretched_image[:, :, 0:3])

    # Return the contrast stretched image
    return stretched_image
```

Figure 34: Depiction of the min-max contrast stretching method implemented within the 'supres.py' Python script file, devoted to multi-band image contrast stretching only.

Appendix C:

```python
    result_dataset_sr = None
    result_dataset_gt = None
    result_dataset_lr = None


# Initialise an array representing the descriptions of all the spectral bands in the raster dataset
all_descriptions = []


# Function declaration, write the raster data of a given spectral band to an external file
def write_band_data(data, result_dataset, band_index, description, shortname=None):
    # Set the spectral band index and desciption variables to be globally accessible
    global all_descriptions

    # Append the description of the spectral band to the array of descriptions
    all_descriptions += [description]

    # If the output file format of the super-resolved spectral band raster data is of the (.npz) type, do the following
    # (.npz) file format is a zipped archive of files named after the variables they contain. The archive is not compressed
    # and each file in the archive contains one variable in (.npy) format. See: https://numpy.org/doc/stable/reference/generated/numpy.savez.html
    if output_file_format == "npz":
        # Allocate the raster dataset of the spectral band to the associated description (name) in the dictionary
        spectral_bands[description] = data

    # Else if the output file format of the super-resolved spectral band raster data is not of the (.npz) type, do the following
    else:
        # Store the description of the spectral band into the result dataset variable
        result_dataset.GetRasterBand(band_index + 1).SetDescription(description)
        # Store the raster dataset of the spectral band into the result dataset variable
        result_dataset.GetRasterBand(band_index + 1).WriteArray(data)

```

Figure 35: Depiction of the 'write_band_data' method implemented within the 's2_tiles_supres.py' Python script file, devoted to generating GeoTIFF representations of Sentinel-2 spectral band composites.

```python
result_dataset_gt.SetProjection(dataset_20m.GetProjection())
#endregion

# Write and output the spectral band raster datasets to an external file
#region
# If the original 10m resolution spectral bands are to be copied into the output file in addition to the super-resolved spectral bands, do the following
if copy_original_bands:
    # Output notice regarding the inclusion of the original 10m resolution spectral bands in the data file
    sys.stdout.write(" the original 10m bands and")

    # For each spectral band comprising the validated 10m resolution spectral band array, do the following
    for band_index, band in enumerate(validated_10m_bands):
        # Write the original 10m resolution spectral band raster datasets to the 10m resolution spectral band raster dataset array, ready to be written to an external file
        write_band_data(data10[:, :, band_index], validated_descriptions[band])

# Output notice regarding the directory of the super-resolved spectral band data file
print("Writing the super-resolved bands in %s" % output_file)
# Output notice regarding the directory of the ground truth spectral band data file
print("Writing the ground truth bands in %s" % 'C:\DSen2\DSen2-master\data\output\georeferenced\ground-truth.tif')
# Output notice regarding the directory of the low-resolution spectral band data file
print("Writing the low-resolution bands in %s" % 'C:\DSen2\DSen2-master\data\output\georeferenced\low-resolution.tif')

# For each spectral band in the validated 20m resolution spectral bands array, do the following
for band_index, band in enumerate(validated_20m_bands):
    # Write the spectral band raster dataset to an external file
    write_band_data(data20_gt[:, :, band_index], result_dataset_gt, band_index, "GT" + validated_descriptions[band], "GT" + band)
    # Write the spectral band raster dataset to an external file
    write_band_data(data20_cubic[:, :, band_index], result_dataset_lr, band_index, "LR" + validated_descriptions[band], "LR" + band)
    # Write the spectral band raster dataset to an external file
    write_band_data(super_resolved_bands[:, :, band_index], result_dataset_sr, band_index, "SR" + validated_descriptions[band], "SR" + band)

# For all spectral band descriptions in the description array, do the following
for description in all_descriptions:
    # Output the description of the spectral band
    print(description)

# If the output file format of the super-resolved spectral band raster data is of the (.npz) type, do the following
# (.npz) file format is a zipped archive of files named after the variables they contain. The archive is not compressed
# and each file in the archive contains one variable in (.npy) format. See: https://numpy.org/doc/stable/reference/generated/numpy.savez.html
if output_file_format == "npz":
    # Save the output file as a uncompressed (.npz) format comprising the spectral band raster datasets
    # Each array will be saved to the output file with its corresponding keyword name. See: https://numpy.org/doc/stable/reference/generated/numpy.savez.html
    np.savez(output_file, bands=spectral_bands)
#endregion
```

Figure 36: Depiction of the extended use-case of the 'write_band_data' method, located within the 's2_tiles_supres.py' Python script file. Generating three separate GeoTIFF representations of the ground truth, bicubically interpolated, and super-resolved Sentinel-2 spectral band composites.

Appendix D:

```python
def splitTrainVal(train_path, train, label):...


def OpenDataFiles(path, run_60, SCALE):...


def OpenDataFilesTest(path, run_60, SCALE, true_scale=False):...


def downPixelAggr(image, SCALE=2):
    from scipy import signal
    import skimage.measure
    from scipy.ndimage.filters import gaussian_filter

    # If the image passed has no channel data (spectral bands), do the following
    if len(image.shape) == 2:
        # Expand the number of dimensions used to represent the image (support spectral band indexing)
        image = np.expand_dims(image, axis=-1)

    # Initialise the blurred image
    image_blur = np.zeros(image.shape)

    # For each spectral band comprising the image, do the following
    for i in range(0, image.shape[2]):
        # Filter the image with a gaussian filter
        image_blur[:, :, i] = gaussian_filter(image[:, :, i], 1 / SCALE)

    # Calculate the dimensions of the donwscaled image
    new_dimensions = tuple(axis_dimensions // SCALE for axis_dimensions in image.shape)

    # Initialise the low-resolution variant of the image
    image_lr = np.zeros(new_dimensions[0:2] + (image.shape[-1], ))

    # Iterate through all the image channels with average pooling (pixel aggregation)
    for i in range(0, image.shape[2]):
        # Downsample the image by applying an average pooling operation (mean) over each (scale x scale) block of pixels
        image_lr[:, :, i] = skimage.measure.block_reduce(image_blur[:, :, i], block_size=(SCALE, SCALE), func=np.mean)

    # Return the donwsampled image (remove all axes with a length of one - error handling)
    return np.squeeze(image_lr)
```

Figure 37: Depiction of the 'downPixelAggr' method implemented within the 'patches.py' Python script file, devoted to generating down-sampled renditions of Sentinel-2 image tiles.

Appendix E:

```python

# Store the ground truth raster dataset for the 10m resolution spectral bands
data10_gt = data10

# For 80->40m upsampling, the ground truth 20m resolution raster dataset requires to be downsampled to the 40m resolution
# Store the ground truth raster dataset for the 20m resolution spectral bands
#data20_gt = data20.astype(np.float64)      # 20m resolution
data20_gt = downPixelAggr(data20, SCALE=2) # 40m resolution

# Initialise the downsampled raster dataset variables, for the 10m, 20m and 60m resolution spectral bands
data10_lr = None
data20_lr = None
data60_lr = None

# If the 60m and 20m resolution spectral band images are to be super-resolved by the DSen2 model, do the following
if run_60:
    # Store the ground truth raster dataset for the 60m resolution spectral bands
    data60_gt = data60

    # Downsample the 60m, 20m and 10m resolution raster datasets by the 60m resolution scale factor configured
    # Scale factor of '6' by default, for 360->60m upsampling
    data10_lr = downPixelAggr(data10_gt, SCALE=downsample_scale_60m)
    data20_lr = downPixelAggr(data20_gt, SCALE=downsample_scale_60m)
    data60_lr = downPixelAggr(data60_gt, SCALE=downsample_scale_60m)

# Else if only the 20m resolution spectral band images are to be super-resolved by the DSen2 model, do the following
else:
    # Downsample the 20m and 10m resolution raster datasets by the 20m resolution scale factor configured
    # Scale factor of '2' by default, for 40->20m upsampling
    # Scale factor of '4' by default, for 80->40m upsampling

    # For 80->40m upsampling, the 10m resolution raster dataset requires to be downsampled to the 40m resolution
    data10_lr = downPixelAggr(data10_gt, SCALE=downsample_scale_20m)
    # For 80->40m upsampling, the 20m resolution raster dataset requires to be downsampled to the 80m resolution
    data20_lr = downPixelAggr(data20, SCALE=downsample_scale_20m)

```

Figure 38: Depiction of the extended use-case of the 'downPixelAggr' method, located in the 'patches.py' Python script file. Generating down-sampled renditions of the ground truth, bicubically interpolated, and super-resolved Sentinel-2 image tiles.

Appendix F:

```python
SCALE = 2000


# Function declaration, apply min-max contrast stretching to spectral band raster datasets
def min_max_contrast_stretch(data, label, filepath=None, write_to_file=True):...


# Function declaration, rotate the locations of pixels comprising an image
def rotate_image(image, constraint, tiled=True):
    # Initialise the transformed image
    transformed_image = np.empty(image.shape)

    # If the constraint is of a transformation type (super-resolve the image with transformations), do the following
    if constraint is not None:
        # If the image has been tiled, do the following
        if tiled is True:
            # For all tiles comprising the spectral band images, do the following
            for tile in range(image.shape[0]):
                # For all spectral band images, do the following
                for band in range(image.shape[1]):
                    # Rotate the currently iterated tile comprising the spectral band image
                    transformed_image[tile, band] = cv2.rotate(image[tile, band], constraint)

        # Else if the image has not been tiled, do the following
        else:
            # For all spectral band images, do the following
            for band in range(image.shape[2]):
                # Rotate the spectral band image
                transformed_image[:, :, band] = cv2.rotate(image[:, :, band], constraint)

    # Else if the constraint is 'none' (super-resolve the image with no transformation), do the following
    else:
        # Set the transformed image to the image passed
        transformed_image = image

    # Return the transformed image
    return transformed_image

```

Figure 39: Depiction of the 'rotate_image' method implemented within the 'supres.py' Python script file, devoted to generating rotary offset transformations of Sentinel-2 image tiles.

```python

# Function declaration, rotate the locations of pixels comprising an image
def rotate_image(image, constraint, tiled=True):...


# Function declaration, invert the locations of pixels comprising an image
def invert_image(image, constraint, tiled=True):
    # Initialise the transformed image
    transformed_image = np.empty(image.shape)

    # If the constraint is of a transformation type (super-resolve the image with transformations), do the following
    if constraint is not None:
        # If the image has been tiled, do the following
        if tiled is True:
            # For all tiles comprising the spectral band images, do the following
            for tile in range(image.shape[0]):
                # For all spectral band images, do the following
                for band in range(image.shape[1]):
                    # Flip the currently iterated tile comprising the spectral band image
                    transformed_image[tile, band] = cv2.flip(image[tile, band], constraint)

        # Else if the image has not been tiled, do the following
        else:
            # For all spectral band images, do the following
            for band in range(image.shape[2]):
                # Flip the spectral band image
                transformed_image[:, :, band] = cv2.flip(image[:, :, band], constraint)

    # Else if the constraint is 'none' (super-resolve the image with no transformation), do the following
    else:
        # Set the transformed image to the image passed
        transformed_image = image

    # Return the transformed image
    return transformed_image

```

Figure 40: Depiction of the 'invert_image' method implemented within the 'supres.py' Python script file, devoted to generating inverted transformations of Sentinel-2 image tiles.

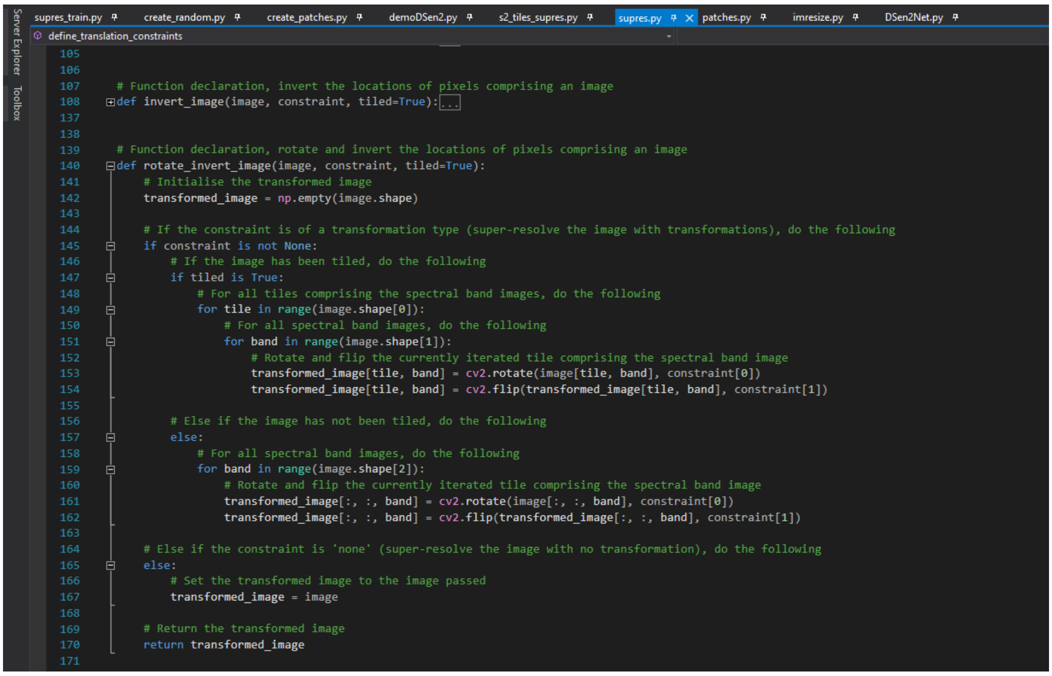


Figure 41: Depiction of the 'rotate_invert_image' method implemented within the 'supres.py' Python script file, devoted to generating rotary offset and inverted transformations of Sentinel-2 image tiles.

Appendix G:

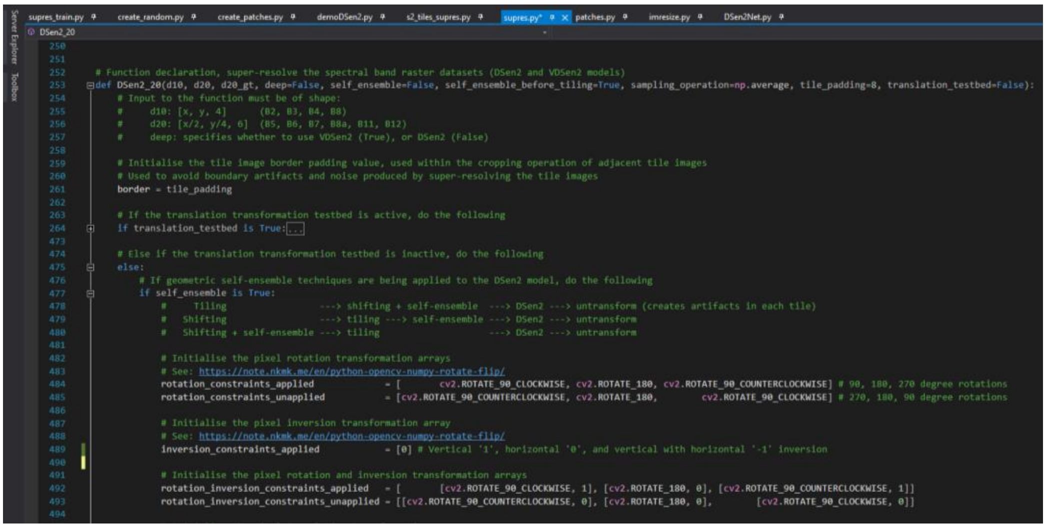
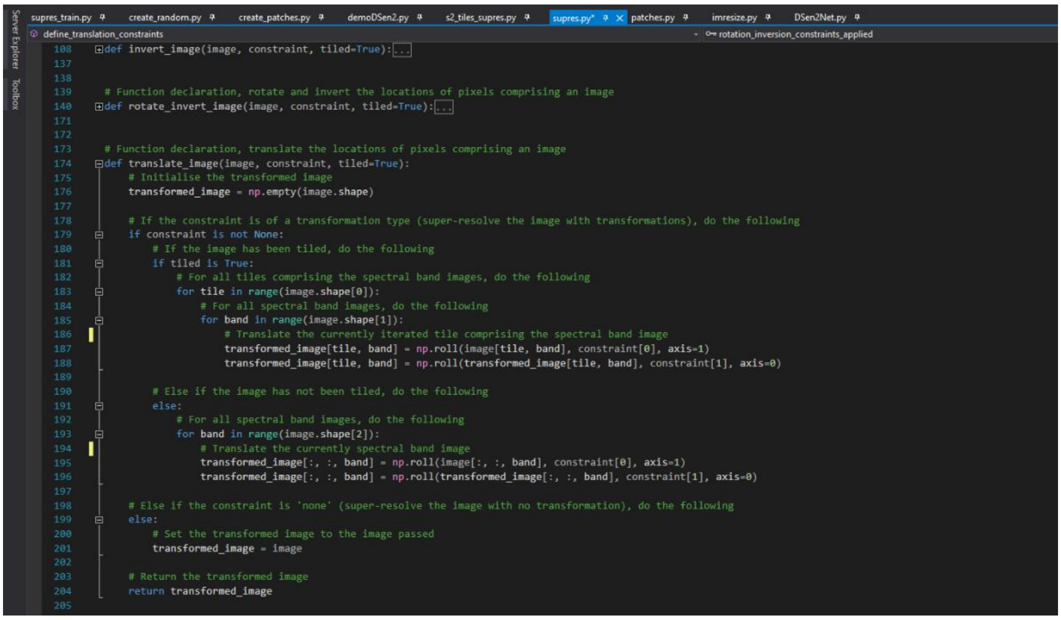


Figure 42: Depiction of the rotation, inversion, and the rotation + inversion transformation constraints, defined in the 'DSen2_20' method that is implemented within the 'supres.py' Python script file.

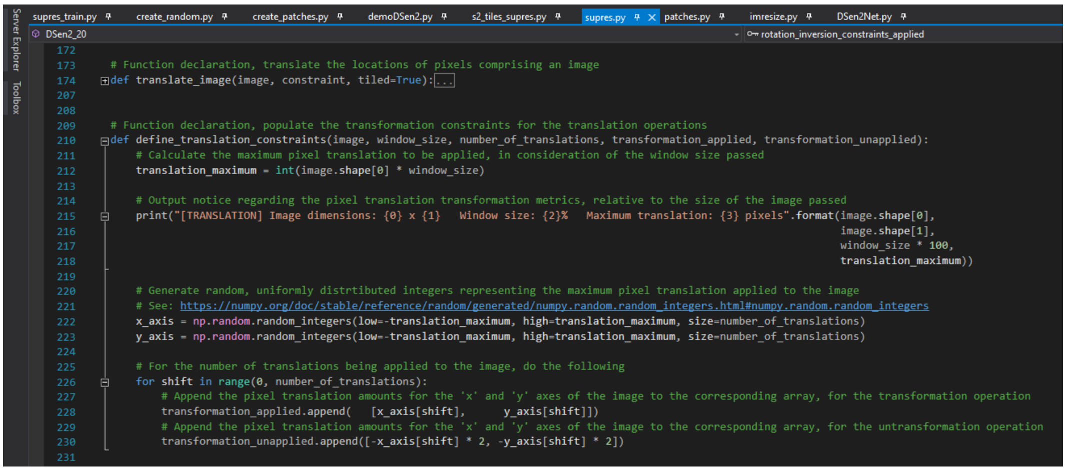
Appendix H:



```
def invert_image(image, constraint, tiled=True):...


 # Function declaration, rotate and invert the locations of pixels comprising an image
def rotate_invert_image(image, constraint, tiled=True):...


 # Function declaration, translate the locations of pixels comprising an image
def translate_image(image, constraint, tiled=True):
    # Initialise the transformed image
    transformed_image = np.empty(image.shape)

    # If the constraint is of a transformation type (super-resolve the image with transformations), do the following
    if constraint is not None:
        # If the image has been tiled, do the following
        if tiled is True:
            # For all tiles comprising the spectral band images, do the following
            for tile in range(image.shape[0]):
                # For all spectral band images, do the following
                for band in range(image.shape[1]):
                    # Translate the currently iterated tile comprising the spectral band image
                    transformed_image[tile, band] = np.roll(image[tile, band], constraint[0], axis=1)
                    transformed_image[tile, band] = np.roll(transformed_image[tile, band], constraint[1], axis=0)

        # Else if the image has not been tiled, do the following
        else:
            # For all spectral band images, do the following
            for band in range(image.shape[2]):
                # Translate the currently spectral band image
                transformed_image[:, :, band] = np.roll(image[:, :, band], constraint[0], axis=1)
                transformed_image[:, :, band] = np.roll(transformed_image[:, :, band], constraint[1], axis=0)

    # Else if the constraint is 'none' (super-resolve the image with no transformation), do the following
    else:
        # Set the transformed image to the image passed
        transformed_image = image

    # Return the transformed image
    return transformed_image
```

Figure 43: Depiction of the 'translate_image' method implemented within the 'supres.py' Python script file, devoted to generating positionally offset transformations of Sentinel-2 image tiles.

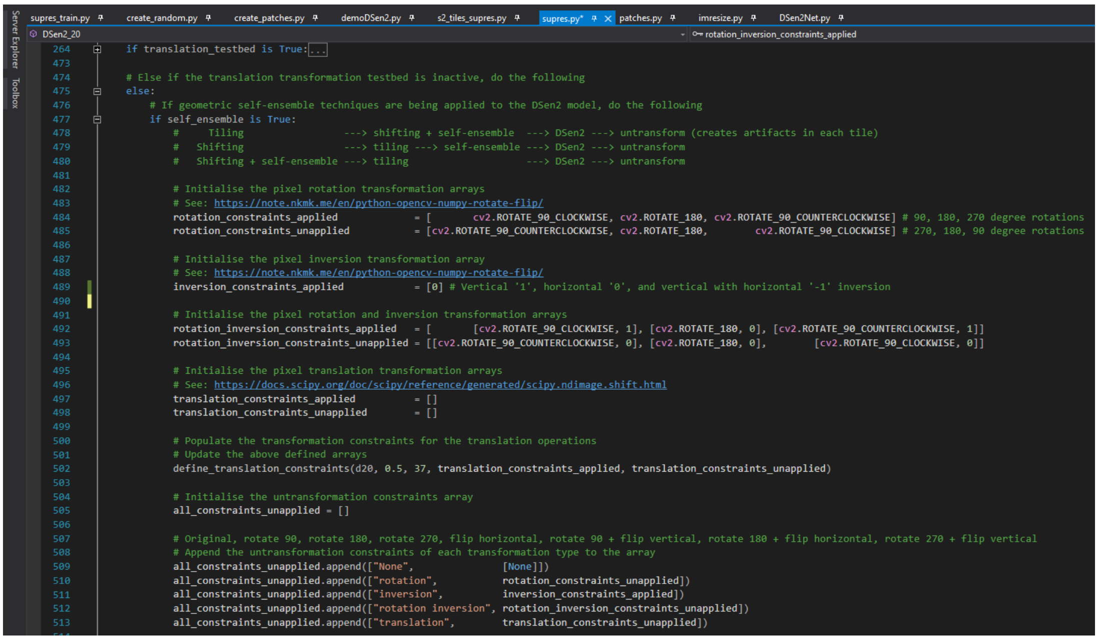
Appendix I:



```

 # Function declaration, translate the locations of pixels comprising an image
def translate_image(image, constraint, tiled=True):...


 # Function declaration, populate the transformation constraints for the translation operations
def define_translation_constraints(image, window_size, number_of_translations, transformation_applied, transformation_unapplied):
    # Calculate the maximum pixel translation to be applied, in consideration of the window size passed
    translation_maximum = int(image.shape[0] * window_size)

    # Output notice regarding the pixel translation transformation metrics, relative to the size of the image passed
    print("[TRANSLATION] Image dimensions: {0} x {1}   Window size: {2}%   Maximum translation: {3} pixels".format(image.shape[0],
                                                                                                                  image.shape[1],
                                                                                                                  window_size * 100,
                                                                                                                  translation_maximum))

    # Generate random, uniformly distrtibuted integers representing the maximum pixel translation applied to the image
    # See: https://numpy.org/doc/stable/reference/random/generated/numpy.random.random_integers.html#numpy.random.random_integers
    x_axis = np.random.random_integers(low=-translation_maximum, high=translation_maximum, size=number_of_translations)
    y_axis = np.random.random_integers(low=-translation_maximum, high=translation_maximum, size=number_of_translations)

    # For the number of translations being applied to the image, do the following
    for shift in range(0, number_of_translations):
        # Append the pixel translation amounts for the 'x' and 'y' axes of the image to the corresponding array, for the transformation operation
        transformation_applied.append(   [x_axis[shift],      y_axis[shift]])
        # Append the pixel translation amounts for the 'x' and 'y' axes of the image to the corresponding array, for the untransformation operation
        transformation_unapplied.append([-x_axis[shift] * 2, -y_axis[shift] * 2])

```

Figure 44: Depiction of the 'define_translation_constraints' method implemented within the 'supres.py' Python script file, devoted to populating the transformation constraints used by the stochastic translation scheme.

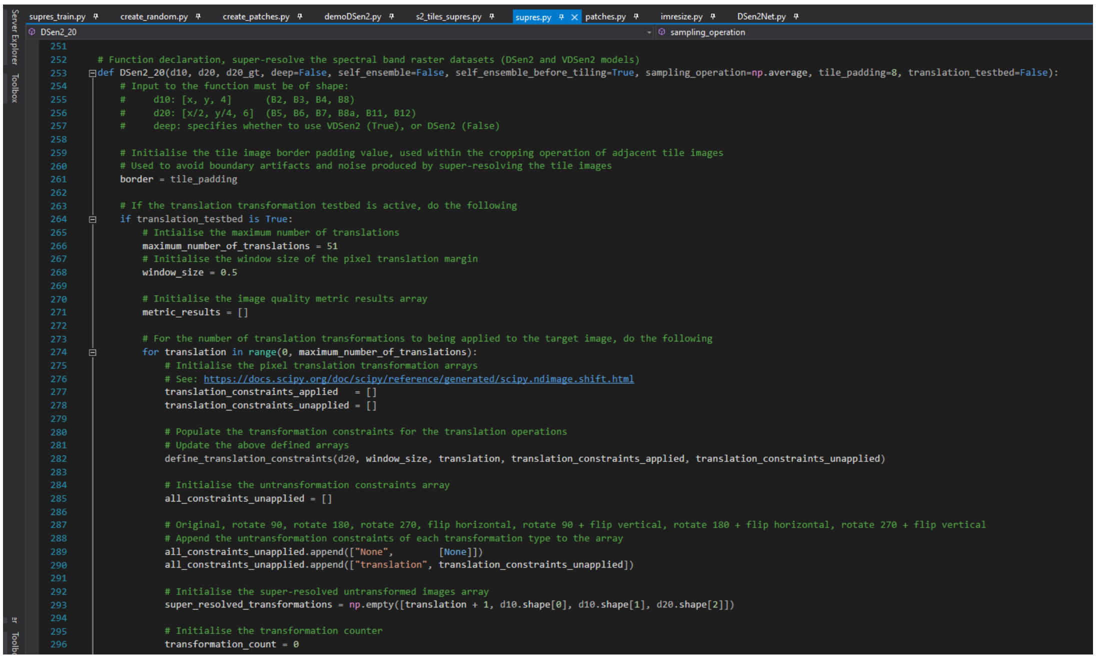
Appendix J:



```python
    if translation_testbed is True:[...]

    # Else if the translation transformation testbed is inactive, do the following
    else:
        # If geometric self-ensemble techniques are being applied to the DSen2 model, do the following
        if self_ensemble is True:
            #     Tiling                 ---> shifting + self-ensemble  ---> DSen2 ---> untransform (creates artifacts in each tile)
            #   Shifting                 ---> tiling ---> self-ensemble ---> DSen2 ---> untransform
            #   Shifting + self-ensemble ---> tiling                    ---> DSen2 ---> untransform

            # Initialise the pixel rotation transformation arrays
            # See: https://note.nkmk.me/en/python-opencv-numpy-rotate-flip/
            rotation_constraints_applied             = [       cv2.ROTATE_90_CLOCKWISE, cv2.ROTATE_180, cv2.ROTATE_90_COUNTERCLOCKWISE] # 90, 180, 270 degree rotations
            rotation_constraints_unapplied           = [cv2.ROTATE_90_COUNTERCLOCKWISE, cv2.ROTATE_180,        cv2.ROTATE_90_CLOCKWISE] # 270, 180, 90 degree rotations

            # Initialise the pixel inversion transformation array
            # See: https://note.nkmk.me/en/python-opencv-numpy-rotate-flip/
            inversion_constraints_applied            = [0] # Vertical '1', horizontal '0', and vertical with horizontal '-1' inversion

            # Initialise the pixel rotation and inversion transformation arrays
            rotation_inversion_constraints_applied   = [       [cv2.ROTATE_90_CLOCKWISE, 1], [cv2.ROTATE_180, 0], [cv2.ROTATE_90_COUNTERCLOCKWISE, 1]]
            rotation_inversion_constraints_unapplied = [[cv2.ROTATE_90_COUNTERCLOCKWISE, 0], [cv2.ROTATE_180, 0],        [cv2.ROTATE_90_CLOCKWISE, 0]]

            # Initialise the pixel translation transformation arrays
            # See: https://docs.scipy.org/doc/scipy/reference/generated/scipy.ndimage.shift.html
            translation_constraints_applied          = []
            translation_constraints_unapplied        = []

            # Populate the transformation constraints for the translation operations
            # Update the above defined arrays
            define_translation_constraints(d20, 0.5, 37, translation_constraints_applied, translation_constraints_unapplied)

            # Initialise the untransformation constraints array
            all_constraints_unapplied = []

            # Original, rotate 90, rotate 180, rotate 270, flip horizontal, rotate 90 + flip vertical, rotate 180 + flip horizontal, rotate 270 + flip vertical
            # Append the untransformation constraints of each transformation type to the array
            all_constraints_unapplied.append(["None",               [None]])
            all_constraints_unapplied.append(["rotation",           rotation_constraints_unapplied])
            all_constraints_unapplied.append(["inversion",          inversion_constraints_applied])
            all_constraints_unapplied.append(["rotation inversion", rotation_inversion_constraints_unapplied])
            all_constraints_unapplied.append(["translation",        translation_constraints_unapplied])
```

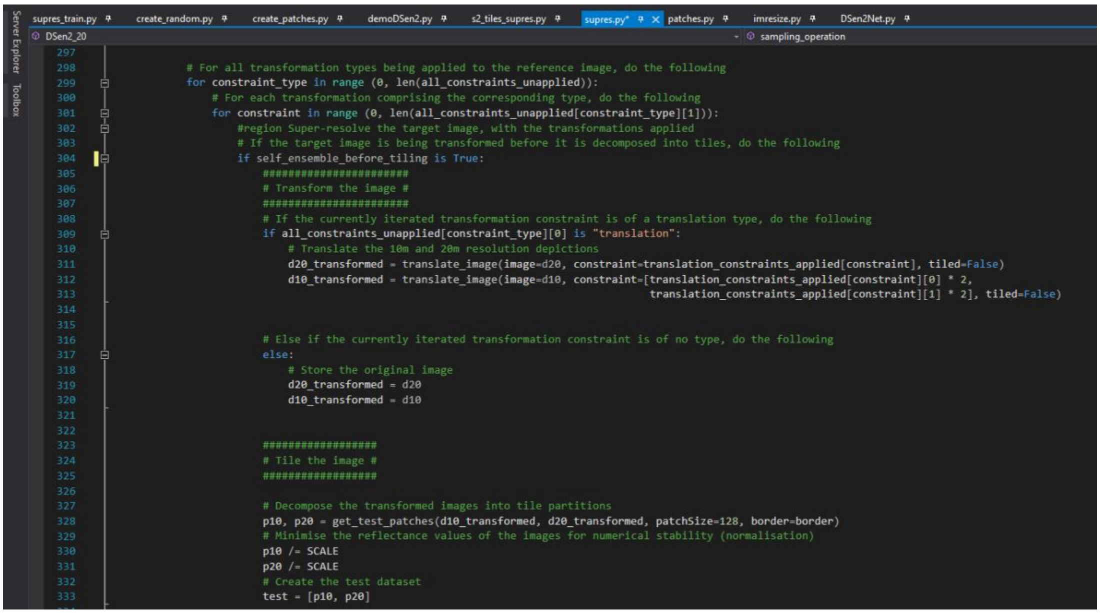
Figure 45: Depiction of the translation transformation constraints, defined in the 'DSen2_20' method that is implemented within the 'supres.py' Python script file. Demonstrating a method-guided approach to their population.

Appendix K:



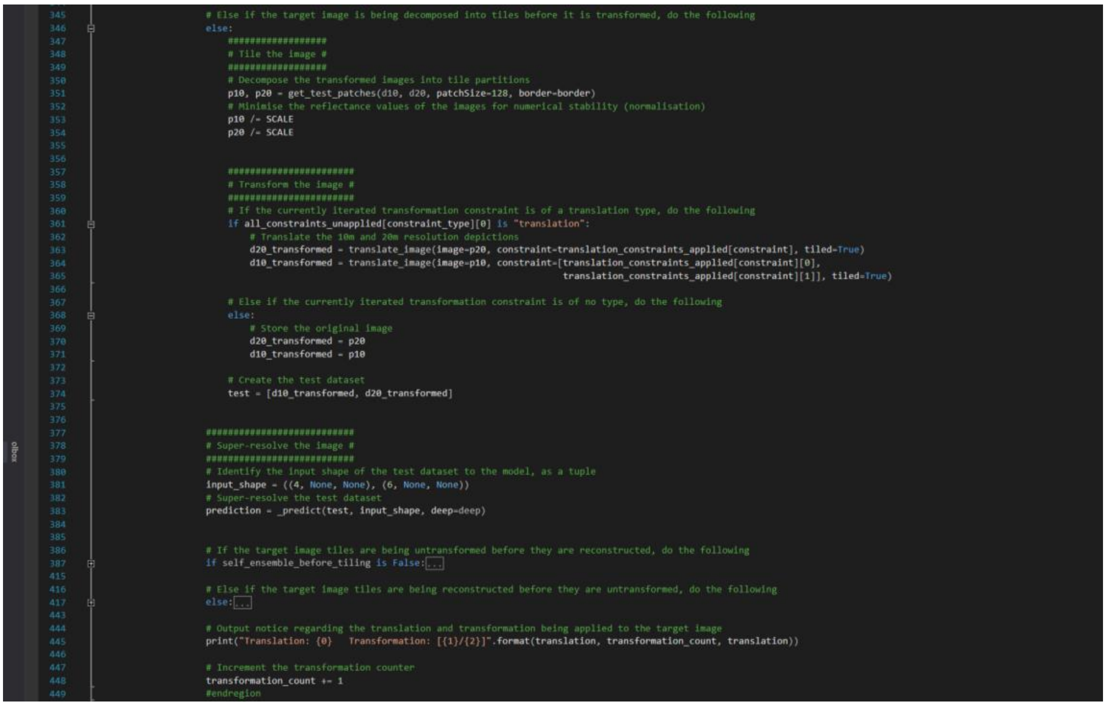
```python
# Function declaration, super-resolve the spectral band raster datasets (DSen2 and VDSen2 models)
def DSen2_20(d10, d20, d20_gt, deep=False, self_ensemble=False, self_ensemble_before_tiling=True, sampling_operation=np.average, tile_padding=8, translation_testbed=False):
    # Input to the function must be of shape:
    #     d10: [x, y, 4]      (B2, B3, B4, B8)
    #     d20: [x/2, y/4, 6]  (B5, B6, B7, B8a, B11, B12)
    #     deep: specifies whether to use VDSen2 (True), or DSen2 (False)

    # Initialise the tile image border padding value, used within the cropping operation of adjacent tile images
    # Used to avoid boundary artifacts and noise produced by super-resolving the tile images
    border = tile_padding

    # If the translation transformation testbed is active, do the following
    if translation_testbed is True:
        # Intialise the maximum number of translations
        maximum_number_of_translations = 51
        # Initialise the window size of the pixel translation margin
        window_size = 0.5

        # Initialise the image quality metric results array
        metric_results = []

        # For the number of translation transformations to being applied to the target image, do the following
        for translation in range(0, maximum_number_of_translations):
            # Initialise the pixel translation transformation arrays
            # See: https://docs.scipy.org/doc/scipy/reference/generated/scipy.ndimage.shift.html
            translation_constraints_applied   = []
            translation_constraints_unapplied = []

            # Populate the transformation constraints for the translation operations
            # Update the above defined arrays
            define_translation_constraints(d20, window_size, translation, translation_constraints_applied, translation_constraints_unapplied)

            # Initialise the untransformation constraints array
            all_constraints_unapplied = []

            # Original, rotate 90, rotate 180, rotate 270, flip horizontal, rotate 90 + flip vertical, rotate 180 + flip horizontal, rotate 270 + flip vertical
            # Append the untransformation constraints of each transformation type to the array
            all_constraints_unapplied.append(["None",        [None]])
            all_constraints_unapplied.append(["translation", translation_constraints_unapplied])

            # Initialise the super-resolved untransformed images array
            super_resolved_transformations = np.empty([translation + 1, d10.shape[0], d10.shape[1], d20.shape[2]])

            # Initialise the transformation counter
            transformation_count = 0
```

Figure 46: Depiction of the stochastic translation scheme testbed, implemented within the 'supres.py' Python script file. Code sample 1 of 4.



```python
        # For all transformation types being applied to the reference image, do the following
        for constraint_type in range (0, len(all_constraints_unapplied)):
            # For each transformation comprising the corresponding type, do the following
            for constraint in range (0, len(all_constraints_unapplied[constraint_type][1])):
                #region Super-resolve the target image, with the transformations applied
                # If the target image is being transformed before it is decomposed into tiles, do the following
                if self_ensemble_before_tiling is True:
                    #######################
                    # Transform the image #
                    #######################
                    # If the currently iterated transformation constraint is of a translation type, do the following
                    if all_constraints_unapplied[constraint_type][0] is "translation":
                        # Translate the 10m and 20m resolution depictions
                        d20_transformed = translate_image(image=d20, constraint=translation_constraints_applied[constraint], tiled=False)
                        d10_transformed = translate_image(image=d10, constraint=[translation_constraints_applied[constraint][0] * 2,
                                                                     translation_constraints_applied[constraint][1] * 2], tiled=False)


                    # Else if the currently iterated transformation constraint is of no type, do the following
                    else:
                        # Store the original image
                        d20_transformed = d20
                        d10_transformed = d10


                    ##################
                    # Tile the image #
                    ##################

                    # Decompose the transformed images into tile partitions
                    p10, p20 = get_test_patches(d10_transformed, d20_transformed, patchSize=128, border=border)
                    # Minimise the reflectance values of the images for numerical stability (normalisation)
                    p10 /= SCALE
                    p20 /= SCALE
                    # Create the test dataset
                    test = [p10, p20]
```

Figure 47: Depiction of the stochastic translation scheme testbed, implemented within the 'supres.py' Python script file. Code sample 2 of 4.



```python
                # Else if the target image is being decomposed into tiles before it is transformed, do the following
                else:
                    ##################
                    # Tile the image #
                    ##################
                    # Decompose the transformed images into tile partitions
                    p10, p20 = get_test_patches(d10, d20, patchSize=128, border=border)
                    # Minimise the reflectance values of the images for numerical stability (normalisation)
                    p10 /= SCALE
                    p20 /= SCALE


                    #######################
                    # Transform the image #
                    #######################
                    # If the currently iterated transformation constraint is of a translation type, do the following
                    if all_constraints_unapplied[constraint_type][0] is "translation":
                        # Translate the 10m and 20m resolution depictions
                        d20_transformed = translate_image(image=p20, constraint=translation_constraints_applied[constraint], tiled=True)
                        d10_transformed = translate_image(image=p10, constraint=[translation_constraints_applied[constraint][0],
                                                                     translation_constraints_applied[constraint][1]], tiled=True)

                    # Else if the currently iterated transformation constraint is of no type, do the following
                    else:
                        # Store the original image
                        d20_transformed = p20
                        d10_transformed = p10

                    # Create the test dataset
                    test = [d10_transformed, d20_transformed]


                ###########################
                # Super-resolve the image #
                ###########################
                # Identify the input shape of the test dataset to the model, as a tuple
                input_shape = ((4, None, None), (6, None, None))
                # Super-resolve the test dataset
                prediction = _predict(test, input_shape, deep=deep)


                # If the target image tiles are being untransformed before they are reconstructed, do the following
                if self_ensemble_before_tiling is False:[...]

                # Else if the target image tiles are being reconstructed before they are untransformed, do the following
                else:[...]

                # Output notice regarding the translation and transformation being applied to the target image
                print("Translation: {0}   Transformation: [{1}/{2}]".format(translation, transformation_count, translation))

                # Increment the transformation counter
                transformation_count += 1
                #endregion
```

Figure 48: Depiction of the stochastic translation scheme testbed, implemented within the 'supres.py' Python script file. Code sample 3 of 4.

```python
            # Sample the tranformed image series to resolve to a single super-resolved image
            super_resolved = sampling_operation(super_resolved_transformations, axis=0)

            # Min-max contrast stretch the super-resolved image
            target_image = min_max_contrast_stretch(super_resolved, "Target", write_to_file=False)

            # Min-max contrast stretch the ground truth image
            reference_image = min_max_contrast_stretch(d20_gt, "Reference", write_to_file=False)

            # Calculate the focus image quality metric results for the target image, relative to the reference image (ground truth)
            PSNR = psnr(target_image, reference_image)
            RMSE = rmse(target_image, reference_image)
            SSIM = ssim(target_image, reference_image)
            UQI  =  uqi(target_image, reference_image)
            SAM  =  sam(target_image, reference_image)

            # Append the image quality metrics to the array, representative of the number of translations incurred
            metric_results.append([translation, PSNR, RMSE, SSIM[0], UQI, SAM])

        # Write the metric results for the test case to an external (.txt) file, corresponding to the window size and number of translations applied
        np.savetxt("C:/DSen2/DSen2-master/data/output/translation/" + str(int(window_size * 100)) + "/TranslationResults" + str(maximum_number_of_translations) + ".txt",
                   np.array(metric_results), fmt="%s")
```

Figure 49: Depiction of the stochastic translation scheme testbed, implemented within the 'supres.py' Python script file. Code sample 4 of 4.

Appendix L:

```python
supres_train.py   create_random.py   create_patches.py   demoDSen2.py   s2_tiles_supres.py   supres.py*   patches.py   imresize.py   DSen2Net.py
DSen2_20

    # If the translation transformation testbed is active, do the following
    if translation_testbed is True:[...]

    # Else if the translation transformation testbed is inactive, do the following
    else:
        # If geometric self-ensemble techniques are being applied to the DSen2 model, do the following
        if self_ensemble is True:
            #     Tiling                 ---> shifting + self-ensemble  ---> DSen2 ---> untransform (creates artifacts in each tile)
            #   Shifting                 ---> tiling ---> self-ensemble ---> DSen2 ---> untransform
            #   Shifting + self-ensemble ---> tiling                    ---> DSen2 ---> untransform

            # Initialise the pixel rotation transformation arrays
            # See: https://note.nkmk.me/en/python-opencv-numpy-rotate-flip/
            rotation_constraints_applied             = [       cv2.ROTATE_90_CLOCKWISE, cv2.ROTATE_180, cv2.ROTATE_90_COUNTERCLOCKWISE] # 90, 180, 270 degree rotations
            rotation_constraints_unapplied           = [cv2.ROTATE_90_COUNTERCLOCKWISE, cv2.ROTATE_180,        cv2.ROTATE_90_CLOCKWISE] # 270, 180, 90 degree rotations

            # Initialise the pixel inversion transformation array
            # See: https://note.nkmk.me/en/python-opencv-numpy-rotate-flip/
            inversion_constraints_applied            = [0] # Vertical '1', horizontal '0', and vertical with horizontal '-1' inversion
            #inversion_constraints_applied            = [0, 1, -1] # Vertical '1', horizontal '0', and vertical with horizontal '-1' inversion

            # Initialise the pixel rotation and inversion transformation arrays
            rotation_inversion_constraints_applied   = [       [cv2.ROTATE_90_CLOCKWISE, 1], [cv2.ROTATE_180, 0], [cv2.ROTATE_90_COUNTERCLOCKWISE, 1]]
            rotation_inversion_constraints_unapplied = [[cv2.ROTATE_90_COUNTERCLOCKWISE, 0], [cv2.ROTATE_180, 0],        [cv2.ROTATE_90_CLOCKWISE, 0]]

            # Initialise the pixel translation transformation arrays
            # See: https://docs.scipy.org/doc/scipy/reference/generated/scipy.ndimage.shift.html
            translation_constraints_applied          = []
            translation_constraints_unapplied        = []

            # Populate the transformation constraints for the translation operations
            # Update the above defined arrays
            define_translation_constraints(d20, 0.5, 37, translation_constraints_applied, translation_constraints_unapplied)

            # Initialise the untransformation constraints array
            all_constraints_unapplied = []

            # Original, rotate 90, rotate 180, rotate 270, flip horizontal, rotate 90 + flip vertical, rotate 180 + flip horizontal, rotate 270 + flip vertical
            # Append the untransformation constraints of each transformation type to the array
            all_constraints_unapplied.append(["None",               [None]])
            all_constraints_unapplied.append(["rotation",           rotation_constraints_unapplied])
            all_constraints_unapplied.append(["inversion",          inversion_constraints_applied])
            all_constraints_unapplied.append(["rotation inversion", rotation_inversion_constraints_unapplied])
            all_constraints_unapplied.append(["translation",        translation_constraints_unapplied])
```

Figure 50: Depiction of the geometric self-ensemble and the stochastic translation scheme, integrated into the DSen2 model framework, implemented within the 'supres.py' Python script file. Code sample 1 of 7.

```python
# Intialise the number of total transformations being performed
number_of_transformations = 0

# For all types of transformation constraint contained in the array, do the following
for constraint_type in range (0, len(all_constraints_unapplied)):
    # Increment the number of total transformations by the number of transformations contained by the transformation type
    number_of_transformations += len(all_constraints_unapplied[constraint_type][1])

# Initialise the super-resolved untransformed images array
super_resolved_transformations = np.empty([number_of_transformations, d10.shape[0], d10.shape[1], d20.shape[2]])

# Initialise the transformation counter
transformation_count = 0

# For all transformation types being applied to the reference image, do the following
for constraint_type in range (0, len(all_constraints_unapplied)):
    # For each transformation comprising the corresponding type, do the following
    for constraint in range (0, len(all_constraints_unapplied[constraint_type][1])):
        # If the target image is being transformed before it is decomposed into tiles, do the following
        if self_ensemble_before_tiling is True:
            #######################
            # Transform the image #
            #######################
            # If the currently iterated transformation constraint is of a rotary type, do the following
            if all_constraints_unapplied[constraint_type][0] is "rotation":
                # Rotate the 10m and 20m resolution depictions
                d20_transformed = rotate_image(image=d20, constraint=rotation_constraints_applied[constraint], tiled=False)
                d10_transformed = rotate_image(image=d10, constraint=rotation_constraints_applied[constraint], tiled=False)

            # Else if the currently iterated transformation constraint is of a inversion type, do the following
            elif all_constraints_unapplied[constraint_type][0] is "inversion":
                # Invert the 10m and 20m resolution depictions
                d20_transformed = invert_image(image=d20, constraint=inversion_constraints_applied[constraint], tiled=False)
                d10_transformed = invert_image(image=d10, constraint=inversion_constraints_applied[constraint], tiled=False)

            # Else if the currently iterated transformation constraint is of a rotary and inversion type, do the following
            elif all_constraints_unapplied[constraint_type][0] is "rotation inversion":
                # Rotate and invert the 10m and 20m resolution depictions
                d20_transformed = rotate_invert_image(image=d20, constraint=rotation_inversion_constraints_applied[constraint], tiled=False)
                d10_transformed = rotate_invert_image(image=d10, constraint=rotation_inversion_constraints_applied[constraint], tiled=False)
```

Figure 51: Depiction of the geometric self-ensemble and the stochastic translation scheme, integrated into the DSen2 model framework, implemented within the 'supres.py' Python script file. Code sample 2 of 7.

```python
            # Else if the currently iterated transformation constraint is of a translation type, do the following
            elif all_constraints_unapplied[constraint_type][0] is "translation":
                # Translate the 10m and 20m resolution depictions
                d20_transformed = translate_image(image=d20, constraint=translation_constraints_applied[constraint], tiled=False)
                d10_transformed = translate_image(image=d10, constraint=[translation_constraints_applied[constraint][0] * 2,
                                                                          translation_constraints_applied[constraint][1] * 2], tiled=False)

            # Else if the currently iterated transformation constraint is of no type, do the following
            else:
                # Store the original image
                d20_transformed = d20
                d10_transformed = d10


            ##################
            # Tile the image #
            ##################
            # Decompose the transformed images into tile partitions
            p10, p20 = get_test_patches(d10_transformed, d20_transformed, patchSize=128, border=border)
            # Minimise the reflectance values of the images for numerical stability (normalisation)
            p10 /= SCALE
            p20 /= SCALE
            # Create the test dataset
            test = [p10, p20]

            # Min-max contrast stretch the transformed image and write it to an external file
            min_max_contrast_stretch(d20_transformed, "Transformed", '../data/output/rgbs/20m/transformations/transformed/' + str(transformation_count + 1), write_to_file=True)
            min_max_contrast_stretch(d10_transformed, "Transformed", '../data/output/rgbs/20m/transformations/transformed/' + str(transformation_count + 10), write_to_file=True)

        # Else if the target image is being decomposed into tiles before it is transformed, do the following
        else:
            ##################
            # Tile the image #
            ##################
            # Decompose the transformed images into tile partitions
            p10, p20 = get_test_patches(d10, d20, patchSize=128, border=border)
            # Minimise the reflectance values of the images for numerical stability (normalisation)
            p10 /= SCALE
            p20 /= SCALE
```

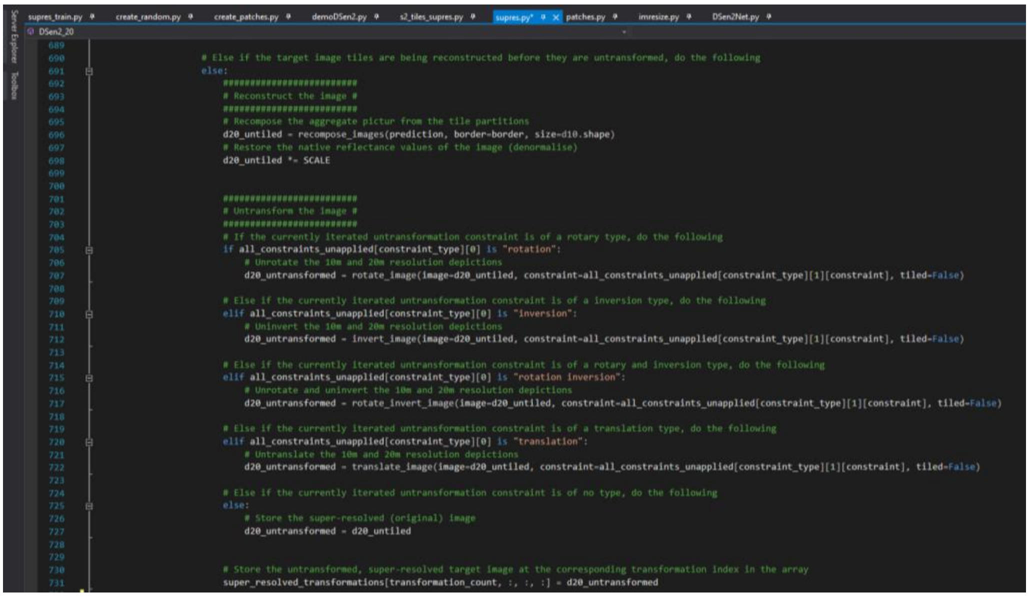
Figure 52: Depiction of the geometric self-ensemble and the stochastic translation scheme, integrated into the DSen2 model framework, implemented within the 'supres.py' Python script file. Code sample 3 of 7.

```python
                    ########################
                    # Transform the image #
                    ########################
                    # If the currently iterated transformation constraint is of a rotary type, do the following
                    if all_constraints_unapplied[constraint_type][0] is "rotation":
                        # Rotate the 10m and 20m resolution depictions
                        d20_transformed = rotate_image(image=p20, constraint=rotation_constraints_applied[constraint], tiled=True)
                        d10_transformed = rotate_image(image=p10, constraint=rotation_constraints_applied[constraint], tiled=True)

                    # Else if the currently iterated transformation constraint is of a inversion type, do the following
                    elif all_constraints_unapplied[constraint_type][0] is "inversion":
                        # Invert the 10m and 20m resolution depictions
                        d20_transformed = invert_image(image=p20, constraint=inversion_constraints_applied[constraint], tiled=True)
                        d10_transformed = invert_image(image=p10, constraint=inversion_constraints_applied[constraint], tiled=True)

                    # Else if the currently iterated transformation constraint is of a rotary and inversion type, do the following
                    elif all_constraints_unapplied[constraint_type][0] is "rotation inversion":
                        # Rotate and invert the 10m and 20m resolution depictions
                        d20_transformed = rotate_invert_image(image=p20, constraint=rotation_inversion_constraints_applied[constraint], tiled=True)
                        d10_transformed = rotate_invert_image(image=p10, constraint=rotation_inversion_constraints_applied[constraint], tiled=True)

                    # Else if the currently iterated transformation constraint is of a translation type, do the following
                    elif all_constraints_unapplied[constraint_type][0] is "translation":
                        # Translate the 10m and 20m resolution depictions
                        d20_transformed = translate_image(image=p20, constraint=translation_constraints_applied[constraint], tiled=True)
                        d10_transformed = translate_image(image=p10, constraint=[translation_constraints_applied[constraint][0],
                                                                                  translation_constraints_applied[constraint][1]], tiled=True)

                    # Else if the currently iterated transformation constraint is of no type, do the following
                    else:
                        # Store the original image
                        d20_transformed = p20
                        d10_transformed = p10

                    # Create the test dataset
                    test = [d10_transformed, d20_transformed]


                ############################
                # Super-resolve the image #
                ############################
                # Identify the input shape of the test dataset to the model, as a tuple
                input_shape = ((4, None, None), (6, None, None))
                # Super-resolve the test dataset
                prediction = _predict(test, input_shape, deep=deep)
```

Figure 53: Depiction of the geometric self-ensemble and the stochastic translation scheme, integrated into the DSen2 model framework, implemented within the 'supres.py' Python script file. Code sample 4 of 7.

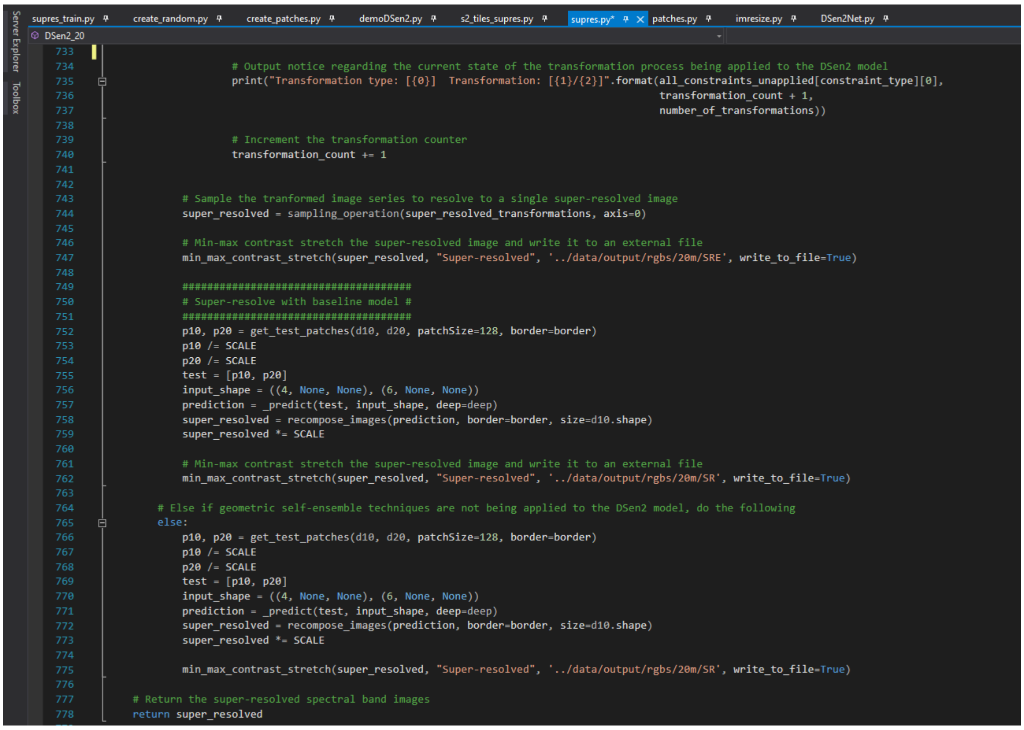
```python
                # If the target image tiles are being untransformed before they are reconstructed, do the following
                if self_ensemble_before_tiling is False:
                    ##########################
                    # Untransform the image #
                    ##########################
                    # If the currently iterated untransformation constraint is of a rotary type, do the following
                    if all_constraints_unapplied[constraint_type][0] is "rotation":
                        # Unrotate the 10m and 20m resolution depictions
                        d20_untransformed = rotate_image(image=prediction, constraint=all_constraints_unapplied[constraint_type][1][constraint], tiled=True)

                    # Else if the currently iterated untransformation constraint is of a inversion type, do the following
                    elif all_constraints_unapplied[constraint_type][0] is "inversion":
                        # Uninvert the 10m and 20m resolution depictions
                        d20_untransformed = invert_image(image=prediction, constraint=all_constraints_unapplied[constraint_type][1][constraint], tiled=True)

                    # Else if the currently iterated untransformation constraint is of a rotary and inversion type, do the following
                    elif all_constraints_unapplied[constraint_type][0] is "rotation inversion":
                        # Unrotate and uninvert the 10m and 20m resolution depictions
                        d20_untransformed = rotate_invert_image(image=prediction, constraint=all_constraints_unapplied[constraint_type][1][constraint], tiled=True)

                    # Else if the currently iterated untransformation constraint is of a translation type, do the following
                    elif all_constraints_unapplied[constraint_type][0] is "translation":
                        # Untranslate the 10m and 20m resolution depictions
                        d20_untransformed = translate_image(image=prediction, constraint=[all_constraints_unapplied[constraint_type][1][constraint][0] // 2,
                                                                                        all_constraints_unapplied[constraint_type][1][constraint][1] // 2],
                                                                                        tiled=True)

                    # Else if the currently iterated untransformation constraint is of no type, do the following
                    else:
                        # Store the super-resolved (original) image
                        d20_untransformed = prediction


                    ##########################
                    # Reconstruct the image #
                    ##########################
                    # Recompose the aggregate pictur from the tile partitions
                    d20_untiled = recompose_images(d20_untransformed, border=border, size=d10.shape)
                    # Restore the native reflectance values of the image (denormalise)
                    d20_untiled *= SCALE


                    # Store the untransformed, super-resolved target image at the corresponding transformation index in the array
                    super_resolved_transformations[transformation_count, :, :, :] = d20_untiled
```

Figure 54: Depiction of the geometric self-ensemble and the stochastic translation scheme, integrated into the DSen2 model framework, implemented within the 'supres.py' Python script file. Code sample 5 of 7.



```python
            # Else if the target image tiles are being reconstructed before they are untransformed, do the following
            else:
                ##########################
                # Reconstruct the image #
                ##########################
                # Recompose the aggregate pictur from the tile partitions
                d20_untiled = recompose_images(prediction, border=border, size=d10.shape)
                # Restore the native reflectance values of the image (denormalise)
                d20_untiled *= SCALE


                ##########################
                # Untransform the image #
                ##########################
                # If the currently iterated untransformation constraint is of a rotary type, do the following
                if all_constraints_unapplied[constraint_type][0] is "rotation":
                    # Unrotate the 10m and 20m resolution depictions
                    d20_untransformed = rotate_image(image=d20_untiled, constraint=all_constraints_unapplied[constraint_type][1][constraint], tiled=False)

                # Else if the currently iterated untransformation constraint is of a inversion type, do the following
                elif all_constraints_unapplied[constraint_type][0] is "inversion":
                    # Uninvert the 10m and 20m resolution depictions
                    d20_untransformed = invert_image(image=d20_untiled, constraint=all_constraints_unapplied[constraint_type][1][constraint], tiled=False)

                # Else if the currently iterated untransformation constraint is of a rotary and inversion type, do the following
                elif all_constraints_unapplied[constraint_type][0] is "rotation inversion":
                    # Unrotate and uninvert the 10m and 20m resolution depictions
                    d20_untransformed = rotate_invert_image(image=d20_untiled, constraint=all_constraints_unapplied[constraint_type][1][constraint], tiled=False)

                # Else if the currently iterated untransformation constraint is of a translation type, do the following
                elif all_constraints_unapplied[constraint_type][0] is "translation":
                    # Untranslate the 10m and 20m resolution depictions
                    d20_untransformed = translate_image(image=d20_untiled, constraint=all_constraints_unapplied[constraint_type][1][constraint], tiled=False)

                # Else if the currently iterated untransformation constraint is of no type, do the following
                else:
                    # Store the super-resolved (original) image
                    d20_untransformed = d20_untiled


                # Store the untransformed, super-resolved target image at the corresponding transformation index in the array
                super_resolved_transformations[transformation_count, :, :, :] = d20_untransformed
```

Figure 55: Depiction of the geometric self-ensemble and the stochastic translation scheme, integrated into the DSen2 model framework, implemented within the 'supres.py' Python script file. Code sample 6 of 7.



```python

                # Output notice regarding the current state of the transformation process being applied to the DSen2 model
                print("Transformation type: [{0}]  Transformation: [{1}/{2}]".format(all_constraints_unapplied[constraint_type][0],
                                                                                     transformation_count + 1,
                                                                                     number_of_transformations))

                # Increment the transformation counter
                transformation_count += 1


            # Sample the tranformed image series to resolve to a single super-resolved image
            super_resolved = sampling_operation(super_resolved_transformations, axis=0)

            # Min-max contrast stretch the super-resolved image and write it to an external file
            min_max_contrast_stretch(super_resolved, "Super-resolved", '../data/output/rgbs/20m/SRE', write_to_file=True)

            ####################################
            # Super-resolve with baseline model #
            ####################################
            p10, p20 = get_test_patches(d10, d20, patchSize=128, border=border)
            p10 /= SCALE
            p20 /= SCALE
            test = [p10, p20]
            input_shape = ((4, None, None), (6, None, None))
            prediction = _predict(test, input_shape, deep=deep)
            super_resolved = recompose_images(prediction, border=border, size=d10.shape)
            super_resolved *= SCALE

            # Min-max contrast stretch the super-resolved image and write it to an external file
            min_max_contrast_stretch(super_resolved, "Super-resolved", '../data/output/rgbs/20m/SR', write_to_file=True)

        # Else if geometric self-ensemble techniques are not being applied to the DSen2 model, do the following
        else:
            p10, p20 = get_test_patches(d10, d20, patchSize=128, border=border)
            p10 /= SCALE
            p20 /= SCALE
            test = [p10, p20]
            input_shape = ((4, None, None), (6, None, None))
            prediction = _predict(test, input_shape, deep=deep)
            super_resolved = recompose_images(prediction, border=border, size=d10.shape)
            super_resolved *= SCALE

            min_max_contrast_stretch(super_resolved, "Super-resolved", '../data/output/rgbs/20m/SR', write_to_file=True)

    # Return the super-resolved spectral band images
    return super_resolved
```

Figure 56: Depiction of the geometric self-ensemble and the stochastic translation scheme, integrated into the DSen2 model framework, implemented within the 'supres.py' Python script file. Code sample 7 of 7.

Appendix M:

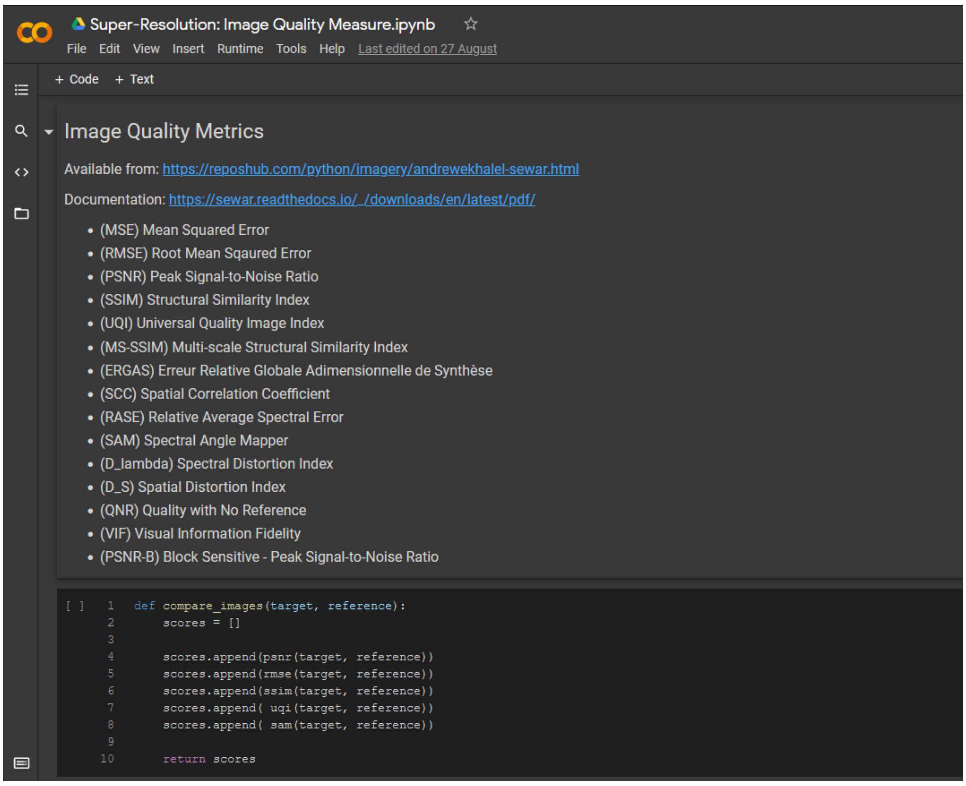


Figure 57: Depiction of the 'Super-Resolution: Image Quality Measure.ipynb' file contents, external to the DSen2 model repository. Which is devoted to performing image quality assessment routines, for the results of all models bound by this study. Code sample 1 of 3.

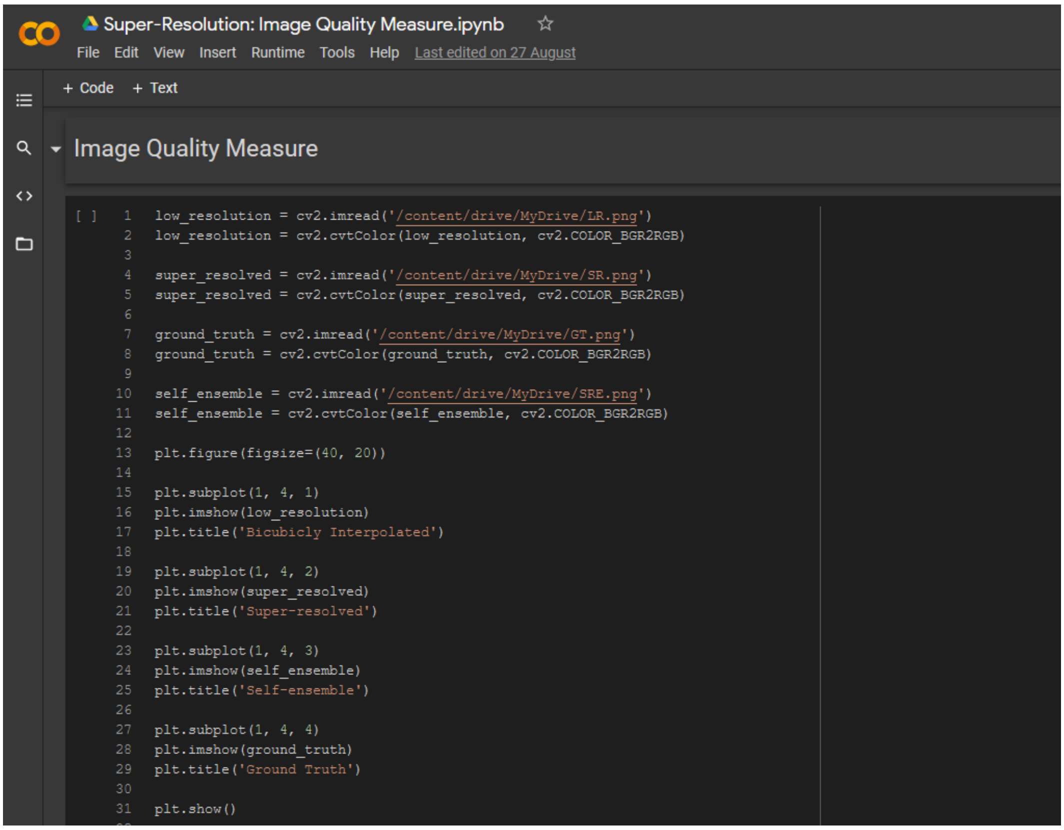
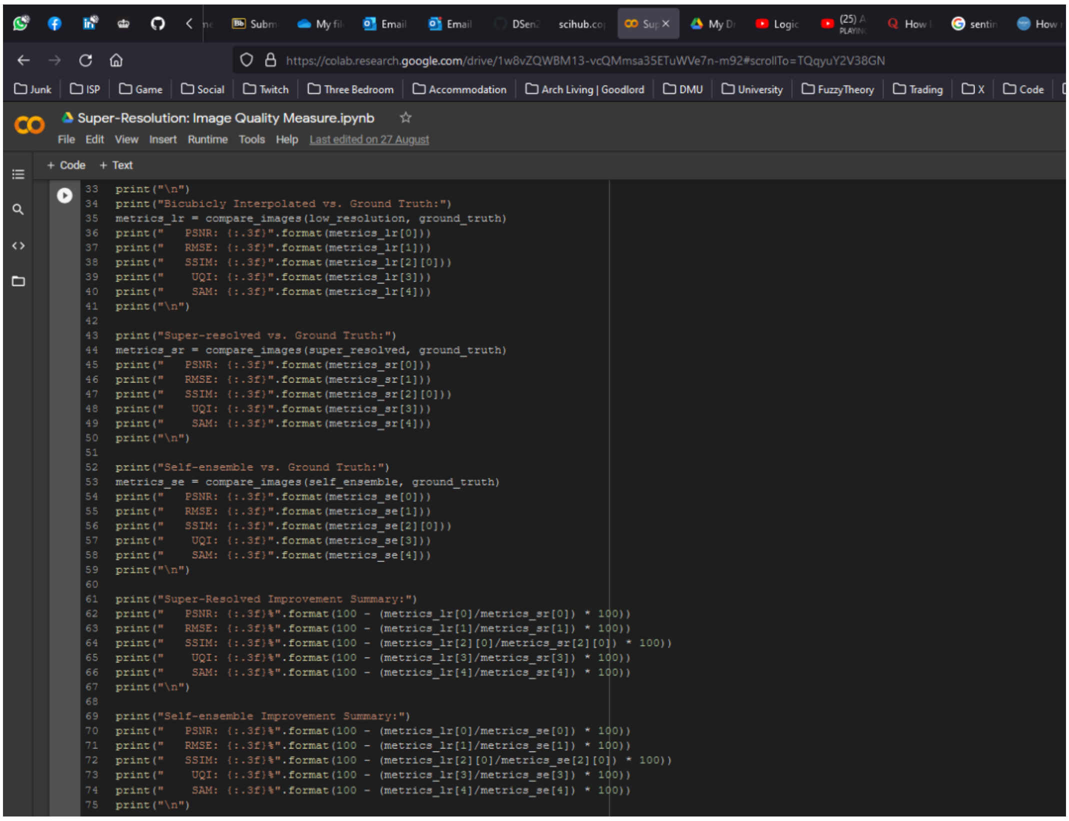


Figure 58: Depiction of the 'Super-Resolution: Image Quality Measure.ipynb' file contents, external to the DSen2 model repository. Which is devoted to performing image quality assessment routines, for the results of all models bound by this study. Code sample 2 of 3.



```
print("\n")
print("Bicubicly Interpolated vs. Ground Truth:")
metrics_lr = compare_images(low_resolution, ground_truth)
print("   PSNR: {:.3f}".format(metrics_lr[0]))
print("   RMSE: {:.3f}".format(metrics_lr[1]))
print("   SSIM: {:.3f}".format(metrics_lr[2][0]))
print("    UQI: {:.3f}".format(metrics_lr[3]))
print("    SAM: {:.3f}".format(metrics_lr[4]))
print("\n")

print("Super-resolved vs. Ground Truth:")
metrics_sr = compare_images(super_resolved, ground_truth)
print("   PSNR: {:.3f}".format(metrics_sr[0]))
print("   RMSE: {:.3f}".format(metrics_sr[1]))
print("   SSIM: {:.3f}".format(metrics_sr[2][0]))
print("    UQI: {:.3f}".format(metrics_sr[3]))
print("    SAM: {:.3f}".format(metrics_sr[4]))
print("\n")

print("Self-ensemble vs. Ground Truth:")
metrics_se = compare_images(self_ensemble, ground_truth)
print("   PSNR: {:.3f}".format(metrics_se[0]))
print("   RMSE: {:.3f}".format(metrics_se[1]))
print("   SSIM: {:.3f}".format(metrics_se[2][0]))
print("    UQI: {:.3f}".format(metrics_se[3]))
print("    SAM: {:.3f}".format(metrics_se[4]))
print("\n")

print("Super-Resolved Improvement Summary:")
print("   PSNR: {:.3f}%".format(100 - (metrics_lr[0]/metrics_sr[0]) * 100))
print("   RMSE: {:.3f}%".format(100 - (metrics_lr[1]/metrics_sr[1]) * 100))
print("   SSIM: {:.3f}%".format(100 - (metrics_lr[2][0]/metrics_sr[2][0]) * 100))
print("    UQI: {:.3f}%".format(100 - (metrics_lr[3]/metrics_sr[3]) * 100))
print("    SAM: {:.3f}%".format(100 - (metrics_lr[4]/metrics_sr[4]) * 100))
print("\n")

print("Self-ensemble Improvement Summary:")
print("   PSNR: {:.3f}%".format(100 - (metrics_lr[0]/metrics_se[0]) * 100))
print("   RMSE: {:.3f}%".format(100 - (metrics_lr[1]/metrics_se[1]) * 100))
print("   SSIM: {:.3f}%".format(100 - (metrics_lr[2][0]/metrics_se[2][0]) * 100))
print("    UQI: {:.3f}%".format(100 - (metrics_lr[3]/metrics_se[3]) * 100))
print("    SAM: {:.3f}%".format(100 - (metrics_lr[4]/metrics_se[4]) * 100))
print("\n")
```

Figure 59: Depiction of the 'Super-Resolution: Image Quality Measure.ipynb' file contents, external to the DSen2 model repository. Which is devoted to performing image quality assessment routines, for the results of all models bound by this study. Code sample 3 of 3.

Appendix N:

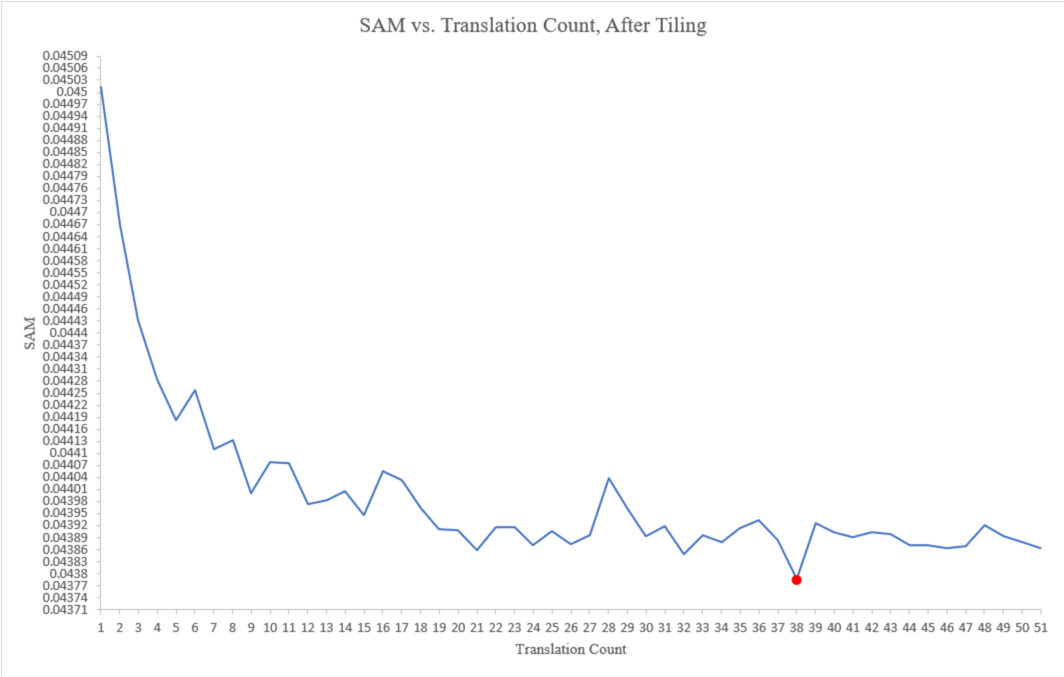


Figure 60: Depiction of the SAM metric evolution, when varying the number of translations employed by the ensemble output, of the stochastic translation scheme. The results represent the average values taken from the three images selected for the experiment (lower is better). A window size of 50 is configured.

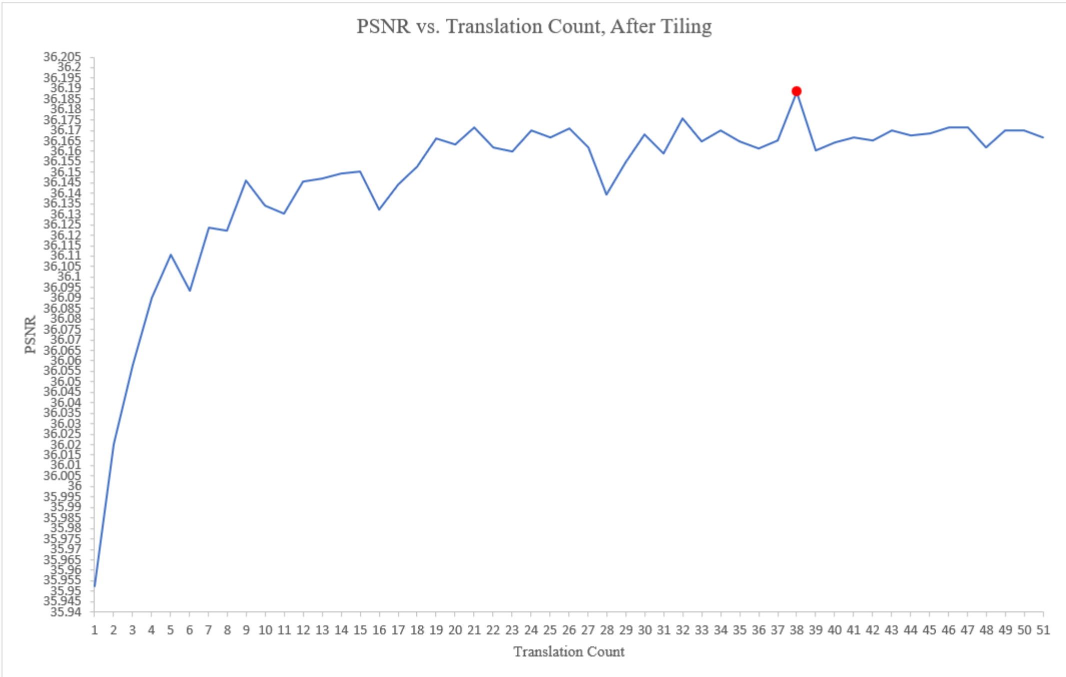


Figure 61: Depiction of the PSNR metric evolution, when varying the number of translations employed by the ensemble output, of the stochastic translation scheme. The results represent the average values taken from the three images selected for the experiment (higher is better). A window size of 50 is configured.

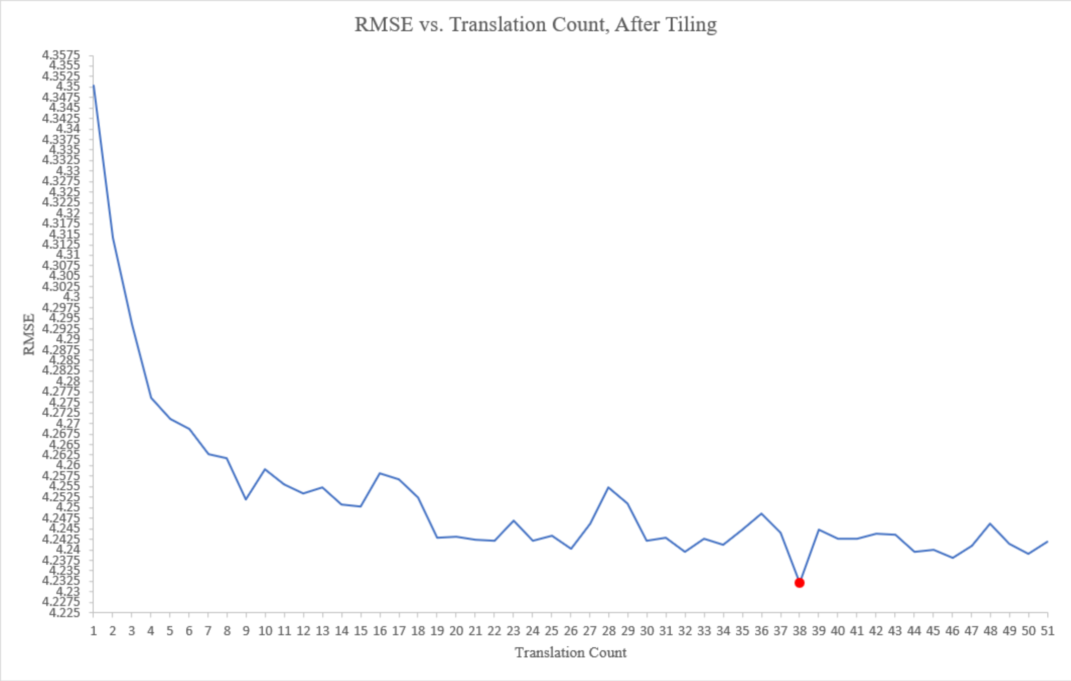


Figure 62: Depiction of the RMSE metric evolution, when varying the number of translations employed by the ensemble output, of the stochastic translation scheme. The results represent the average values taken from the three images selected for the experiment (lower is better). A window size of 50 is configured.

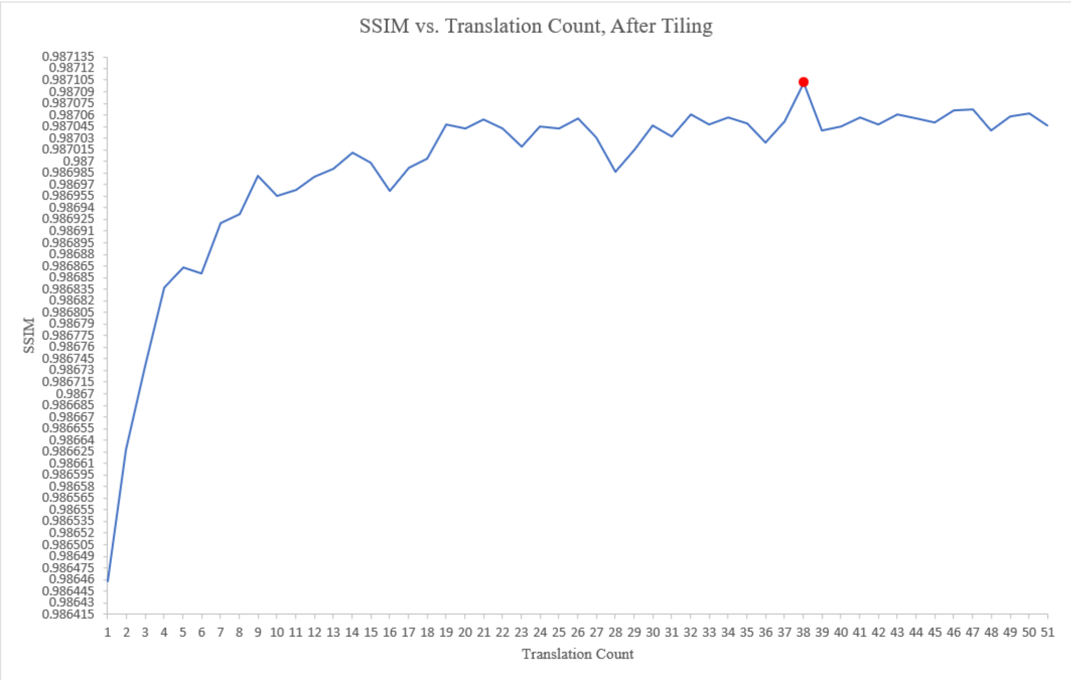


Figure 63: Depiction of the SSIM metric evolution, when varying the number of translations employed by the ensemble output, of the stochastic translation scheme. The results represent the average values taken from the three images selected for the experiment (higher is better). A window size of 50 is configured.

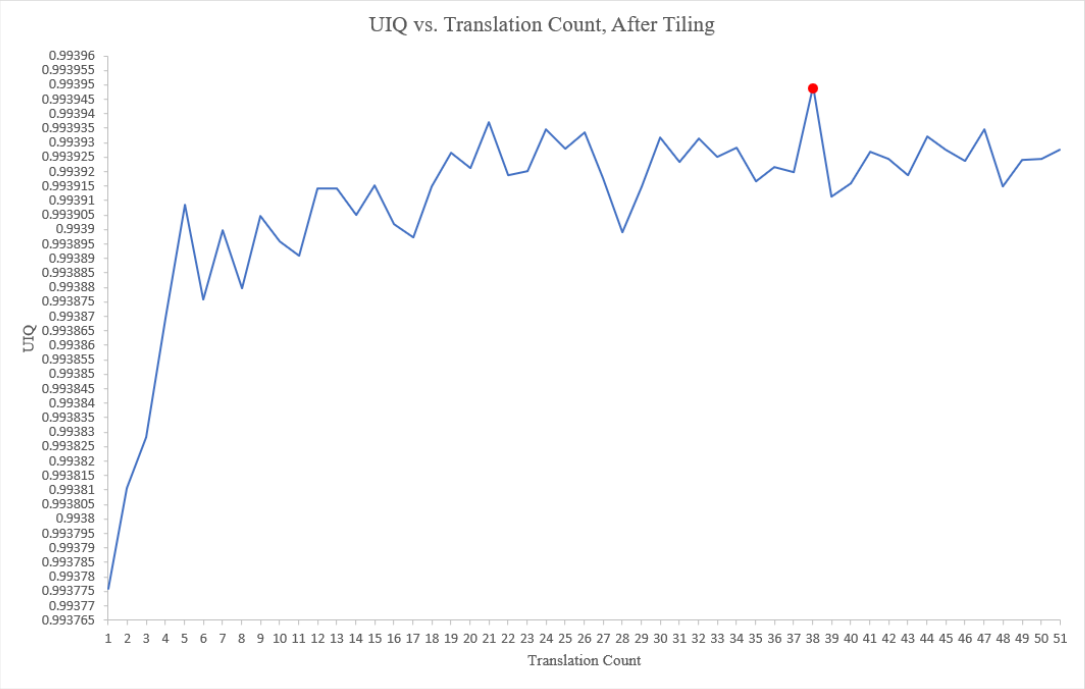


Figure 64: Depiction of the UIQ metric evolution, when varying the number of translations employed by the ensemble output, of the stochastic translation scheme. The results represent the average values taken from the three images selected for the experiment (higher is better). A window size of 50 is configured.

Appendix O:

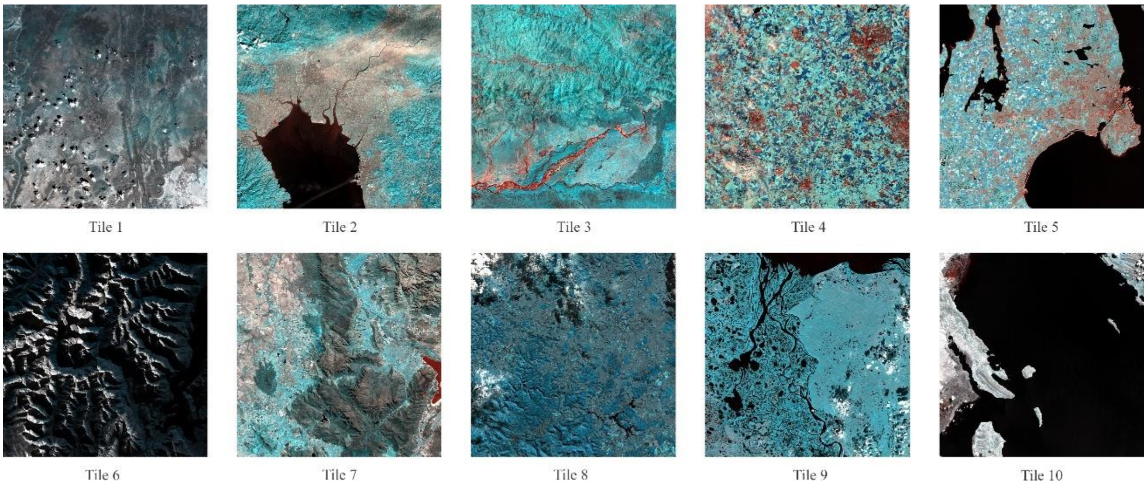


Figure 65: Depiction of all ten Sentinel-2 tile images, of size: 10,980 x 10,980 pixels, elected for the wealth of the study's investigations.