



Harokopio University of Athens
School of Environment, Geography and Applied Economics
Department of Geography

**Pixel-level fusion and classification of high spatial resolution
optical and SAR imagery**

by
Konstantinos Panos

A dissertation submitted in partial fulfillment of the requirements for the
degree of Master of Science

in
Applied Geography and Spatial Planning – Geoinformatics

Athens, 2020



Harokopio University of Athens
School of Environment, Geography and Applied Economics
Department of Geography

Supervisor

Issaak Parcharidis
Professor, Department of Geography, Harokopio University of Athens

Committee

Emmanuel Vassilakis
Assistant Professor, Department of Geology and Geoenvironment, National
and Kapodistrian University of Athens

Panagiotis Artelaris
Assistant Professor, Department of Geography, Harokopio University of
Athens

Ethics and copyright statement

I, Konstantinos Panos, hereby declare that

1. I am the copyright owner of this original work, and to the best of my knowledge, my work does not offend anyone or infringe on the copyright of any third party.
2. I accept that Library and Information Centre of Harokopio University may, without altering the content of my work, make it available in electronic form through its Digital Library, copy it in any medium and/or format, and hold more than one copy for maintenance and safety reasons.

Abstract

This work attempts to fuse optical and SAR imagery at the pixel-level with four different methods, classify the results, and evaluate them with quantitative metrics. The data-set is one multi-spectral PlanetScope image, and one single-polarization SAR ICEYE-X2 image over Dar es Salaam, Tanzania. Compares the multiplicative (MLT), Brovey transform (BT), high-pass filter addition (HPFA), and principal component analysis (PCA) techniques. For the classification process, the random forest classifier is implemented. To evaluate the fusion results, the standard deviation, entropy, and mutual information metrics are deployed. For the classification assessment, the out-of-bag (OOB), overall accuracy (OA), and Cohen's kappa (K) are used. A spatial stratified random sampling technique is adopted, for both training and testing data. The fusion, classification, and evaluation are implemented in python while the pre-processing and spatial sampling in SNAP, and QGIS. From the fusion analysis, all the fusion products show double the entropy and similar mutual information values. From the classification analysis, the MLT shows the most promising results with OA = 85.06%, and K = 0.80, while the multi-spectral has OA = 83.81%, and K = 0.79. The results confirm the hypothesis that some fusion products outperform in classification accuracy the multi-spectral.

Keywords: Fusion · Classification · Evaluation · PlanetScope · ICEYE-X2 · Dar es Salaam

Περίληψη

Η εργασία επιχειρεί να συγχωνεύσει οπτικά και SAR δεδομένα σε επίπεδο ψηφίδας με τέσσερις διαφορετικές μεθόδους συγχώνευσης, να ταξινομήσει τα αποτελέσματα τους, και να τα αξιολογήσει βάση ποσοτικών δεικτών. Το σετ δεδομένων αποτελείται από μια πολυφασματική εικόνα PlanetScope, και από μια μονής πόλωσης εικόνα SAR ICEYE-X2, όπου και οι δυο έχουν παρθεί για την πόλη Νταρ ες Σαλάμ της Τανζανίας. Συγκρίνει τεχνικές όπως η πολλαπλασιαστική μέθοδος (MLT), η μετατροπή Brovey (BT), η πρόσθεση φίλτρου υψηλού περάσματος (HPFA), και η ανάλυση κυρίων συνιστώσων (PCA). Όσον αφορά την ταξινόμηση, εφαρμόστηκε η μέθοδος τυχαίου δάσους (random forest). Για την αξιολόγηση της συγχώνευσης χρησιμοποιήθηκε η τυπική απόκλιση, εντροπία, και αμοιβαία πληροφορία (mutual information). Για την αξιολόγηση της ταξινόμησης αξιοποιήθηκε το out-of-bag (OOB), η συνολική ακρίβεια (OA), και ο Cohen's kappa (K). Υιοθετείται η μέθοδος της χωρικά τυχαίας στρωματοποιημένης δειγματοληψίας για τα δεδομένα εκπαίδευσης και επαλήθευσης. Η συγχώνευση, ταξινόμηση, και αξιολόγηση εφαρμόστηκε μέσω της python ενώ η προ-επεξεργασία των εικόνων και η δειγματοληψία εφαρμόστηκαν μέσω των SNAP, και QGIS. Από την ανάλυση της συγχώνευσης, όλα τα προϊόντα έχουν διπλάσια εντροπία σε σχέση με την πολυφασματική και παρόμοιες τιμές στον δείκτη αμοιβαίας πληροφορίας. Από την ανάλυση της ταξινόμησης, η MLT μέθοδος έχει τα πιο υποσχόμενα αποτελέσματα με $OA = 85.06\%$, και $K = 0.80$, ενώ η πολυφασματική εικόνα λαμβάνει $OA = 83.81\%$, και $K = 0.79$. Τα αποτελέσματα επαληθεύουν την αρχική υπόθεση πως κάποια από τα συγχωνευμένα προϊόντα θα επικρατούν στα ποσοτικά μέτρα αξιολόγησης έναντι του πολυφασματικού προϊόντος.

Λέξεις κλειδιά: Συγχώνευση · Ταξινόμηση · Αξιολόγηση · PlanetScope · ICEYE-X2 · Νταρ ες Σαλάμ

Contents

1	Introduction	1
2	Literature review	3
3	Preliminaries	6
3.1	Fusion	6
3.1.1	Multiplicative fusion	8
3.1.2	Brovey transform	9
3.1.3	High-pass filter addition	9
3.1.4	Principal component analysis	10
3.2	Classification	11
3.2.1	Decision trees	11
3.2.2	Random forest	12
3.3	Evaluation	13
3.3.1	Quality metrics	13
3.3.2	Classification accuracy	14
4	Materials and Methods	15
4.1	Case study	15
4.2	Data-sets	16
4.3	Methodology	17
4.3.1	Pre-processing	18
4.3.2	Spatial sampling	21
4.3.3	Pixel-level fusion and quality metrics workflow	22
4.3.4	RF and classification accuracy	24
5	Results	25
6	Discussion	45
7	Conclusions	47

Acknowledgements	49
Bibliography	50
Appendix	54

List of Figures

3.1	Fusion and classification workflows at each fusion level (adapted by Pohl & Genderen (1998)).	7
3.2	Different pixel-level fusion methods.	8
4.1	Dar es Salaam, the capital city of Tanzania, with the AOI.	16
4.2	The general methodology of fusion, classification, and evaluation processes.	17
4.3	Spatial distortions due to terrain and multi-look corrections.	19
4.4	Acceptable alignment due to the image to image georeference.	20
4.5	The final stacked three-dimensional product, with its five bands.	20
4.6	The spatial stratified random sampling method on a 5×5 grid.	21
5.1	Result of the MLT fusion workflow.	25
5.2	Zoomed result of the MLT fusion.	26
5.3	Result of the BT fusion.	26
5.4	Zoomed result of the BT fusion.	27
5.5	Differences in the narrow, Gaussian, and Sobel convolution.	27
5.6	Result of the HPFA Sobel fusion.	28
5.7	Zoomed result of the HPFA Sobel fusion.	28
5.8	Result of the PCA fusion.	29
5.9	Zoomed result of the PCA fusion.	29
5.10	Scree plot of PCA fusion.	30
5.11	Visualization of each individual PC.	31
5.12	Zoomed visualization of each individual PC.	32
5.13	Pair-plots of the original stacked image (multi-spectral and SAR bands)	33
5.14	Pair-plots of the fused bands.	34
5.15	Classification result of the multi-spectral bands.	36
5.16	Classification result of the SAR band.	37
5.17	Classification result of the MLT bands.	38
5.18	Classification result of the BT bands.	39
5.19	Classification result of the HPFA bands.	40

5.20 Classification result of the PCA bands. 41

5.21 Confusion matrix of each classified image. 43

5.22 Corresponding figure to the table 5.3. 44

Chapter 1

Introduction

Each year more and more constellations of earth observation satellites are launched into space to monitor earth's biophysical parameters, and to assess the impact of humanity on the environment. These missions generate large amounts of remote sensing data from different types of sensors ([Sajjad & Kumar, 2019](#)). The rapid growth of passive and active sensors in combination with the evergrowing availability of high-resolution imagery will enhance current processing frameworks. These updated frameworks will be able to provide additional information on scientific and commercial issues regarding climate change, natural hazards, and security. By fusing multi-sensory data, it's possible to create a value-added product with greater predictive ability than a single-sensory one. According to [Pohl & Genderen \(1998\)](#), the purpose of image fusion is to integrate multi-sensor data in order to obtain more information than can be derived from single-sensor data alone. This is based on the fact that images obtained from different sensors capture different parts of the electromagnetic spectrum, thus providing a better understanding of the objects observed within the imaged scene ([Kulkarni & Rege, 2020](#)).

Multi-spectral (or optical) and synthetic aperture radar (SAR) sensors are widely used in applications such as forest monitoring, urban feature extraction, change detection, flood monitoring, and land cover classification. Most of the time, these data-sets are used separately without considering the benefits of their combination. Multi-spectral imagery contains information on the reflective and emissive characteristics of the earth's surface features, while SAR contains information on the surface roughness, texture, and dielectric properties of natural and human-made objects ([Amarsaikhan et al., 2010](#)). The different characteristics of each sensor may provide new insights in land cover classification and urban feature extraction problems ([Lu et al., 2017](#)), which in turn can improve the decision-making ability of organizations and governments around the globe on issues regarding the environment monitoring and sustainable planning.

However, the necessity of data fusion raises some challenges regarding their evaluation process. A reliable fusion algorithm should preserve most of the mutual and beneficial information from the input images, be robust to some imperfect conditions such as misregistration and noise, and not produce any visual artifacts (Li et al., 2017). The difficulty with multi-spectral and SAR fusion evaluation is that most quality metrics are for pan-sharpening techniques, which is a form of fusion for spatial enhancement. In the pan-sharpening workflow, the objective is for the fused product to be as close, in terms of spectral information, to the original, while gaining the spatial information from the high-resolution panchromatic image. In multi-spectral and SAR cases, this objective is not the most desired outcome because spectral similarity might result in a product with no added-value. For this reason, especially in classification applications, the evaluation of a fused image is best determined by both specialized quality metrics and the classification accuracy (Abdikan et al., 2015).

This dissertation deals with the fusion of high-resolution multi-spectral and SAR imagery. The purpose of fusing these products is to enhance the spectral information of a scene, and subsequently, improve the overall classification accuracy. This work also examines the hypothesis that fusion products yield better classification results than the multi-spectral. A big emphasis is given in the evaluation strategies of fusion and classification. The literature shows there is no panacea method in pixel-level fusion, hence this study compares different fusion approaches. More specifically, the main objectives are to

- **fuse** the PlanetScope and ICEYE-X2 products with different pixel-level fusion algorithms over the capital city of Tanzania, Dar es Salaam,
- **classify** the fused products and determine which combination of fusion algorithm and classifier parameters is suited for the region, and
- **evaluate** the fusion and classification results with quantitative measurements.

Lastly, this work strives to transform into a short practical guide to specific pixel-level fusion methods in remote sensing classification applications. The field of remote sensing fusion can seem daunting to a new student or practitioner, due to the preliminaries it requires and the limited literature. Hence, this document emphasizes both the implementation and the theory behind image fusion.

Chapters 2 reviews some recent papers in multi-spectral and SAR fusion, classification, and evaluation strategies, while chapter 3 addresses the fundamental principles of pixel-level fusion, classification algorithms, and evaluation metrics. Chapter 4 analyzes the data, test site, and methodology of this work. Chapter 5 lists the results of the analysis, chapter 6 discuss potential alternative methods and suggests strategies for future improvements, and finally chapter 7 draws the main conclusions.

Chapter 2

Literature review

This chapter focuses on the recent work of satellite imagery fusion, especially on the topic of MS/SAR fusion. It reviews studies that apply fusion techniques to enhance the classification accuracy and studies that attempt to evaluate with quantitative measurements the fused products. While there is more literature to be found on these topics, I present the ones that influence my work the most, in chronological order. Some papers emphasize the classification aspect of the data fusion while others the quantitative evaluation methods.

The purpose of [Amarsaikhan et al. \(2010\)](#) research is to compare the performances of different data fusion methods in order to enhance various urban features, and subsequently, to improve the urban classification accuracy of Ulaanbaatar's city (Mongolia) using a refined Bayesian classifier. They use two images in total, one QuickBird (2.4m), and one TerraSAR-X (1m). The green, red, and near-infrared bands are chosen from the optical sensor and the intensity band of both polarization from the SAR sensor. The co-registration is done in the Gauss-Kruger map projection using multiple ground control points (GCPs) defined in a field survey, with each image having a root mean squared error (RMSE) of less than 1.5 pixels. The applied fusion techniques are wavelet transform (WT), Brovey transform (BT), Ehlers fusion (EF), and principal component analysis (PCA). While all the methods enhance the spectral and spatial features of the study area, they chose to keep the BT image due to superior quality, in terms of spatial separation. To derive texture features, contrast, entropy, and dissimilarity measures are extracted from the fused product. For the classification, they use the standard Bayesian maximum likelihood classifier (MLC) and propose a refined MLC which uses spatial thresholds defined from local knowledge. The fused imagery combined with the extracted features and the refined MLC achieve an overall classification accuracy of 90.96%. The conclusions show that multi-source information can significantly improve the interpretation and classification of urban land cover types, and a refined MLC is more advantageous compared to the standard one.

The work of [Marcello et al. \(2012\)](#) approaches the challenging topic of quality evaluation for pan-sharpened MS imagery. They compare, analyze, and evaluate the spectral and spatial effectiveness of the BT, intensity hue saturation (IHS), extended fast IHS, PCA, Mullat algorithm, Atrous algorithm, Atrous-IHS, and Atrous-PCA. The database of which they examine those methods contains GeoEye, WorldView, QuickBird, and Ikonos imagery. Their work includes objective and subjective metrics such as visual quality metrics and quantitative measurements respectively. Metrics such as correlation coefficient (CC), relative dimensionless global error (ERGAS), and universal quality index (Q) are tested for each fusion method. Their results show there is a major trade-off between maximum spatial detail and spectral degradation. Additionally, the spectral quality indicators CC and ERGAS seem reliable enough for pan-sharpening, while there are some inconsistencies with the Q index.

[Dahiya et al. \(2013\)](#) discuss the use of high-resolution images, for urban feature extraction, through pixel-level fusion methods. While they do not incorporate SAR data, only MS such as Ikonos images (4m), the algorithms for pan-sharpening are the same as the ones for MS/SAR fusion. They implement nine pixel-level fusion techniques, many of which we have already addressed in preview papers, but with some additions such as multiplicative, subtractive, and hyperspherical color space approaches. A large portion of their paper details the necessary pre and post-processing steps in order to apply the fusion methods. Their visual inspection suggests that multiplicative techniques, even though they darken the urban scene, seem to enhance objects with high reflectances such as roads and buildings. BT and PCA can improve the edges of buildings, making them sharper. They apply quality metrics on the final product and also do histogram analysis. Final remarks are that fused images enable better identification of buildings, roads, and trees, and especially the PCA and BT methods which preserve all the urban features such as textures, edges, shadows, and spectral contents.

[Abdikan et al. \(2015\)](#) investigate the state of land use classification accuracy when they fuse dual-polarized SAR data to multi-spectral. They use TerraSAR-X (8m) and RapidEye (5m) over the Gediz basin (Turkey), which composites a mix of agricultural fields and artificial structures. Adjustable MS/SAR fusion, EF, high pass filtering (HPF), and Bayesian fusion are the approaches they perform. They implement quality metrics via statistical analysis such as the bias of mean, difference in the variance, entropy, CC, and ERGAS. For classification, they apply and compare the support vector machines (SVM), random forest (RF), k-nearest neighbors (k-NN), and MLC classifiers. The RapidEye image alone gives classification accuracy up to 94.89% with the SVM-RBF kernel classifier, while the fused image with the adjustable fusion (VV polarization) gives 95.32%. Even though not all the fused methods increase the classification accuracy, the study provides promising results. It highlights the importance of a ranking system for the quality metrics, due to the superior comparable nature it can have over non-ranking systems. Single fusion techniques are inadequate to improve the quality of land use mapping, hence the need for multiple fusion methods in order to find the most satisfactory ones. They suggest that quality metrics should not be the only way of interpreting the final products, because it's possible to have low scores on the quality metrics and still yield good classification results.

[Gibril et al. \(2017\)](#) aim are to produce a LU/LC map of a tropical agricultural area in Hilir Perak (Malaysia) by examining different image fusion methods, in order to maximize the benefits of using SAR imagery. They also assess pixel-based and object-based classification methods to find the most suitable LU/LC map. For multi-spectral imagery they use Landsat-8 (15m with pan-sharpening) and for SAR the RadarSAT-2 (12.5m). They co-register both images to the UTM projection with an image-to-image technique, achieving RMSE less than 15m. The WT, BT, and EF are used as fusion mechanisms. Training data from 10 different land cover classes are extracted and classified on a pixel and object-level. For the pixel-based classification the MLC, SVM, spectral angle mapper, and layer stacking are applied while for the object-based they utilize a tree-based approach and the standard NN method. They also extract NDVI values for image segmentation purposes to enhance the object-based classification. Their results show a 74.96% classification accuracy with Landsat-8 alone combined with SVM, while the highest classification accuracy for the multi-sensor is 79.78% with layer stacking and 63.70% with BT. The tree-based approach has 93.35% accuracy and the NN method 76.33%. Their analysis indicates that a rule-based approach is more efficient in mapping LU/LC features from fused images.

The paper of [Kulkarni & Rege \(2020\)](#) addresses the necessity of fusion between MS/SAR imagery, reviews the state of the art methods for pixel-level fusion, and evaluates them with quantitative criteria. They focus on the pre-processing techniques of MS/SAR, emphasizing on the importance of a good speckle filter. Comparisons between Lee, enhanced Lee, gamma MAP and wavelet shrinkage filters which are implemented on RISAT-1 ScanSAR CRS HV (36m) imagery shows their major difference and highlights reasons to choose some filters instead of others, depending on the application. They fuse the SAR imagery with an optical ResourceSAT-2 AWiFS (56m) and use the region of Odisha (India) as their case study. They implement BT, generalized IHS, PCA, and WT with rules based on the wavelet coefficients. Quantitative evaluation is performed using an extensive list of quality metrics such as CC, ERGAS, entropy, and spatial frequency. Most of the quality metrics show poor performance for the BT method, in comparison to GIHS/PCA, and the PCA fusion result shows the least spectral angle among the rest methods. In their conclusions, they mention that all image fusion methods are specific to their data-sets due to differences in sensor geometry, polarization, frequency, and resolution. Some methods may yield better results in their quality metrics for a specific sensor while others may not.

Chapter 3

Preliminaries

This chapter presents the theoretical background of pixel-level fusion, classification, and evaluation methods. It gives a brief introduction to specific techniques and workflows, rather than a broad approach. Therefore, the reader should have a strong grasp on the fundamentals of multi-spectral and SAR theory.

3.1 Fusion

The earliest papers that apply remote sensing fusion, such as the one by [Cliche \(1985\)](#), fuse multi-spectral with panchromatic products rather than multi-spectral with SAR. Their main purpose was to create a new product with the same spectral information but with higher spatial resolution. These pansharpening methods developed rapidly after the launch of SPOT-1 in 1986 ([Meng et al., 2019](#)). With recent advancements in SAR sensors, new image fusion techniques have been developed to address the need for better classification products. These image fusion techniques classify into three categories depending on the level at which fusion takes place (see figure [3.1](#)).

Pixel-level fusion consists of merging information from different images on a pixel-by-pixel basis to improve the performance of image processing tasks such as classification or segmentation ([Solberg, 2006](#)). Low processing level translates into low complexity for the workflow, and a small loss of information. That occurs because the data fuse in their raw format, which serves to maintain most of the original information. This level has the highest computational load among the rest, and requires the images to be strictly registered prior to fusion. As classification methods, the most common are statistical and neural networks, while evaluation strategies are similar on all levels.

Feature-level fusion requires the extraction of prominent features such as contours, edges, geometries, textures, and indices in order to combine them into a single feature map ([Kulkarni & Rege, 2020](#)).

Medium processing level translates into medium to low complexity for the workflow with some loss of information, due to feature extraction.

Decision-level fusion first extracts prominent features just like the feature-level fusion, and then classifies each feature individually. The classified features combine by applying decision rules to reinforce common interpretation, and to resolve differences (Pohl & Genderen, 1998). High processing level translates into high complexity for the workflow, and further loss of information, but allows for a higher level of abstraction and implementation of sophisticated algorithms. Decision rules such as consensus theory (Benediktsson et al., 1992; Benediktsson et al., 1999) and Dempster-Shafer theory (Le Hégat-Masclé et al., 1997) are common in this workflow. Registration is less stringent, compared to the pixel-level.

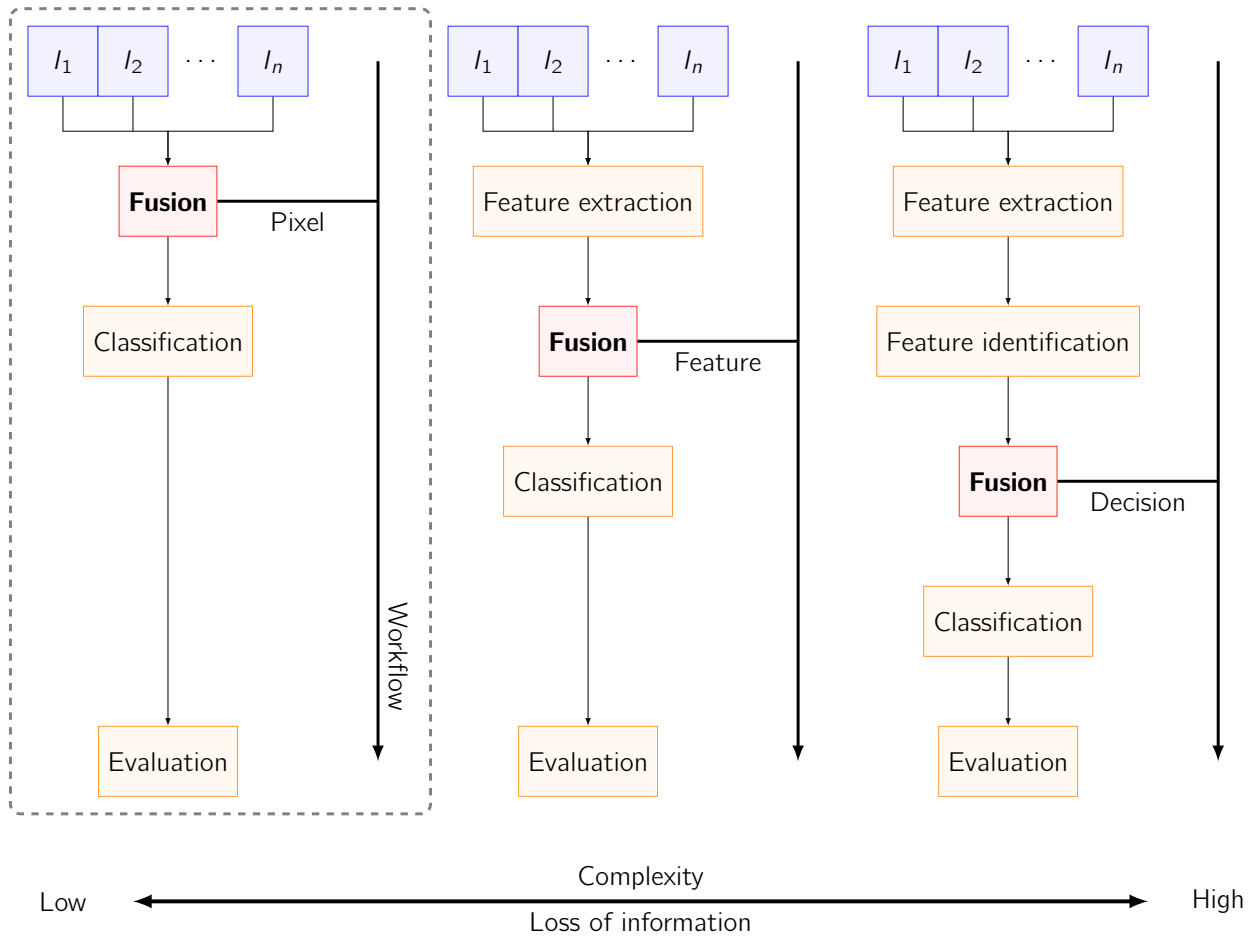


Figure 3.1: Fusion and classification workflows at each fusion level (adapted by Pohl & Genderen (1998)).

Fusion at the pixel-level can be executed by several methods, depending on the desired outcome (see figure 3.2). These methods classify into four groups, namely numerical combinations, frequency filters, statistical, and component substitution. This work applies two numerical (multiplicative and Brovey transform), one frequency filter (high-pass filter additive), and one statistical (PCA). The component

substitution incorporates inverse transformations and can apply to all previous methods. For example, the PCA belongs to the statistical group, but it can evolve to the principal component substitution (PCS) method.

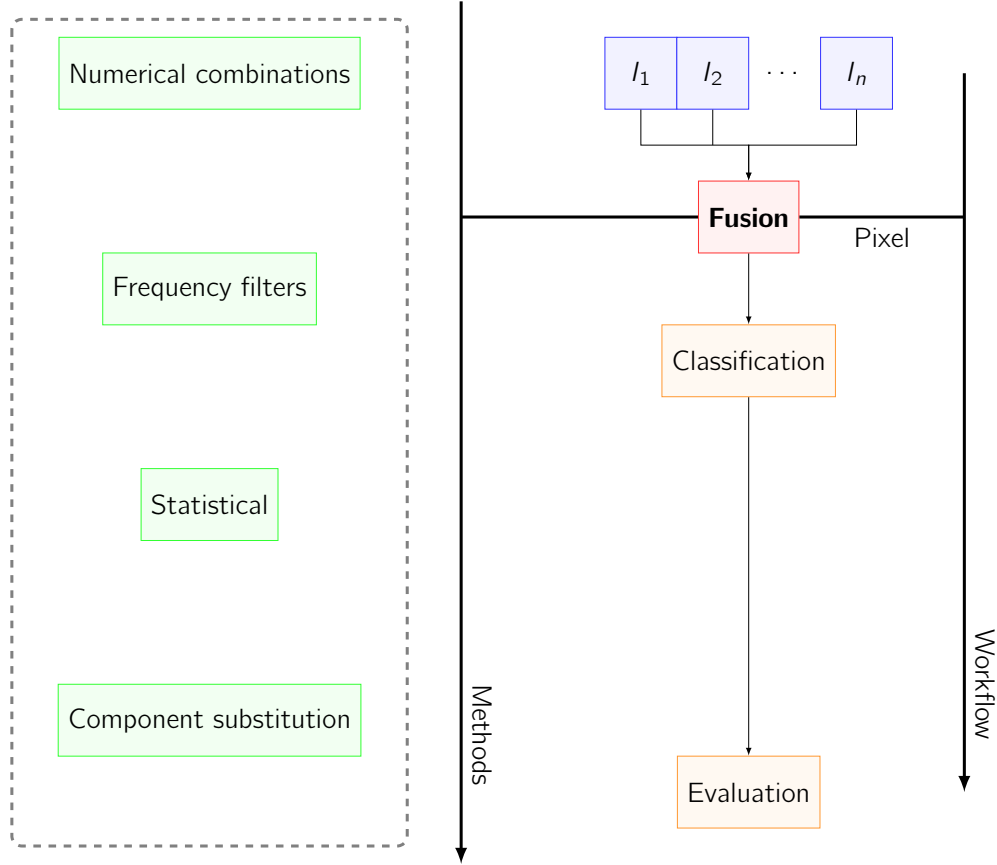


Figure 3.2: Different pixel-level fusion methods.

3.1.1 Multiplicative fusion

The multiplicative fusion (MLT) combines two equal size images by multiplying each spectral band with the SAR one (Al-Wassai et al., 2011). The fused pixel intensities $g_f(i, j)$ are given by

$$g_f(i, j) = \sqrt{g_b(i, j) \cdot g_p(i, j)} \quad (3.1)$$

where $g_b(i, j)$ the pixel intensity of the b -th spectral band and $g_p(i, j)$ the pixel intensity of the p -th SAR polarization. To counteract the increase in brightness due to multiplication, the square root of the combined products is usually taken. This method can transform into a weighted one, just by assigning weights to each band product.

3.1.2 Brovey transform

According to Ehlers et al. (2010), in Brovey transform (BT), each resampled spectral band is first divided by the sum of all resampled spectral bands and then multiplied by the intensity of the SAR band. It normalizes the multi-spectral pixel values and multiplies the result by any other desired data to add the intensity or brightness component to the fused product. It preserves the relative spectral contributions of each pixel while injecting the intensity of the corresponding SAR band (Marcello et al., 2012; Tu et al., 2001). As stated by Canty (2014), the corrected pixel intensities are given by

$$g_f(i, j) = \frac{g_b(i, j)}{\sum_k g_{bk}(i, j)} \cdot g_p(i, j) \quad (3.2)$$

where k the total number of multi-spectral bands. This method also works for pan-sharpening purposes, by replacing the $g_p(i, j)$ with the panchromatic band. Since the recent advancements and popularity of SAR imagery, more and more papers use BT for optical and SAR fusion.

3.1.3 High-pass filter addition

The high-pass filter addition (HPFA) was first introduced by Schowengerdt (1980), as a pan-sharpening method (like the BT) for the Landsat MSS data-set. The idea is that high-frequency information (derived with a high-pass filter from the SAR image) is incorporated into the multi-spectral image. The HPFA method uses either a narrow 3×3 or a wide 5×5 filter (Gangkofner et al., 2007). The general mathematical form of this fusion is given by

$$g_f(i, j) = g_b(i, j) + \gamma \cdot \text{freq}(g_p(i, j)) \quad (3.3)$$

where γ a scaling factor and $\text{freq}(\cdot)$ the high frequency information of $g_p(i, j)$. The amount of influence the SAR image has to the multi-spectral depends on the γ factor. It starts at $\gamma = 1$ and depending on the application, the researcher adjusts it accordingly. To offset the increase in brightness for $\gamma \gg 1$, just like in MLT fusion, the square root of the combined product is taken or it's divided by the number of fused products.

According to the application, there are several filter to chose from. Because there is no de facto filter for spectral enhancment and classification applications, this study compares four different filters to find the most suited one. The first are typical narrow and wide filters, like the followings

$$\text{conv}(3 \times 3) = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \text{ and } \text{conv}(5 \times 5) = \begin{pmatrix} -1 & -1 & -1 & -1 & -1 \\ -1 & 1 & 2 & 1 & -1 \\ -1 & 2 & 4 & 2 & -1 \\ -1 & 1 & 2 & 1 & -1 \\ -1 & -1 & -1 & -1 & -1 \end{pmatrix} \quad (3.4)$$

where the only difference in their structure is that the narrow filter abruptly transitions from the outer edges to the center, while the wide filter implements a smoother transition. The third filter is a Gaussian high-pass filter (known as Gaussian blur), which in fact is a low-pass filter, but it transforms into a high-pass one when it's subtracted from the original multi-spectral image.

The formula of a Gaussian filter in two dimensions is

$$L_p(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (3.5)$$

where $L_p(x, y)$ the low-pass information of p -th polarization, x the distance from the origin in the horizontal axis, y the distance from the origin in the vertical axis, and σ the standard deviation of the Gaussian distribution (Shapiro & Stockman, 2000). The x and y values represent how narrow or wide the convolution will be, which they depend on the σ value.

The forth filter is the Sobel operator, which is used in image processing, particularly within edge detection algorithms (Jin-Yu et al., 2009). The operator uses two 3×3 filters to calculate approximations of the vertical and horizontal derivatives. If G_x the vertical derivative approximations and G_y the horizontal, the computations for each one are

$$G_x = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} * g_p(i, j) \text{ and } G_y = \begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} * g_p(i, j) \quad (3.6)$$

where the symbol $*$ indicates the convolution operation. The resulting gradient approximations are combined to give the gradient magnitude $\sqrt{G_x^2 + G_y^2}$.

3.1.4 Principal component analysis

Principal component analysis (PCA) is a statistical method useful for image encoding, compression, enhancement, de-correlation, change detection, dimensionality reduction, and fusion. It transforms a multivariate data-set of intercorrelated variables into a data-set of new uncorrelated linear combinations of the original variables (Richards & Jia, 1999).

Citing Mudrova & Procházka (2005), the PCA can be described as a transform of a given set input bands with the same length L formed in the n -dimensional vector $b = [b_1, b_2, \dots, b_n]^T$ into a new vector g_f according to

$$g_f = A(b - m_b) \quad (3.7)$$

where m_b is the vector of mean values of all input bands defined by $m_b = \mathbb{E}[b]$ and matrix A is determined by the covariance matrix C_b . The evaluation of the C_b matrix is possible according to

$$C_b = \mathbb{E}[(b - m_b)(b - m_b)^T] = \frac{1}{L} \sum_k b_k b_k^T - m_b m_b^T \quad (3.8)$$

Because the rows of A are orthonormal, it's possible to invert the PCA results according to the relation

$$b = A^T g_f + m_b \quad (3.9)$$

The application of this method in the image fusion workflow implements in two ways. The first one is to apply the PCA to a stacked image, where it has collocated multi-spectral and SAR bands. The second is to apply the PCA to the multi-spectral bands, replace the component with the highest explained variance with the SAR band, and then apply the inverse PCA method to bring the data-set to its original form (Pohl & Genderen, 2016). This version is known as principal component substitution (PCS). Both methods are used in fusion workflows, but in this case, where the two data-sets are of equal spatial resolution, and the only needed information is the spectral, the first version of PCA applies.

3.2 Classification

This section discusses the basic idea behind decision trees and random forest classifiers, and how they apply to the remote sensing classification problems. Tree-based methods are simple, and useful for interpretation, but are not near as competitive as some of the best supervised learning algorithms such as SVM, MLC, and ANNs. These methods are used in regression and classification problems, except in the case of classification, they are known as decision trees. Because they do not yield the same results as some machine learning algorithms, approaches like random forest involve producing multiple trees that are then combined to generate a single consensus prediction. These types of methods are known as ensemble methods.

3.2.1 Decision trees

Decision tree classifiers have not been used widely by the remote sensing community for land cover classification despite their non-parametric nature and their attractive properties of simplicity and computational efficiency in handling abnormal, heterogeneous, and noisy data (Ghose et al., 2010). Trees are composed of a root node (containing all data), a set of internal nodes (splits), and a set of terminal nodes (leaves). Each node of the decision tree structure makes a binary decision that separates either one class or some of the classes from the remaining classes. The process executes by moving down the tree until it reaches the leaf node, also known as a top-down approach (Xu et al., 2005). The task of growing a decision tree is quite similar to growing a regression tree. The only difference is that a decision tree uses the Gini index or entropy as a binary split criterion.

The Gini index is defined by (James et al., 2013)

$$G = \sum_k \hat{p}_{mk}(1 - \hat{p}_{mk}) \quad (3.10)$$

where \hat{p}_{mk} represents the proportion of training observations in the m -th region that is from the k -th class. Its a measure of total variance across all classes. When all the \hat{p}_{mk} are close to zero, then the Gini index also takes a small value. The Gini index refers to the measure of node purity, where a small value indicates that a node contains observations predominantly from a single class.

An alternative to the Gini index is entropy, given by (James et al., 2013)

$$H = - \sum_k \hat{p}_{mk} \log_2 \hat{p}_{mk} \quad (3.11)$$

The entropy takes a value near zero if the \hat{p}_{mk} 's are all near zero or near one. Hence, like the Gini index, the entropy will take small values if the m th node is pure.

An ideal leaf node is one that contains only records of the same class. In practice reaching this leaf node may require an excessive number of splits that are costly. The decision to continue splitting is based on either the Gini index or entropy.

3.2.2 Random forest

The random forest classifier grows many decision trees to achieve higher classification accuracy. This idea is known as bagging, which is short for "bootstrap aggregating". It uses multiple versions of the same classifier to make an ultimate decision by taking a plurality vote among the predictors. In the case of random forest, each tree votes for a given pixel a class, and then the forest chooses the final class of the pixel that has the most votes over all the trees in the forest (James et al., 2013).

Each tree is trained on a bootstrapped sample of the original training data-set and searches only across a randomly selected subset of the input bands to determine a split (Gislason et al., 2006). There is a straightforward way to estimate the classification error of a bagged model, without the need to perform cross-validation or the validation set approach. Because trees are repeatedly fit to bootstrapped subsets of the train data-set, on average each bagged tree makes use of around two-thirds of the data-set (James et al., 2013). The remaining one-third of the samples that are not used to fit a given bagged tree refers to the out-of-bag (OOB) data-set, which measures the classification error, known as the OOB score.

The random forest can measure each input band's importance, with the use of the OOB data-set. Each band is randomly permuted, and the permuted OOB cases are sent down the tree again. Subtracting the number of correctly classified cases using permuted data from the number of correctly classified

cases using non-permuted data gives the importance value of the band. These values differ for each tree, however, the average of each value over all trees in the forest gives an estimation over the importance score for each band (Lowe & Kulkarni, 2015). The random forest handles high-dimensional data while being much more computationally efficient. As each tree is only using a portion of the input sample bands, the algorithm is considerably lighter than conventional bagging with a comparable tree-type classifier (Gislason et al., 2006).

To fully understand how the classifier works, this section ends by quoting an example from the work of Lowe & Kulkarni (2015) which compares the random forest to a judicial court:

For example, in judicial court a robber may be on trial for stealing. The jury members will classify the robber as guilty or innocent. They have been "trained" by a random subset of every account of robbery in history. This subset contains all robberies a single juror has seen or heard about. Having been "trained" by other accounts of robbery, each jury member will take different variables of the trial into consideration. One member might take into account the value of the object stolen, the victim of the robbery, and the robber's age while another jury member might classify the robber's guilt based on gender, his/her religion, and the robber's age. At the end of the trial all jury members vote on the classification of the robber. In this limited example, the jury is the forest, each member is a tree, the robber is the case to be classified, and the pieces of evidence of the trial are the features to be used for classification.

3.3 Evaluation

This section discusses the quantitative measures for evaluating the pixel-level fusion and classification results. Quantitative and qualitative analysis (visual inspection) is always vital in assessing the output of an algorithm. Visual analysis alone contains subjective factors and may be influenced by personal preference. Due to its subjective nature, it can not determine the reliability of the results.

3.3.1 Quality metrics

The quality metrics typically group into two categories, one that needs a reference image and one that doesn't. A similar attribute of both groups is that most of the quantitative indicators are created explicitly to address the pan-sharpening fusion (Zhang, 2008). The main goal of the pan-sharpening fusion is to enhance the fused image with spatial information while keeping the spectral one as close to the original image as possible. With this in mind, most of the indicators consider a change in the pixel value a negative effect (e.g., RMSE, SAM, ERGAS, SSIM). To compensate for that obstacle, this study examines three indicators that primarily detect information richness. These indicators are standard deviation, entropy, and mutual information. Only the mutual information requires a reference image, but does not consider a change in the spectral information of a pixel as a negative result.

The standard deviation measures the contrast in the fused band. When the value is high, it means that the fused band has high contrast, thus larger variation in the intensity of its pixels. Entropy, the same metric mentioned in the classification section, gives an estimate of the image's information. A high value of entropy shows that the image is rich in information, thus making it "superior" to its predecessor (Yin & Jiang, 2013).

The mutual information is an indicator of the distance between the joint probability distribution of the original and fused band (Jagalingam & Hegde, 2015). This metric calculates according to

$$MI(B, P) = \sum_{i=1}^{|B|} \sum_{j=1}^{|P|} \frac{|B_i \cap P_j|}{n} \log \frac{n|B_i \cap P_j|}{|B_i||P_j|} \quad (3.12)$$

where $|B_i|$ the sample pixels of band B and $|P_j|$ the sample pixels of polarization P . A high value of mutual information indicates greater distance between these two distributions.

3.3.2 Classification accuracy

The classification accuracy metrics used in this study are mainly three, the overall accuracy (OA), Cohen's kappa coefficient (K), and OBB score from the random forest classifier. While these measurements ensure an objective view of the classification process, it's important to understand that they are calculated based on the training and testing data-sets. Poor selection of training and testing areas may result in unsatisfactory results, thus concluding that the classification process produces false results. To be precise about these metrics, one has to have as testing data the entire AOI. If there is a "ground truth" for all the pixels in the image, the metrics can be more representative, hence reflecting the true land cover type. Nevertheless, the testing data which does not span the entire AOI can also give reliable results.

The OA is a by-product of the confusion matrix. The confusion matrix is square, where its columns represent the "ground truth" values, and the rows represent the classifier's predictions. It's calculated by summing the number of correctly classified pixels, and then dividing by the total number of pixels. The correctly classified values locate along the upper-left to the lower-right diagonal of the confusion matrix. The Cohen's kappa coefficient measures the agreement between the "ground truth" pixel and the predicted pixel, using the following expression

$$K = \frac{P(c) - P(e)}{1 - P(e)} \quad (3.13)$$

where $P(c)$ the proportion of times the pixels agree, and $P(e)$ the proportion of times the pixels would agree by chance. When there is no agreement other than that which would be expected by chance, then K is zero, else when there is a total agreement, K is one (Carletta, 1996).

Chapter 4

Materials and Methods

4.1 Case study

This section details the urban characteristics of Dar es Salaam (see figure [4.1](#)). The primary reasons for choosing this city as an area of interest are SAR data availability and its expansion rate. Without the SAR availability from ICEYE, this study could not be possible. It also facilitates the analysis of the fusion, and classification algorithms in an ever-changing urban environment, which blends densely populated areas with peri-urban ones.

Africa today is the world's least urbanized continent, with only 43% of its population living in urban areas. But high birthrates and rising internal migration mean African cities are entering an era of unprecedented growth. According to U.N. projections, the continent will have 18 cities with more than five million people by 2030. Dar es Salaam, currently Africa's fifth-most populous city, ranks second behind Kampala in neighboring Uganda. It's projected to grow from six million people today to 13.4 million by 2035, crossing the "megacity" threshold of 10 million people sometime before 2030 ([Rosen, 2019](#)).

Such population growth comes with extreme land cover changes, which must be addressed for a sustainable urban plan to take place. According to [Kombe \(2005\)](#), Tanzania is one of the countries in the sub-Saharan region with the highest proportion of its urban residents living in informal settlements, which are vast labyrinthine neighborhoods of flimsily built concrete houses. Between 50% and 80% of its urban population live in informal settlements. In Dar es Salaam, over 70% of the estimated 2.4 million inhabitants live in such houses. These informal settlements constitute the major and most vibrant sector, shaping the urban growth of the city ([Kombe & Kreibich, 2000](#)).

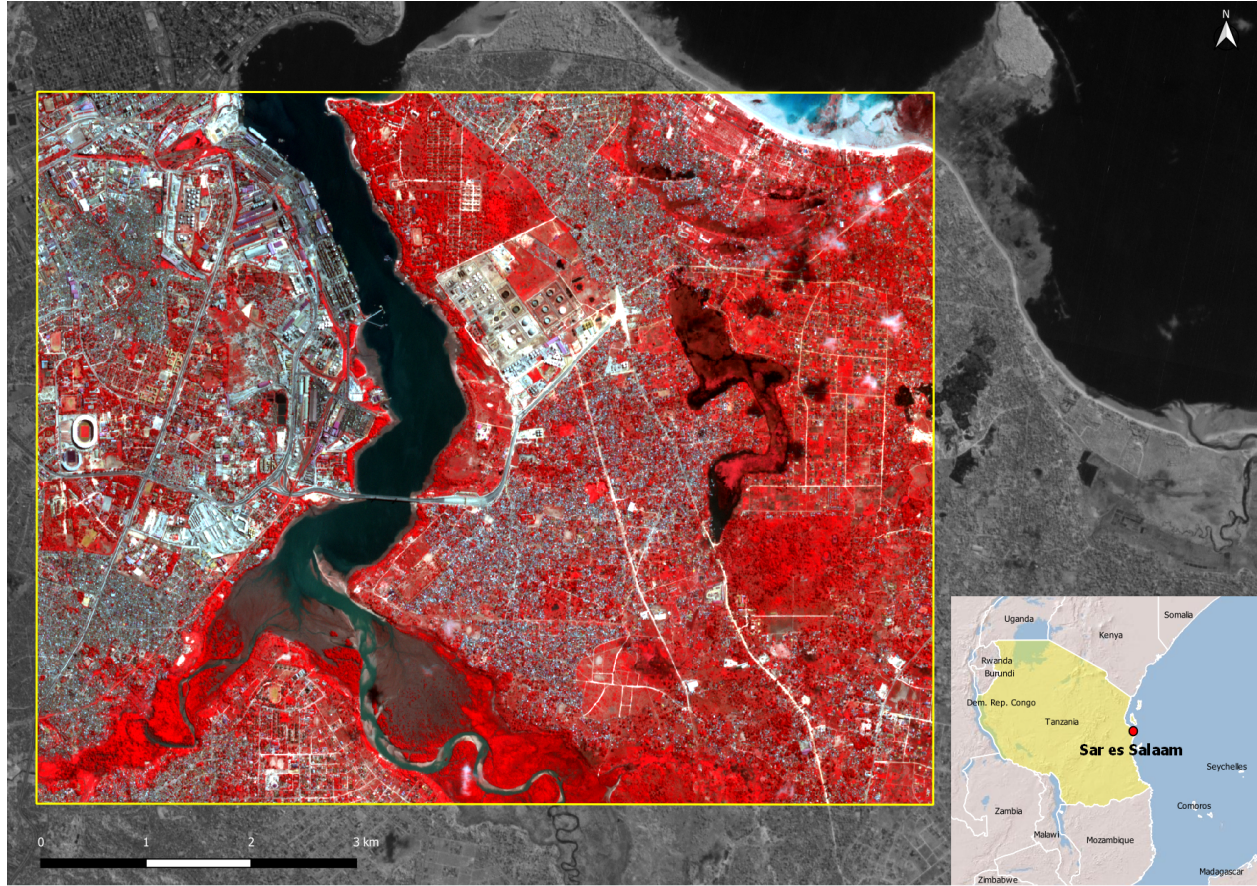


Figure 4.1: Dar es Salaam, the capital city of Tanzania, with the AOI.

4.2 Data-sets

This work uses only two high-resolution images. The first is a multi-spectral PlanetScope surface reflectance (SR) image and the second is a SAR ICEYE-X2 (GRD) single-polarization image (see table 4.1). The SAR image is downloaded from the ICEYE's website as a sample product, and the multi-spectral is obtained with the support of the NSRF research project "Laertes". The prerequisite for the pixel-level fusion is to have two products with similar spatial resolution and acquisition date, which in this case both images meet the condition. At the time, there hasn't been any other similar attempt (in the literature) to fuse these two specific products.

Satellite	Product	Acquisition date	Acquisition time	Bands	Resolution (m)
PlanetScope	OrthoScene (SR)	18/6/2019	07:21:51	R, G, B, NIR	$\approx 3^*$
ICEYE-X2	Stripmap (GRD)	23/5/2019	19:52:17	VV	

Table 4.1: Basic information for each sensor. *While the document for each sensor specifies a spatial resolution of 3 meters, in reality it's in the range of [2.9-3] meters.

The ICEYE constellation, as of April 2020, consists of three 85kg X-band SAR microsatellites, and it's expected to grow to 18 satellites by the end of 2021. The service is based on a completely new satellite and sensor design, leveraging recent advances in various technologies. Each satellite is in a sun-synchronous orbit with a nominal 18-22 day ground track repeat cycle (depending on satellite) with 15 imaging orbits per day (ICEYE, 2020). Planet currently operates the Dove constellation that compromises of satellites only $10 \times 10 \times 30$ centimeters in size, and the recently retired RapidEye constellation, which after operating for 11 years outperforming its design life, retired in March 2020. In total, Planet has flown on 26 successful launches and deployed over 351 satellites in orbit (Planet, 2020).

4.3 Methodology

This section details all the necessary steps that take to fuse and classify the two data-sets. Figure 4.2 shows in a flowchart the general methodology of this work.

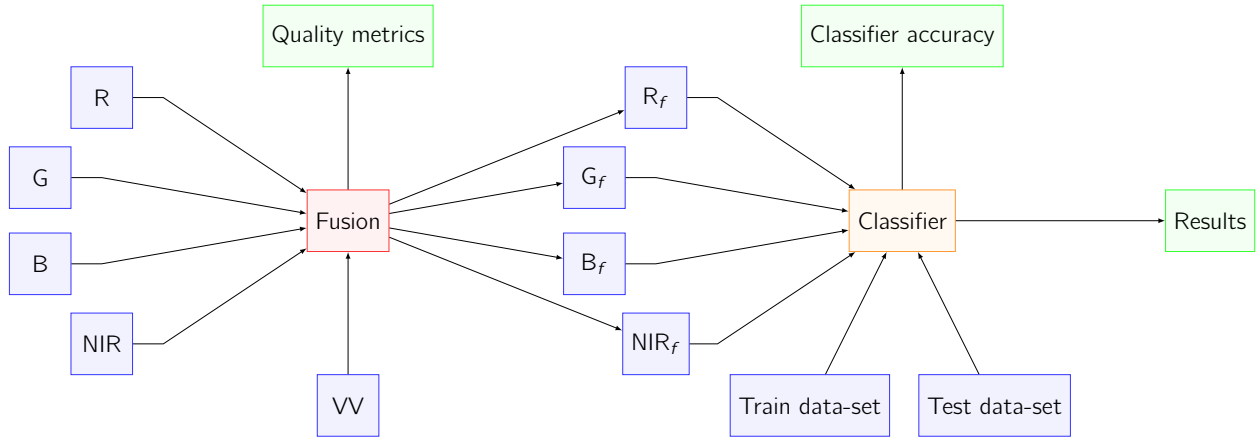


Figure 4.2: The general methodology of fusion, classification, and evaluation processes.

This methodology implies that the pre-processing stage for the multi-spectral (R, G, B, NIR) and single-polarization (VV) bands has already been completed. After the pre-processing, these images are imported into a fusion mechanism that stacks them, implements the fusion algorithm, calculates the quality metrics of the fused products, and then outputs the new bands (R_f , G_f , B_f , NIR_f) with their accompanied quality metric results. These four fused bands are then imported into the trained random forest classifier, and through the testing data-set, it produces the final classified layer and calculates the classification accuracy. In the end, it's possible to assess the best fused and classified results, based on the quantitative measures and visual inspection of the classified layer.

The fusion, classification, and calculation of all quantitative measures are implemented through the programming language python, while the whole pre-processing workflow in ESA's sentinel application platform (SNAP). The sample selection and visualization are carried out by the open-source QGIS. Each processing stage is analyzed in detail in the following sections with its accompanied python code,

which is in the appendix. This study uses free and open-source software, a conscious choice that was taken from the beginning.

4.3.1 Pre-processing

The pre-processing stage is not only necessary for the whole workflow to function, but it plays a crucial role in the fusion process. Choosing the wrong speckle filter or making errors in the image-to-image georeference can lead to a different input layer for the fusion and classifier algorithms.

The PlanetScope image, especially the SR, derives from the standard analytic product (in radiance) which is processed to top of atmosphere reflectance and then atmospherically corrected to bottom of atmosphere reflectance for all four orthorectified bands ([Collinson & Wilson, 2017](#)). These radiometrically corrected bands are clipped into the desired AOI, the downtown and port area of Dar es Salaam, and then reprojected into a common coordinate reference system (CRS: WGS 84/UTM zone 37S). The AOI is a single rectangular polygon, in order to clip both the multi-spectral and SAR imagery into the same coordinates. The subset process is important not only to reduce the computational time of the algorithms, but also to make easier the collocation, resampling, and rasterization process of the train and test data-sets. At this stage, the pre-processing of the PlanetScope images is finished. The final step, which is to collocate and resample into a common grid with the SAR image, is only possible when the ICEYE-X2 image also undergoes some mandatory changes.

The ICEYE-X2 image is a GRD product, which is orbital corrected and multi-looked, so there is no need for further improvements in those areas. ICEYE-X2 GRD products are not calibrated by default. The objective of SAR calibration is to provide imagery in which the pixel values can directly relate to the radar backscatter of the scene ([Miranda & Meadows, 2015](#)). The first step is to calibrate the image and to choose the appropriate speckle filter. Due to the coherent processing of the backscattered signal, there is some corruption in the SAR images produced by the multiplicative speckle noise (also known as "salt and pepper" pattern). To effectively reduce the speckle noise, a filter applies that suppresses the noise in homogeneous regions, preserve or improve the edges of the image, and improve the visual appearance. However, there is always a trade-off between these three expectations, since there is no single filter that can improve all the features simultaneously ([Kulkarni & Rege, 2020](#)). While there are many speckle filters to choose from, only a few are suitable for enhancing the edges of the urban features. A comparison took place between Lee, refined Lee, Gamma MAP, and Frost filters. After visual inspection, the Gamma MAP filter (size of 3×3 and one number of looks) is the most suitable one for this application.

The next phase of SAR pre-processing is the terrain correction (or geometric correction). Due to topographical variations of a scene and the tilt of the satellite sensor, there are some distortions ([Filipponi, 2019](#)). It implements the range doppler terrain correction operator, which uses the range doppler orthorectification method ([Small & Schubert, 2008](#)) for geocoding SAR images from single

2D raster radar geometry. For the operator to derive the precise geolocation information, it uses an external digital elevation model (DEM). The spatial resolution of the DEM must be on a similar scale to the input image for it to produce the best results. A DEM of such spatial resolution is not available (3×3 meters), hence this paper goes for the SRTM 30×30 meters. This difference in the spatial resolution will affect the output image and eventually have an impact on the fusion process, but not to the point of completely deluding the results. Figure 4.3 shows an example of spatial distortion due to the terrain and multi-look corrections. Such errors are bound to happen due to the nature of SAR geometry.

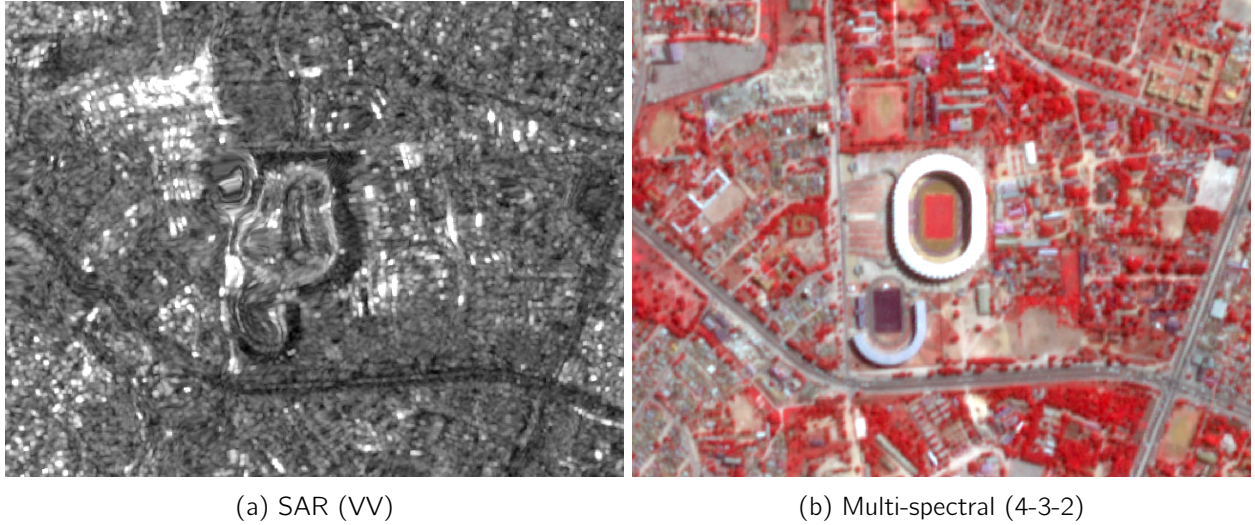


Figure 4.3: Spatial distortions due to terrain and multi-look corrections.

This process also reprojects the corrected image into the same CRS as before. The last two steps are to clip the image into the same multi-spectral AOI and convert the raw amplitude values into decibel by applying a logarithmic transformation.

Before the collocation, the image-to-image georeferencing process of the two products with multiple ground control points (GCPs) begins. These GCPs are spatially homogeneous distributed, to compensate for the misalignments and distortions. The final RMSE is around 9 meters, which accounts for 3 pixels. For a pixel-level fusion technique, a degree of 3 pixels misalignment is high, nonetheless, it's acceptable and works in practice. Figure 4.4 shows an example of a good alignment between the two products.

At this stage, the collocation between the multi-spectral and SAR imagery proceeds. Collocation allows spatially overlapping products to stack into a single one. The process requires the pixel values of the slave product to resample into the spatial resolution of the master product. The master multi-spectral has a spatial resolution of 3×3 meters, while the slave SAR falls short by a few centimeters (2.98×2.98 meters). The collocation process completes, and the stacked product (see figure 4.5) with its five bands is ready to become the input layer for the fusion algorithms.

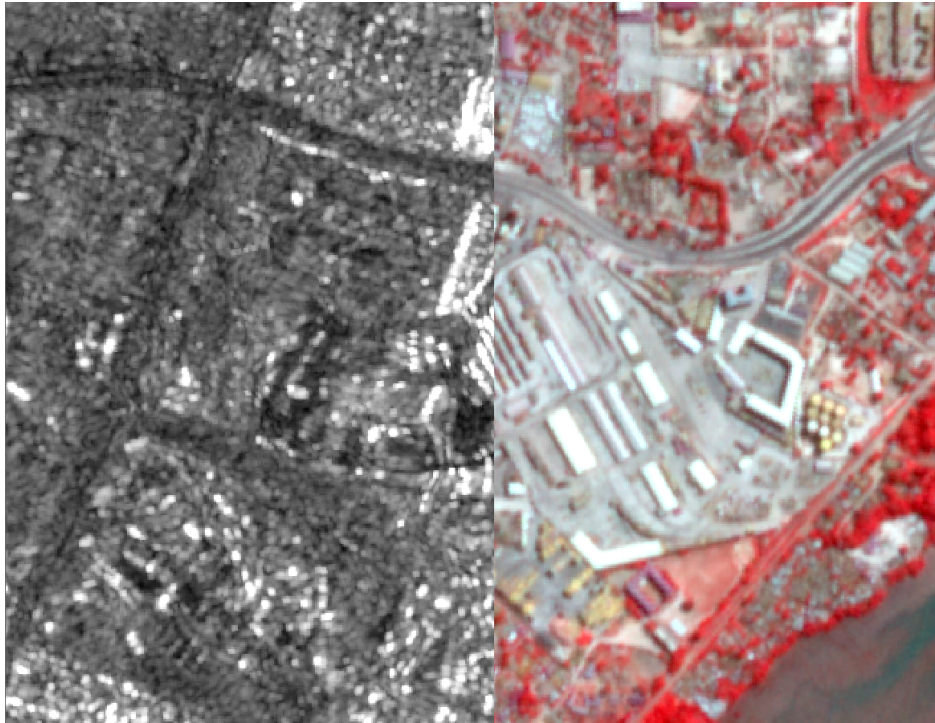


Figure 4.4: Acceptable alignment due to the image to image georeference.

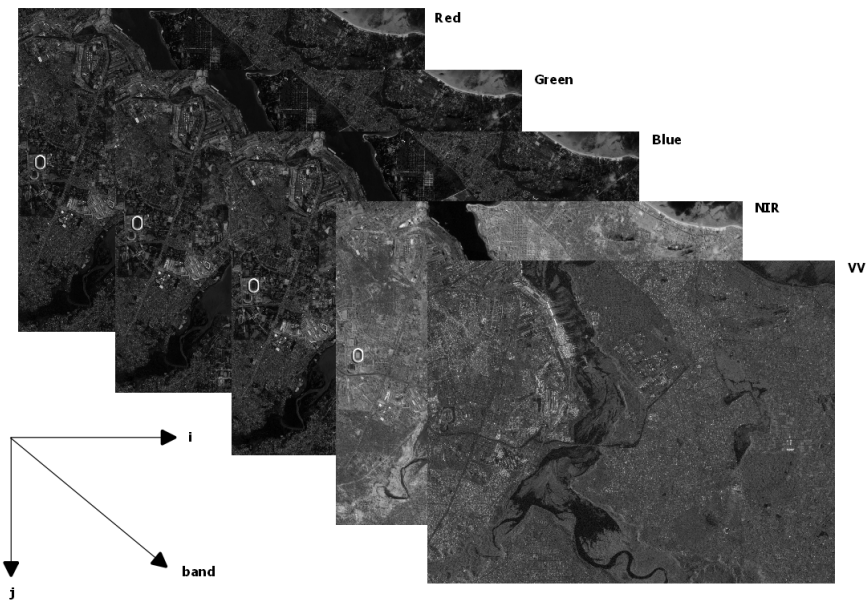


Figure 4.5: The final stacked three-dimensional product, with its five bands.

4.3.2 Spatial sampling

Before implementing the fusion algorithms, there is one final step, to create the training and testing sample data-sets for the random forest classifier. For the classifier to function, there needs to be a split in the sample data-set to avoid overfitting the model. A spatial stratified random sampling technique is adopted (5×5 grid) for both the training and testing data. This technique ensures that each training tile will have one sample polygon of each class (when possible), and the testing tile will have three. It may look like a one to three ratio, but in reality, it's one to six, due to the deliberately bigger size of the testing samples. It ensures that the random forest classifier will not overfit the training data, thus having the ability to generalize on the testing one. At this point, it's important to highlight the five classes, which are inspired by the Corine LC/LU program ([Kosztra et al., 2017](#)).

- **Artificial surface** contains urban fabric, roads, industrial, commercial, transport units, mines, dumpsites, construction sites, and artificial non-agricultural vegetated areas.
- **Semi-natural grassland** contains shrub and herbaceous vegetation associations, natural grassland, moors and heathland, sclerophyllous vegetation, and transitional woodlands.
- **Forest and urban green** contains broad-leaved, coniferous, mixed forest, and parks inside the urban fabric.
- **Wetland** class contains inland and coastal wetlands.
- **Water body** class contains inland and marine waters.

For the test data, the chosen sample areas are intentionally hard to classify, in order to have reliable classification accuracy. Figure 4.6 highlights the differences in size and density of the sample polygons.

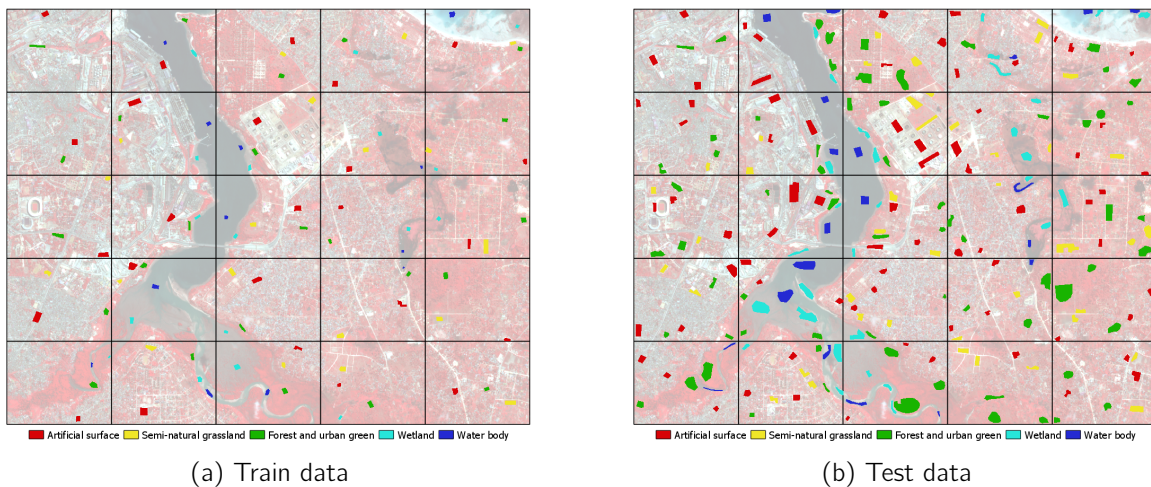


Figure 4.6: The spatial stratified random sampling method on a 5×5 grid.

After digitizing the sample polygons, it's necessary to rasterize them for the classifier. The workflow of rasterization can be described in four steps (see appendix p. 54):

- read the training and testing sample data-set with the stack image,
- save the layer, x-size, y-size, projection, geo-transformation, and driver values into variables,
- create the raster layer, using the `gdal` library and the abovementioned variables,
- repeat the workflow for training and testing data.

After repeating the process for both data-sets, it imports the resulting raster files into the random forest classifier. Prior to the classification, the implementation of fusion comes first.

4.3.3 Pixel-level fusion and quality metrics workflow

Because the whole pixel-level fusion workflow runs in python, this section describes the step by step the implementation of the fusion methods discussed in chapter 3. Before that, it mentions the usage of each library in this workflow.

- `numpy` let the whole workflow to function, due to it's fundamental nature to scientific computing in python. Most of the other libraries depend on the `numpy` functions. In short, it allows for fast computation between arrays.
- `pandas` allow for the creation of pair-plots. It enables the process to construct a data frame structure, which is easier to manipulate for the making of such a plot.
- From `osgeo` imports the `ogr`, `gdal`, and `gdal_array` functions. They are used to read and write the raster/vector files.
- `matplotlib.pyplot` and `seaborn` give access to easy visualization tools in order to produce the result layers, confusion matrixes, and pair-plots.
- From `scipy` imports the `ndimage` function, a useful tool to apply all the different filters for the HPFA fusion.
- From `sklearn.decomposition` imports the PCA function. To avoid writing the whole PCA process from start, this package simplifies the implementation process.
- From `sklearn.ensemble` imports the `RandomForestClassifier` function, which is responsible for the training and predicting the final image layer.
- From `sklearn.metrics` imports the `mutual_info_score` function which calculates one quality metric of the fusion process. It also imports the `accuracy_score`, `cohen_kappa_score`, and `confusion_matrix` functions where they are used to calculate the classification accuracy.
- From `skimage.measure` imports `shannon_entropy` function, another quality metric.
- `time` library keeps track of each process computation time. This work uses it for the timing of the image prediction process.

There is a procedure each fusion algorithm needs to complete, except for PCA, which has some additional steps. This procedure reads the stacked image and both train and test samples, writes the image into a new array, and splits the array into two sub-arrays, one for multi-spectral bands and one for the SAR (see appendix p. 55). For the PCA, due to restrictions from the library (requires for the

data to be in a `[:, band]` format), it needs to do additional steps (see appendix p. 55). Then follows the implementation of each fusion algorithm.

For the MLT algorithm (see appendix p. 56), the function takes as inputs the two sub-arrays that were created before and return the fused array, which is always of equal size with the multi-spectral. It creates a new empty array the length of the optical input data, applies the equation 3.1 for each band, and then saves it.

For the BT algorithm (see appendix p. 56), the function takes the same inputs and returns the same type of arrays as the MLT. It creates a new empty array as before, applies the equation 3.2 for each band, and saves it.

For the HPFA algorithm (see appendix p. 56), the functions take the same inputs and return the same type of arrays as the MLT and BT fusions. The functions calculate four different fusion arrays to find the most optimum one. The narrow filter function creates the 3×3 convolution, applies it to the SAR image to extract the high frequency information, creates a new empty array, implements the equation 3.3 for each band, and writes it into the array. The process for the wide filter function is the same, but with different filter size. The Gaussian function first calculates the Gaussian blur (with $\sigma = 3$, the equivalent of a narrow size filter), and subtracts the filter from the SAR band. Creates a new empty array, applies the equation 3.3 for each band, and saves it. For the Sobel operator, it generates the two derivative approximation filters, implements both of them to the SAR band, and calculates the total Sobel operator.

For the PCA algorithm (see appendix p. 57), the function creates the PCA model with its parameters (number of dimensions) and then fit the model to the transposed input data.

Next step is to implement these functions with the correct arguments (see appendix p. 58). The code in the appendix refers to the BT fusion, but it's the same for MLT and HPFA. The PCA is different due to the transpose functions.

After implementing the fusion algorithms, it's possible to calculate their quality metrics. This process applies universally to all fusion algorithms, even to the PCA (see appendix p. 58). It calculates the entropy for both the multi-spectral and fused bands, the mutual information for the fused bands (but uses the original ones to calculate the distance), and the standard deviation for both the optical and fused bands.

The fusion workflow is complete, but to visualize the results, the fused and original bands need to undergo a distribution normalization, due to restrictions posed by the `matplotlib.pyplot` library. To visualize more than one band, the library requires them to be normalized in `[0, 1]` range. The code in appendix p. 59 does exactly that. Creates two new empty arrays to save the flatten results.

Then normalizes both stack image bands by flattening and applying the min-max scaler. Reshapes the outputs into their original form, creates a minimum and maximum variable for stretching the histogram, and finally masks the values above and below the chosen minimum and maximum values. Also, besides the fused bands visualization, its useful to produce the pair-plots for the original multi-spectral image and the PCA (see appendix p. 60).

4.3.4 RF and classification accuracy

This section analyses the classification workflow with the evaluation metrics. The masking process is important to separate the training and testing areas for the classifier (see appendix p. 60). The two sample arrays contain pixel values ranging from 0 → no specified class to 5 → water body. All the pixels that have values above zero are the usable training and testing samples. It assigns to four new variables, the training data, and testing data while masking any pixel value that is below zero.

Then the random forest classifier is created, and its parameters are tuned (see appendix p. 61). The first parameter to set up is the number of trees. According to [Liaw & Wiener \(2002\)](#), a large number of trees will provide more stable results for the band's importance metric. But there is a certain threshold that when reached, the OA and OOB score values stop becoming larger. For this application, 300 trees provide the best result. The second parameter is the split criterion. For every tree selects the entropy split criterion due to better OA and K result. The third parameter is the minimum samples of leaves. This parameter ensures that the model will try to generalize outliers and noisy data. The `random_state` and `n_jobs` parameters ensure repeatability and faster calculations (`n_jobs = -1` lets parallel computation to take place). After the classification tuning, it fits the training data to optimize for the testing data. Calculation of the OOB, OA, K, and band importance follows.

At this stage, the confusion matrix is generated, visualized, and saved in the disk (see appendix p. 61). The last steps are to predict the classes for the entire image and generate the probability layers (see appendix p. 62). An additional metric of how the classifier is affected by the fusion process is to compare the entire classified area of only the multi-spectral bands with all the classified fusion bands (see appendix p. 62).

The whole workflow was also done with a standardized stacked data-set to confirm that no significant changes exist between the original and standardized one. The resulted standardized layers had worse quality metrics and classification accuracies, thus the process is implemented with the raw data.

Chapter 5

Results

This section outlines the fusion and classification results. Figures 5.1, 5.2 compare the MLT fusion with the original image. Both layers are set to a 4-3-2 combination for easier interpretability. While it doesn't look that much different from the original, it's visible that it has inherited a spectral distortion from the SAR band. Because each multi-spectral pixel is simply multiplied by its corresponding SAR, it has allowed for some noise to pass through the final image, but also some separability, especially near the estuary on the south.

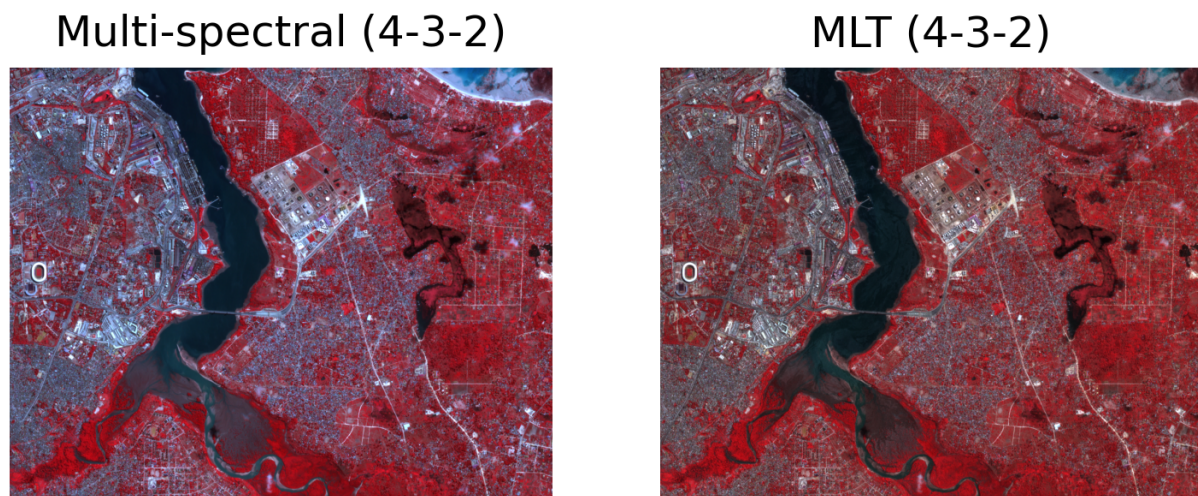
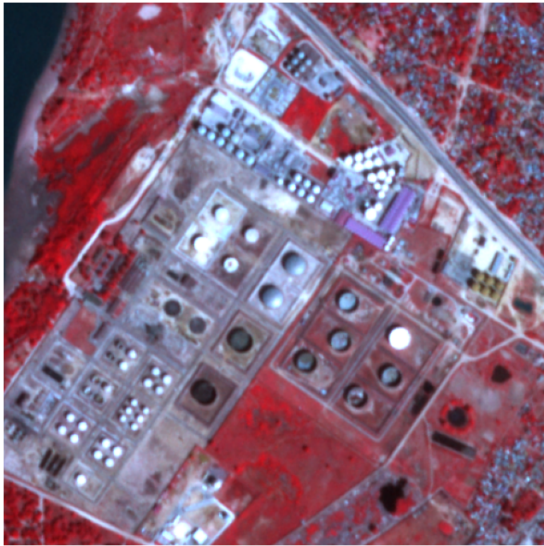


Figure 5.1: Result of the MLT fusion workflow.

Multi-spectral (4-3-2)



MLT (4-3-2)

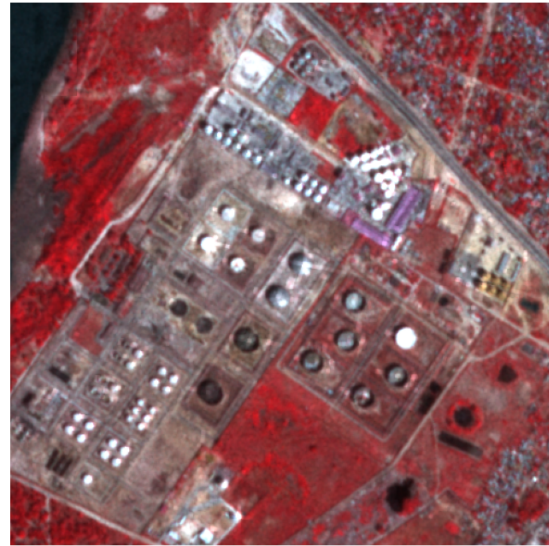


Figure 5.2: Zoomed result of the MLT fusion.

Figures 5.3, 5.4 shows the BT fusion results, which are also compared with the original. In this comparison, it's much easier to distinguish the difference in the spectral information. The normalization process of the BT algorithm changes the original colors and transfers more of the SAR information to the fused image. That is evident in the zoomed figure, where the urban edges of the SAR have transformed into a similar type of edges in the fused image.

Multi-spectral (4-3-2)

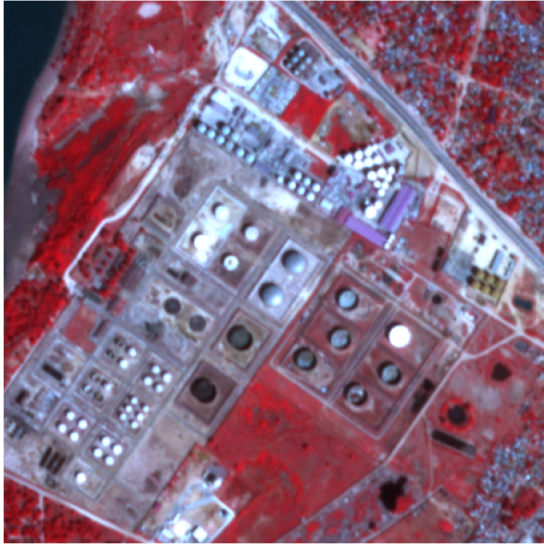


BT (4-3-2)



Figure 5.3: Result of the BT fusion.

Multi-spectral (4-3-2)



BT (4-3-2)

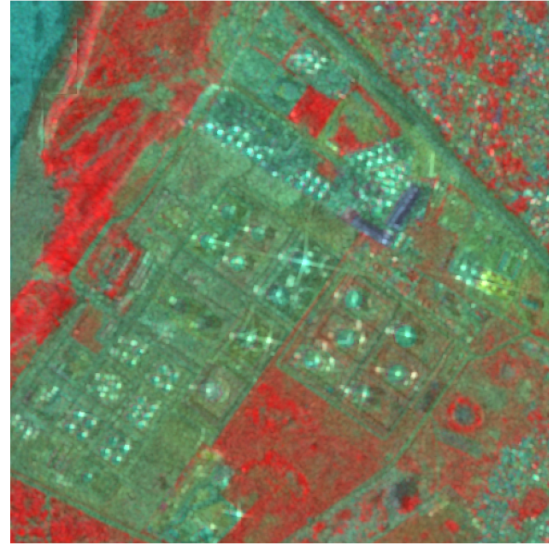


Figure 5.4: Zoomed result of the BT fusion.

Before detailing the HPFA fusion result, the differences in each filter are presented. Figure 5.5 shows on the upper left the original SAR image, and on the other sub-figures, the extracted SAR information of the narrow, Gaussian, and Sobel filter.

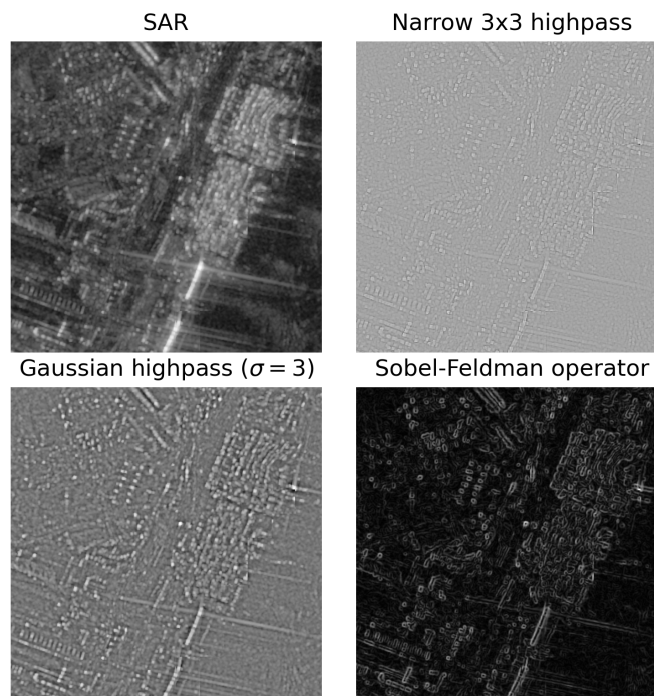


Figure 5.5: Differences in the narrow, Gaussian, and Sobel convolution.

The wide filter is excluded due to its similar nature to the Gaussian. Based on the OA and K metrics, and also on visual inspection, the most optimal filter for this application is the Sobel operator with $\gamma = 10$. Figures 5.6, 5.7 highlight the difference between the multi-spectral and fused Sobel image. This result is the most subtle because of the similarity with its predecessor. Visually there is no much difference, probably because of the low γ value. That's not to say the result is unsatisfactory, just that it's not a visually obvious one.

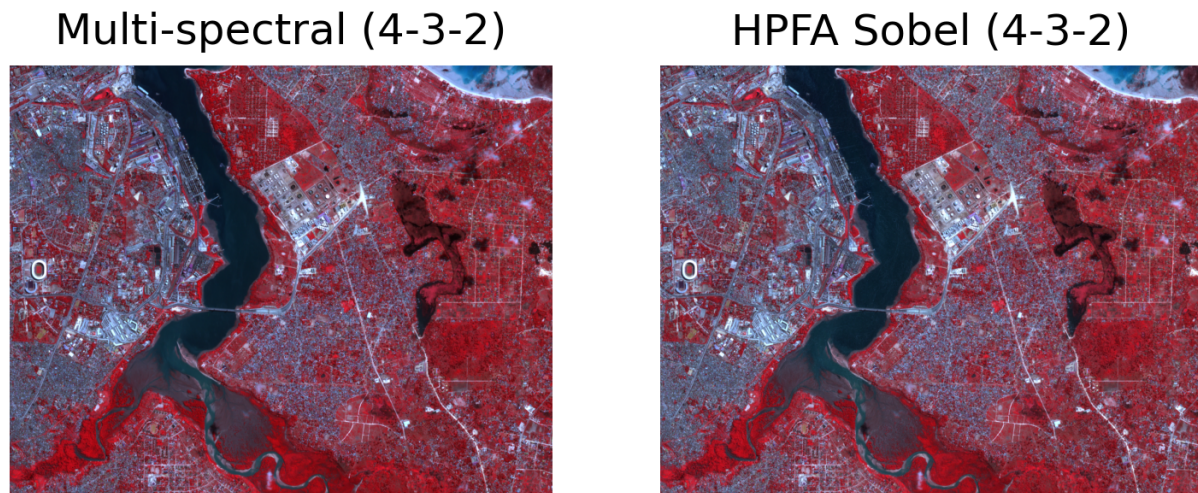


Figure 5.6: Result of the HPFA Sobel fusion.

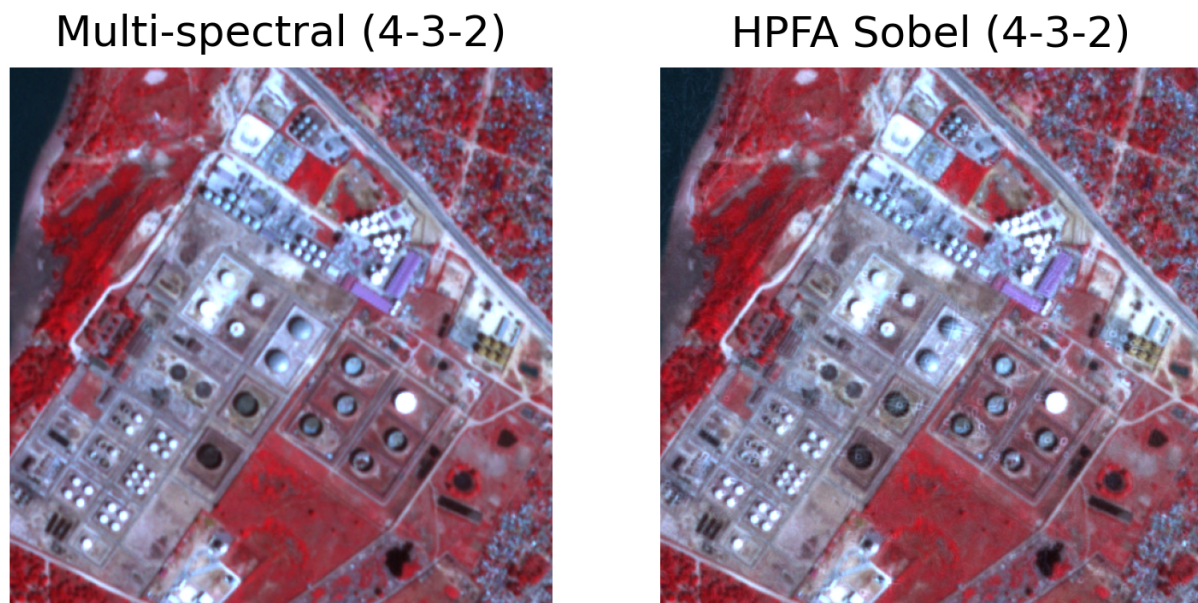


Figure 5.7: Zoomed result of the HPFA Sobel fusion.

In figures 5.8, 5.9 the PCA fusion result is shown. It has the most notable visual and spectral differences. The PCA combination, instead of being the usual 4-3-2, is changed to 3-2-1 in order to include the

first band, which represents the first PC. It's important to include the first PC instead of the fourth, due to the amount of the explained variance that the first PC contains. Figure 5.10 shows clearly that the first PC contains more than 90% of the explained variance, the second PC explains around 8% of the variance, and the third and fourth PCs explain less than 1% combined.

Multi-spectral (4-3-2)



PCA (3-2-1)

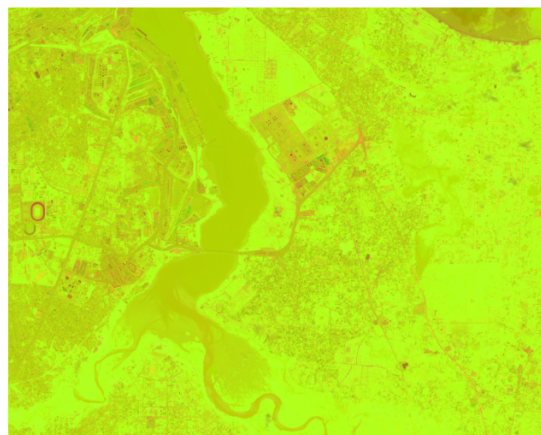
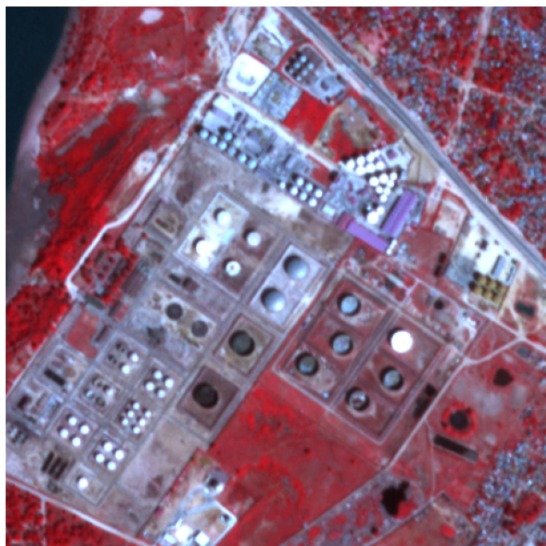


Figure 5.8: Result of the PCA fusion.

Multi-spectral (4-3-2)



PCA (3-2-1)

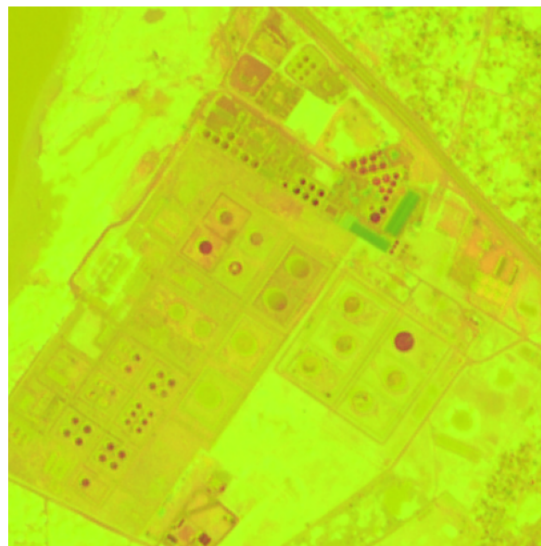


Figure 5.9: Zoomed result of the PCA fusion.

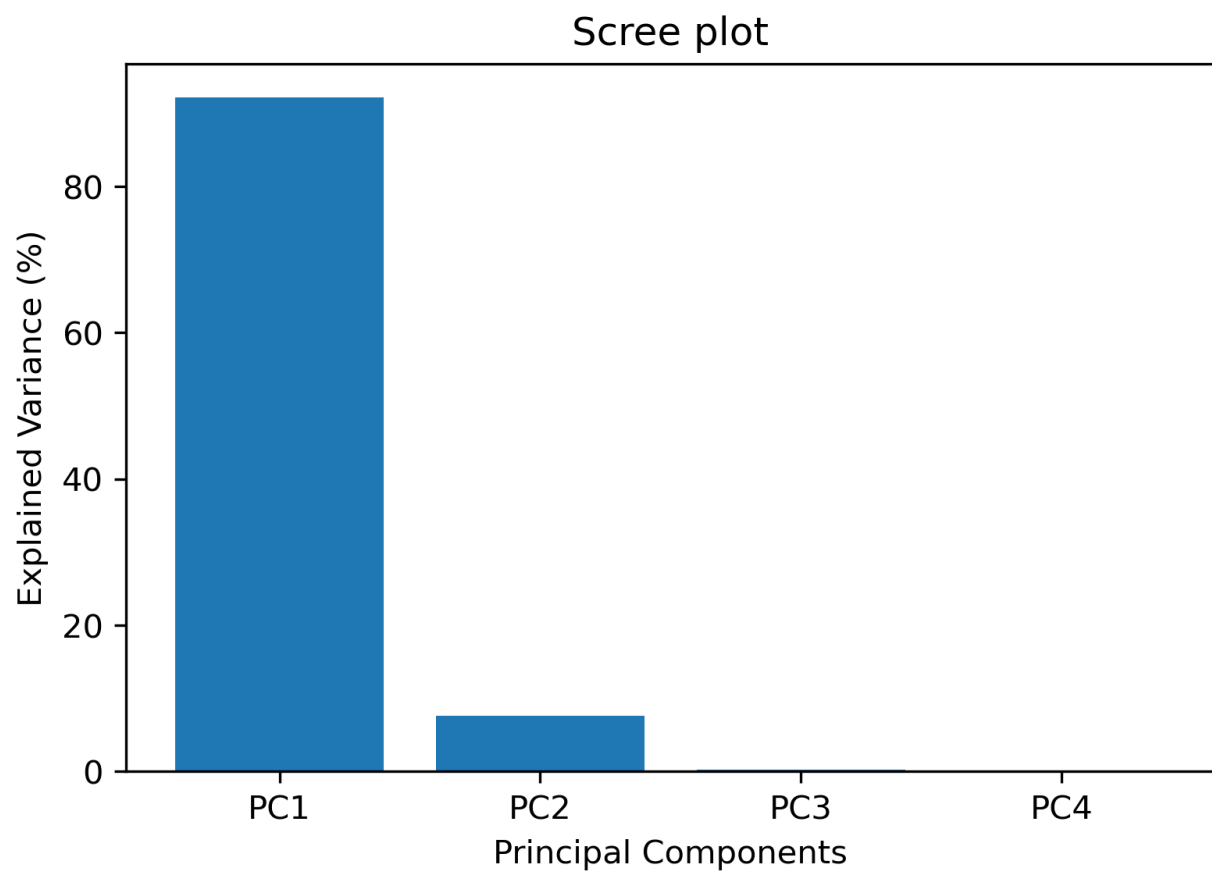


Figure 5.10: Scree plot of PCA fusion.

The visualization of such result is shown in figures 5.11, 5.12. Because the first PC has the most information from the rest of the PCs, visually it seems a bit flat. The second and third PCs are the ones that show more prominent features, which are easy to detect in the zoomed version.

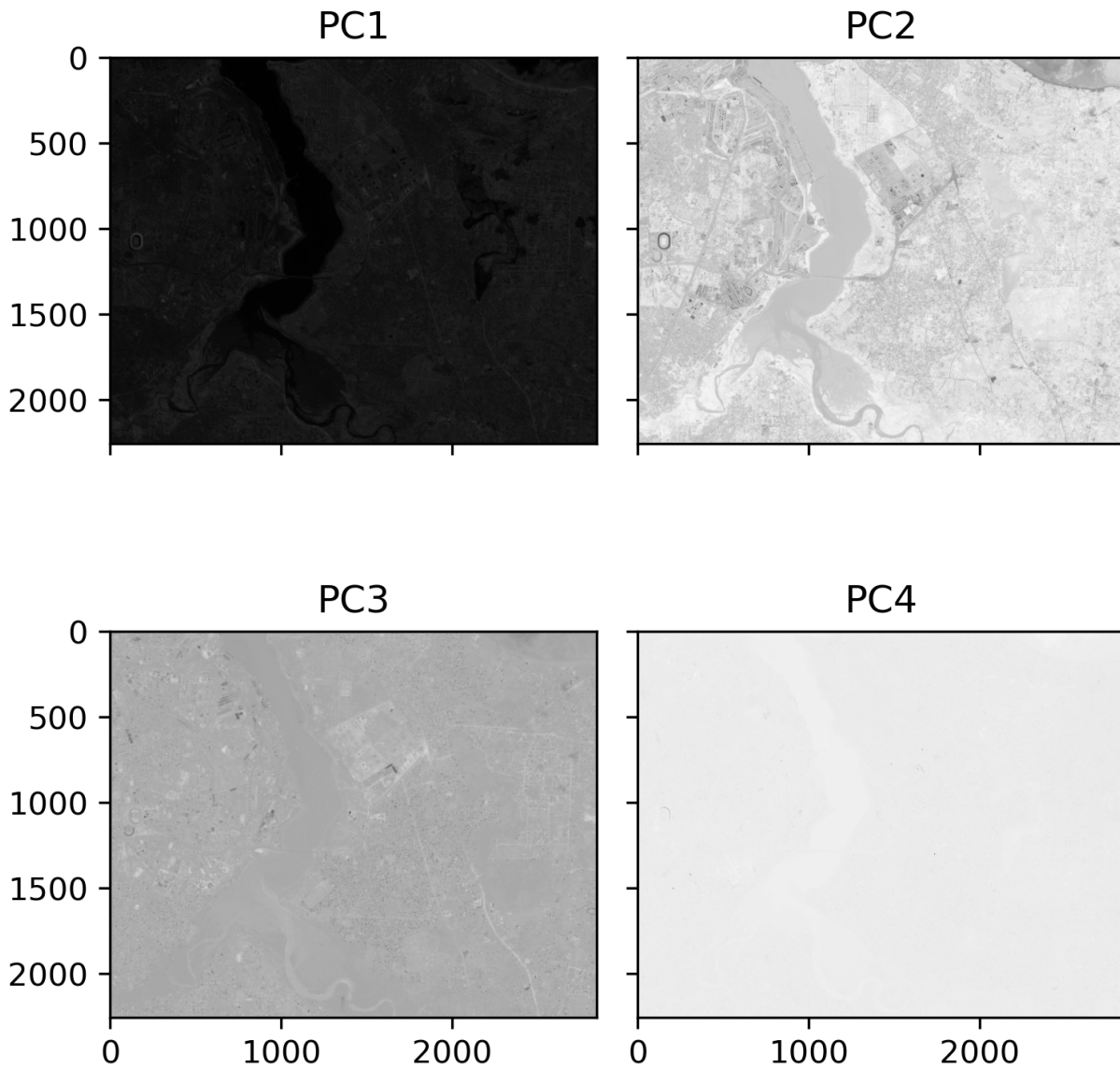


Figure 5.11: Visualization of each individual PC.

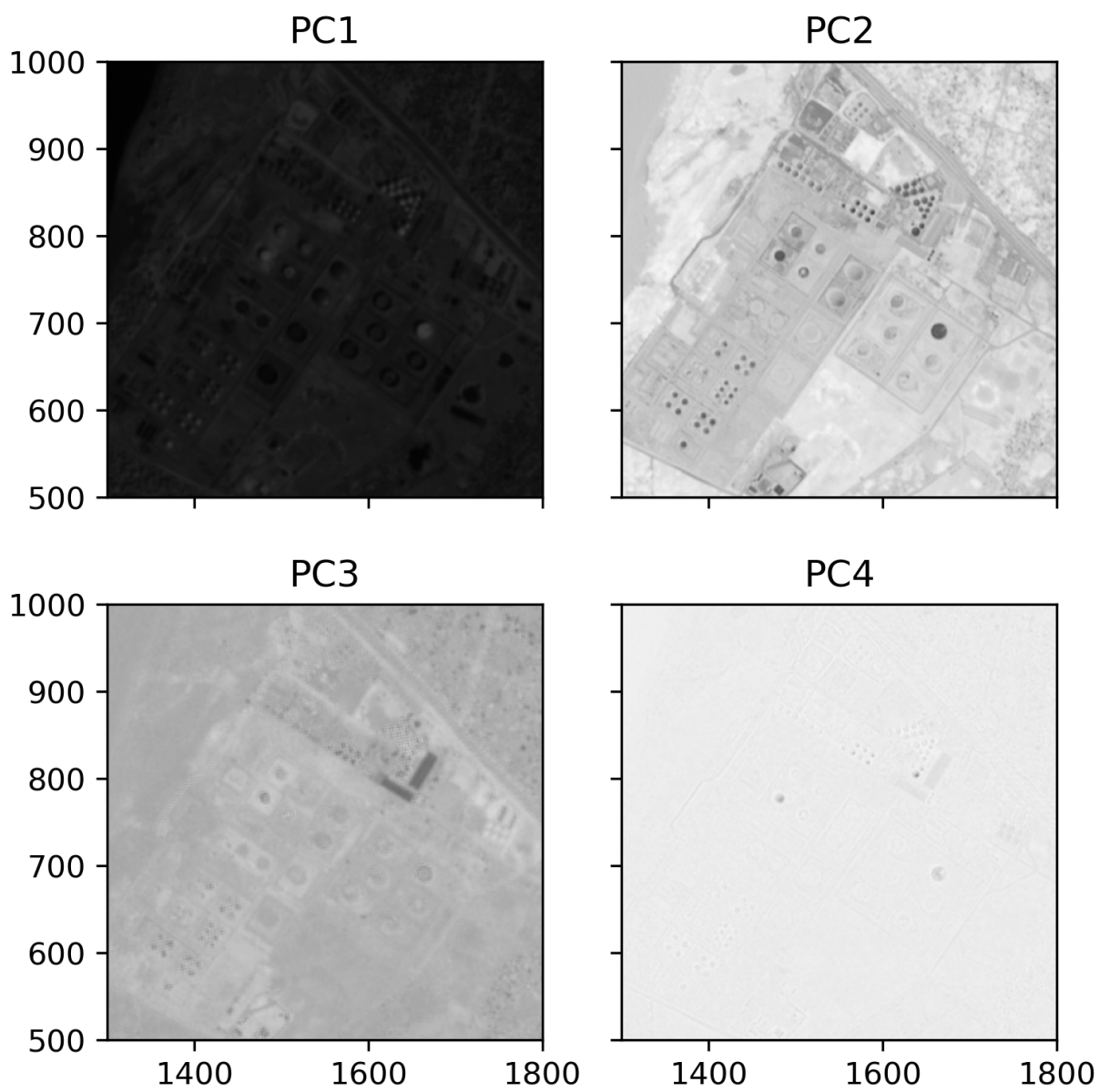


Figure 5.12: Zoomed visualization of each individual PC.

Another piece of information that it's possible to derive from the PCA analysis is the de-correlation results. These are shown in figure 5.13 for the original bands and in figure 5.14 for the PCs. The original bands, especially the red, green, and blue, are extremely correlated compared to the NIR and VV. In contrast, the fused bands have no correlation between them, and their distributions are normal.

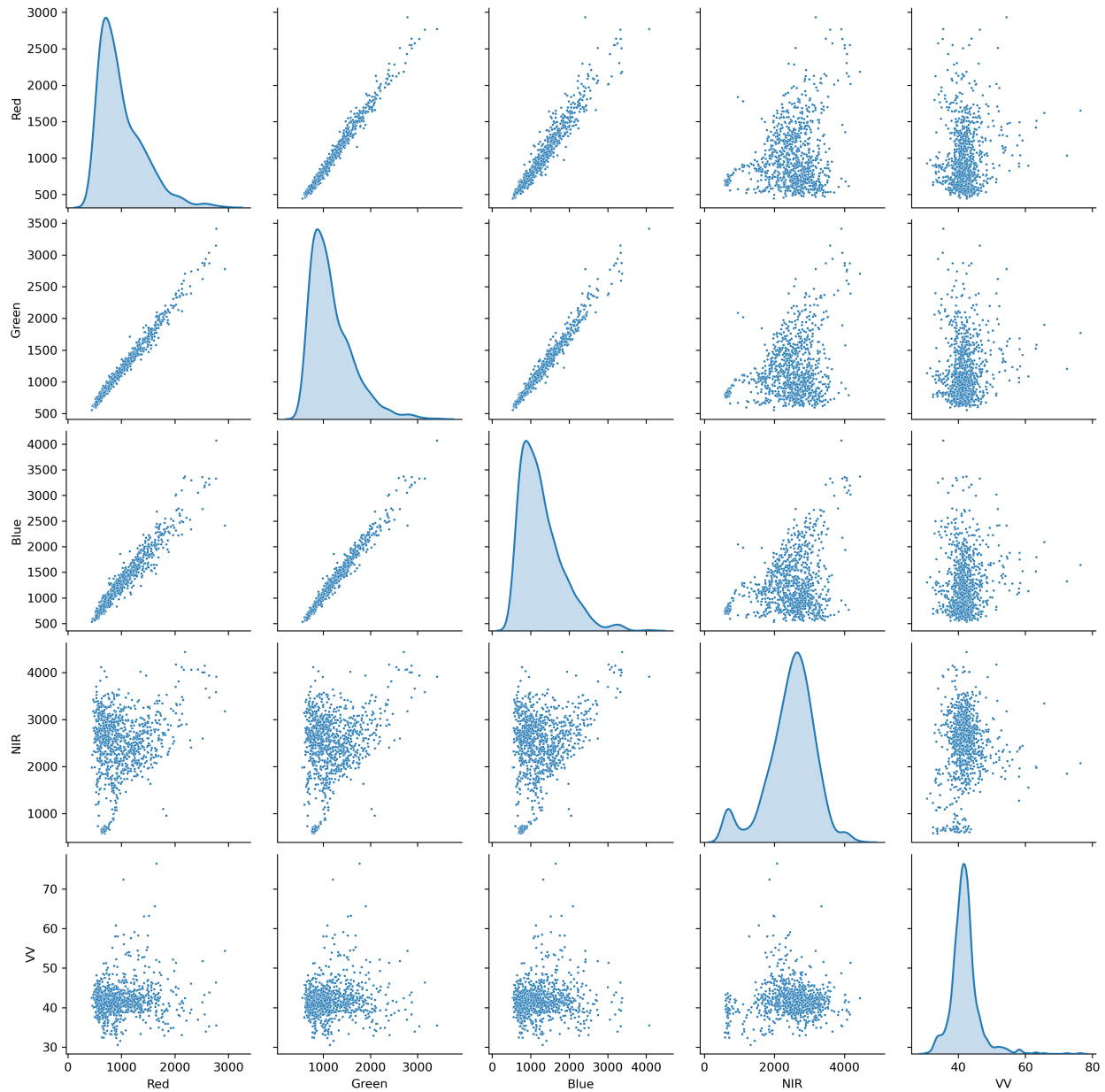


Figure 5.13: Pair-plots of the original stacked image (multi-spectral and SAR bands)

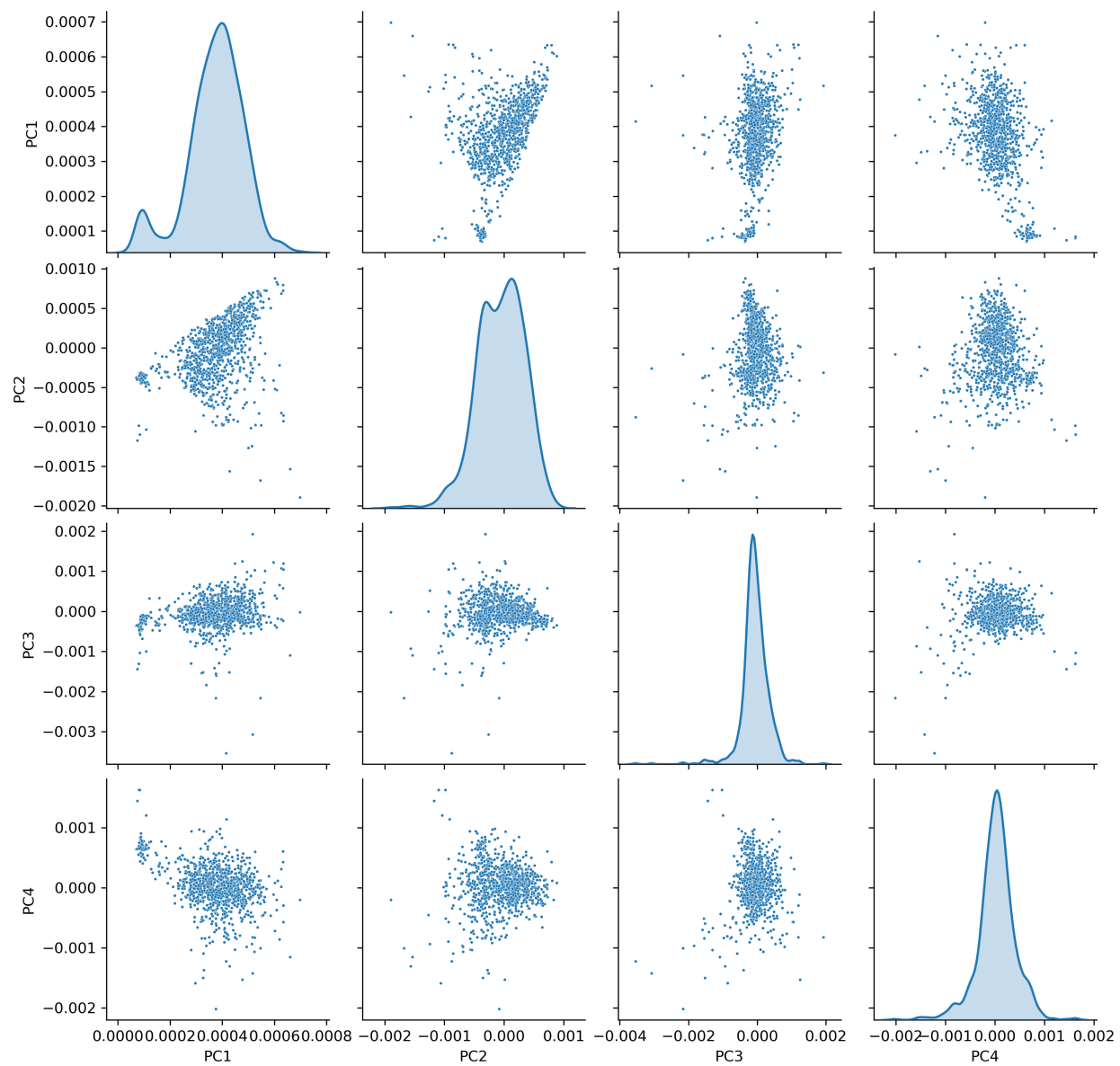


Figure 5.14: Pair-plots of the fused bands.

After the visual inspection, the evaluation of the quality metrics, namely the standard deviation, entropy, and mutual information, takes place (see table 5.1). The standard deviation values of the fused bands compared to the spectral show the difference in approach each method follows. The PCA due to the de-correlation process has for all bands zero values, the MLT with its square root minimizes all the band values, the BT band values are low because of the normalization process, and HPFA Sobel has similar values with the spectral bands. For different γ , the standard deviation is much larger than the original.

The entropy indicator shows all the fused bands having double the information of their corresponding spectral ones. That does not mean the classification results have to be two times better than the original (in some cases is worse). It just highlights the spectral information and noise of SAR imagery was injected into the original. It's a good indication the fused bands are double in entropy.

The mutual information metric shows that in general, all the bands have the same distribution distance from the multi-spectral. The bigger the distance, the less information they share, thus the bands share less information with the original, but that doesn't mean they have less information in general. They probably have the additional information from the SAR image that multi-spectral bands aren't "aware" of and hence don't share.

Metric	Band	Multi-spectral	MLT	BT	HPFA (Sobel)	PCA
Standard deviation	Red	406.45	40.56	1.58	426.17	0.00
	Green	436.98	40.40	1.59	455.17	0.00
	Blue	525.82	46.68	1.87	543.09	0.00
	NIR	689.64	52.65	4.65	691.51	0.00
Entropy	Red	10.04	21.91	22.06	22.22	22.12
	Green	10.21	22.89	21.93	22.18	22.51
	Blue	10.55	21.97	21.96	22.24	22.53
	NIR	11.04	21.54	22.09	22.20	22.54
Mutual Information	Red		6.48	6.57	6.69	6.62
	Green		6.58	6.60	6.78	7.00
	Blue		6.87	6.86	7.05	7.25
	NIR		6.91	7.29	7.23	7.60

Table 5.1: Comparison of the quality metrics between the multi-spectral and fused bands.

At this stage, having finished with the fusion results, the classification assessment can begin. First, the multi-spectral predicted layer is shown in figure 5.15, while the SAR in figure 5.16. It's visible that the classification result of the multi-spectral bands yields a satisfactory image. In general, it has separated all the five classes but has also made some noticeable mistakes. The first and biggest error is in the north-east part, where there is the sandy beach with its coastal waters. Due to coastal currents, the shallow waters are full of sediments. These sediments have the same spectral signature with most of the urban features, thus the random forest classifier identifying it as an artificial surface. The second error is in the south part of the image, at the river and near the estuary. It has classified the riverbanks as an artificial surface instead of semi-natural grassland or wetland. The reason is the same as the previous one. Finally, while not an error, it's clear that the urban classification, while accurate in space, not so "confident" in the intra-class parts. It's more evident in the north-west part of the river where the main port of Dar es Salaam while being correctly classified as artificial, has many empty spots.

As for the SAR classification, it's unreliable to consider it as a stand-alone image. It has a lot of noise, and the majority of the classes are mixed up together, without a clear pattern.

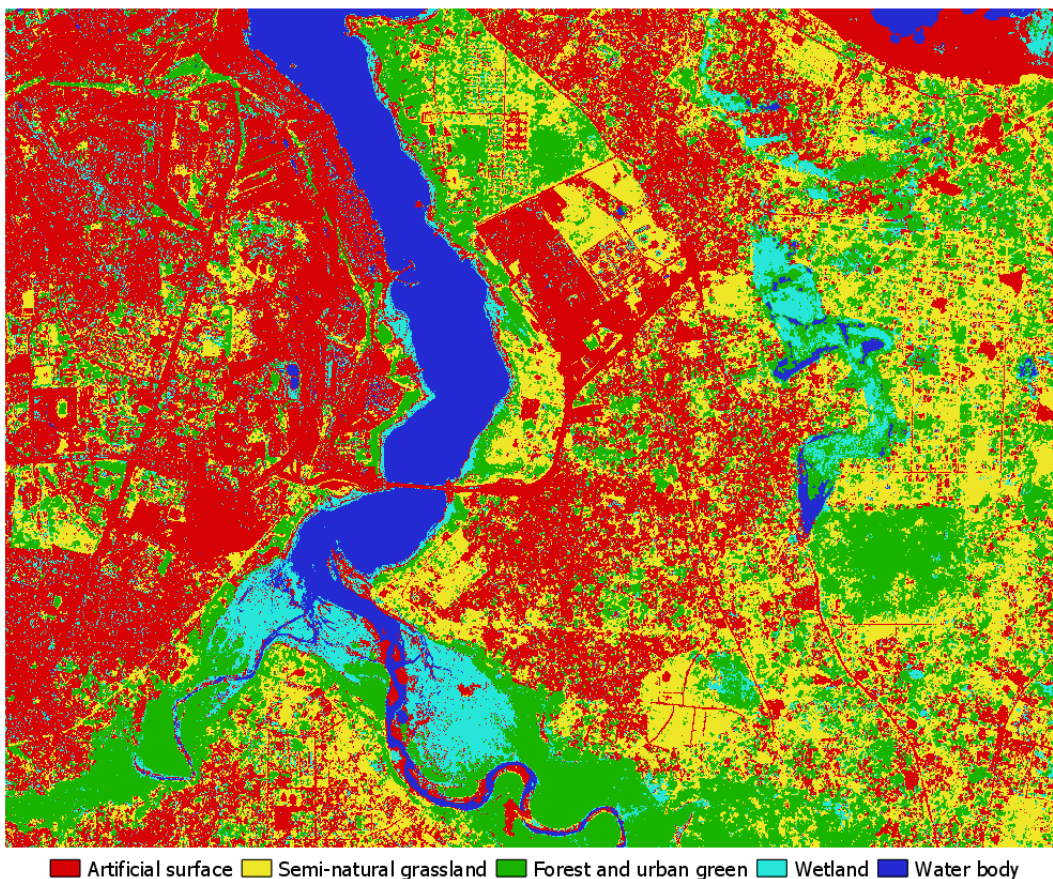


Figure 5.15: Classification result of the multi-spectral bands.

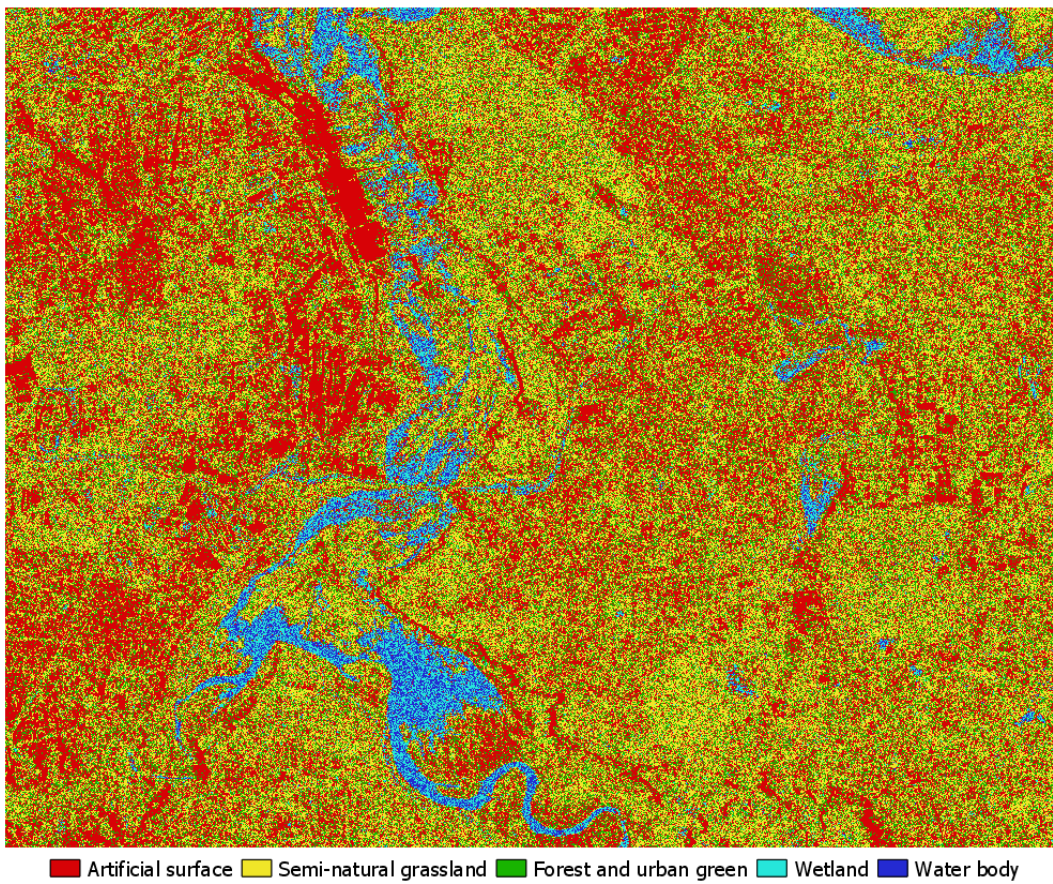


Figure 5.16: Classification result of the SAR band.

The MLT classification result is shown in figure 5.17. It has the same structure as the multi-spectral one, with some minor improvements. The misclassification problem with the riverbanks seems fixed to an extent. There is also a better depiction of the water body in the lake in the east part of the region. It has also filled some of the urban holes in the whole region, but it isn't able to transfer the correct amount from the SAR imagery to fix the coastal problem on the north.

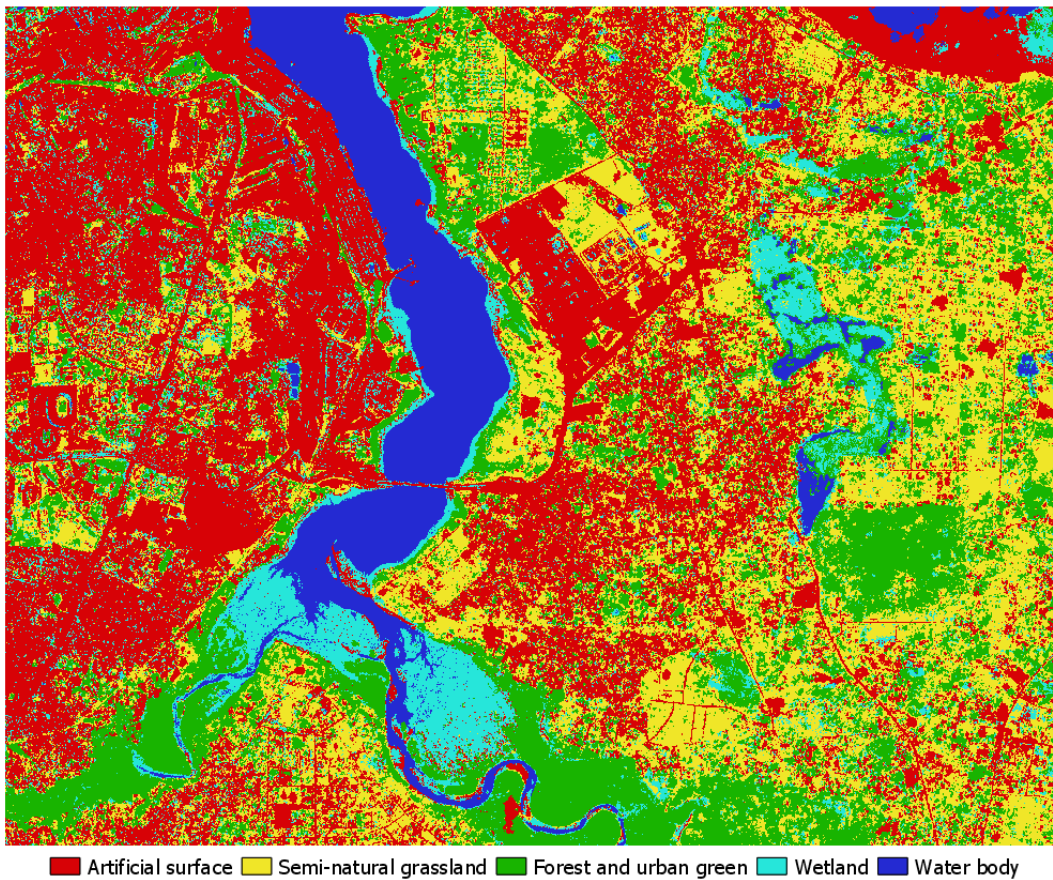


Figure 5.17: Classification result of the MLT bands.

Figure 5.18 shows the result from the BT classification. While this is the best predicted artificial surface class, it has some major errors in the rest of the classes. First, it has worsened the urban misclassification in the south by fixing the riverbanks but distorting the nearby wetland. It has distorted the entire lake area of the east and didn't improve on the coastal misclassification issue. In general, it seems that this type of fusion had the majority of the errors in the wetland class.

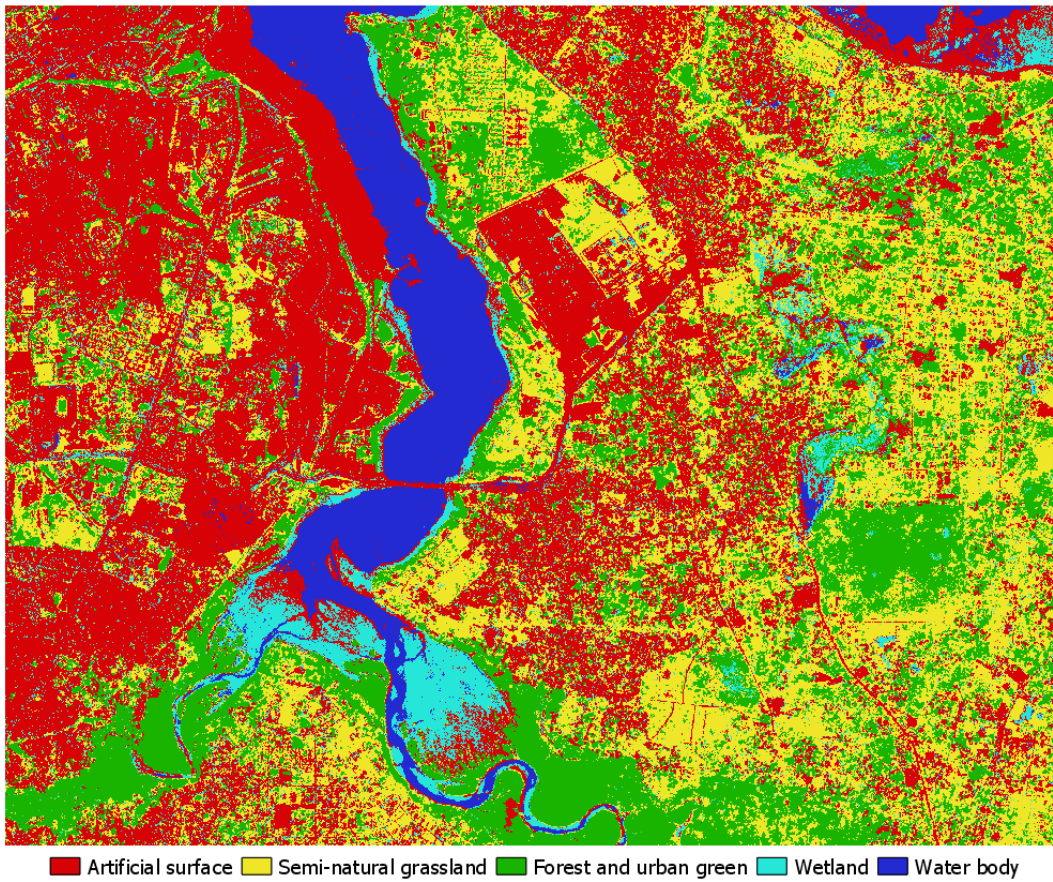


Figure 5.18: Classification result of the BT bands.

The HPFA Sobel classification result is shown in figure 5.19. Visually it's pretty similar to the multi-spectral classification and has no major changes in land cover types. The only minor improvements are in the artificial surface class, near the wetlands.

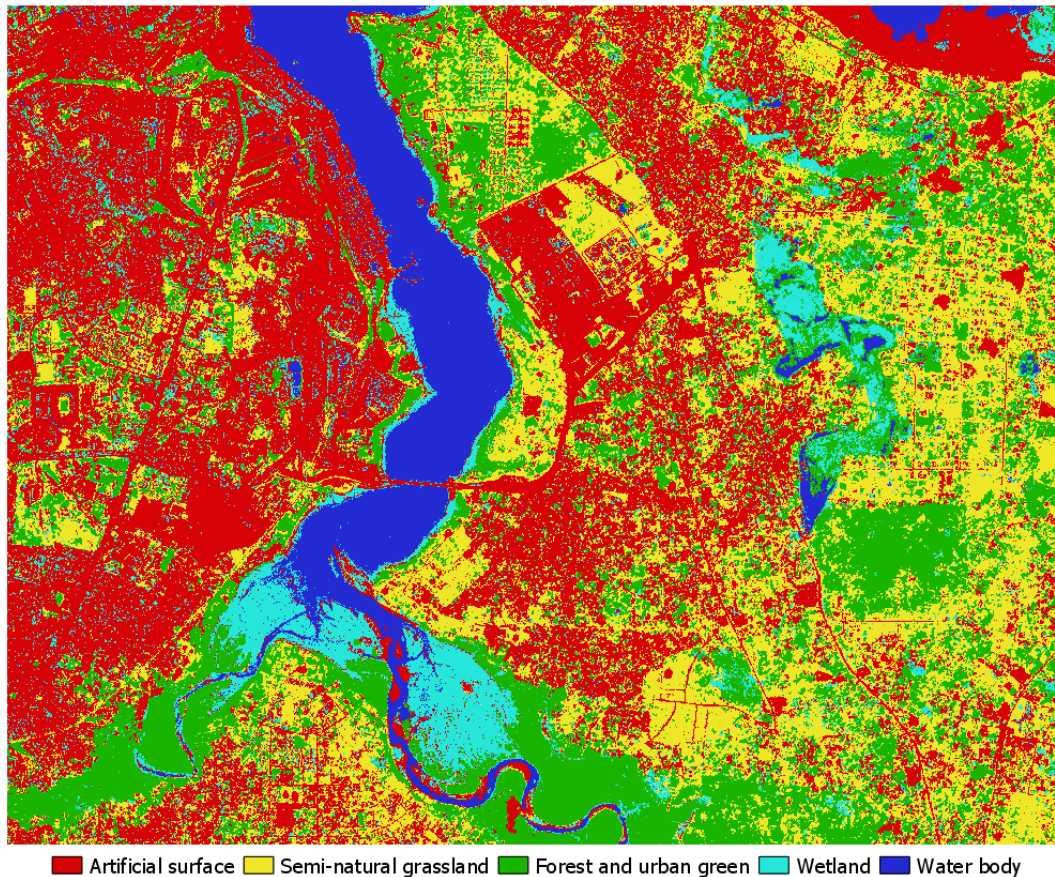


Figure 5.19: Classification result of the HPFA bands.

Lastly, the result from the PCA classification is shown in figure 5.20. Similar to the HPFA, PCA has a comparable pattern with the multi-spectral one. The only difference with the HPFA classifications is that PCA favors a bit more the forest and urban green areas. It doesn't fix the coastal problem and still has the same problem in the south with the wetland misclassification.

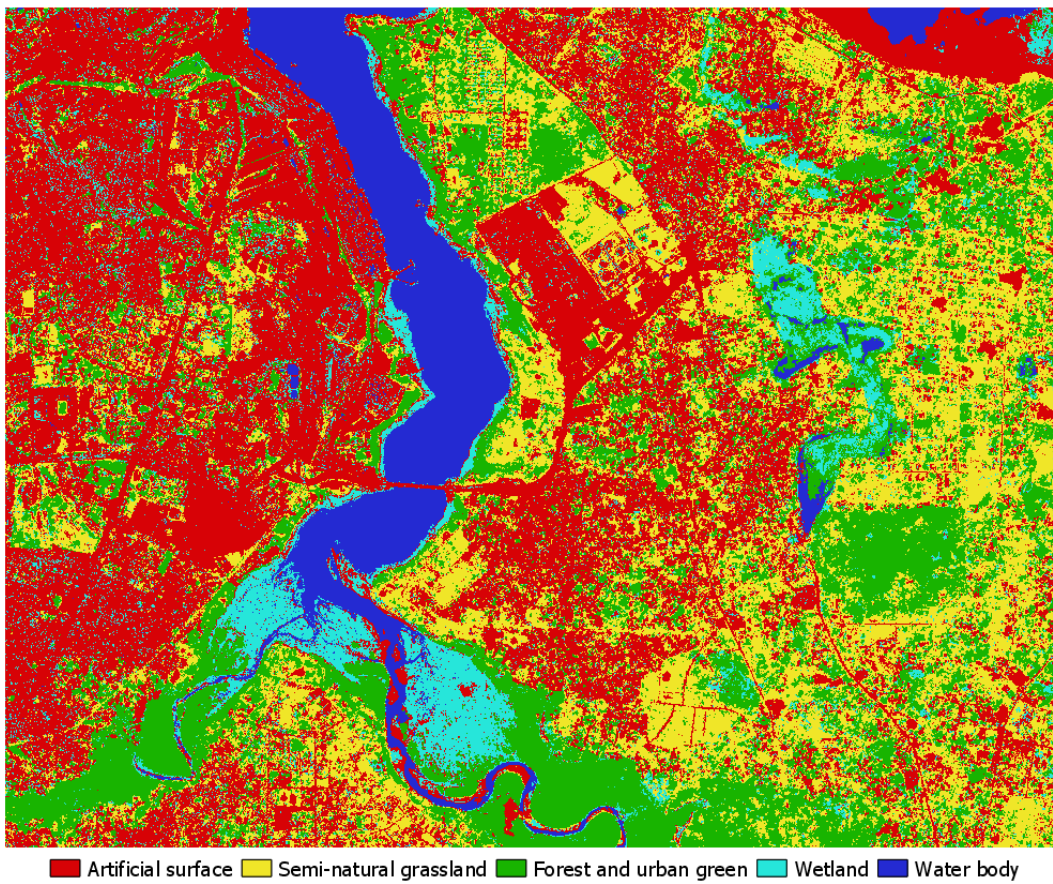


Figure 5.20: Classification result of the PCA bands.

Having seen the classification layers, next up follows the classification accuracies. Table 5.2 shows the OOB, OA, K, and feature importance metrics. The majority of the fusion bands have similar metric values as the multi-spectral, with the biggest OOB score belonging to the PCA classification, while the highest OA and K the MLT classification. Changes in the values on the scale of 0.2% may seem small but are significant enough to fix the estuary/wetland errors and to fill some urban spots, the multi-spectral wasn't able to fill.

Table 5.2 highlights each fused bands performance compared to the multi-spectral. It seems the red band is mostly used in the multi-spectral classification, the green band in the PCA classification with a percentage of 47% (almost half the total importance), the blue band in the MLT classification, and the NIR band being equally used in the multi-spectral and BT classification. Every fused product uses a different combination of the available bands to classify the AOI.

Metrics	Multi-spectral	SAR	MLT	BT	HPFA (Sobel)	PCA
OBB	92.62%	42.27%	92.26%	90.01%	92.30%	92.98%
OA	83.81%	35.20%	85.06%	81.47%	84.34%	84.15%
k	0.79	0.15	0.80	0.75	0.79	0.79
Band importance						
Band 1	0.29	1.00	0.17	0.22	0.25	0.26
Band 2	0.18	0.00	0.19	0.24	0.18	0.47
Band 3	0.26	0.00	0.30	0.18	0.22	0.20
Band 4	0.36	0.00	0.34	0.36	0.35	0.07

Table 5.2: The OOB, OA, K, and feature importance metrics for the classification results.

Besides the metrics, it's important to also consider the confusion matrixes. Figure 5.21 shows the confusion matrix of each classified layer. On the x-axis it has the predicted number of pixels for each class, and on the y-axis has the number of ground truth pixels for each class. Ideally, the matrix has all the pixel values in the upper-left diagonal line, but in practice, no classifier can correctly categorize each pixel to it, accurate class. Thus making the confusion matrix shows the user which classes confused the random forest the most. Because of the NIR band, all the classifiers have successfully classified most of the forest and urban green areas. Also, the water body and artificial surface classes are pretty similar for all of them.

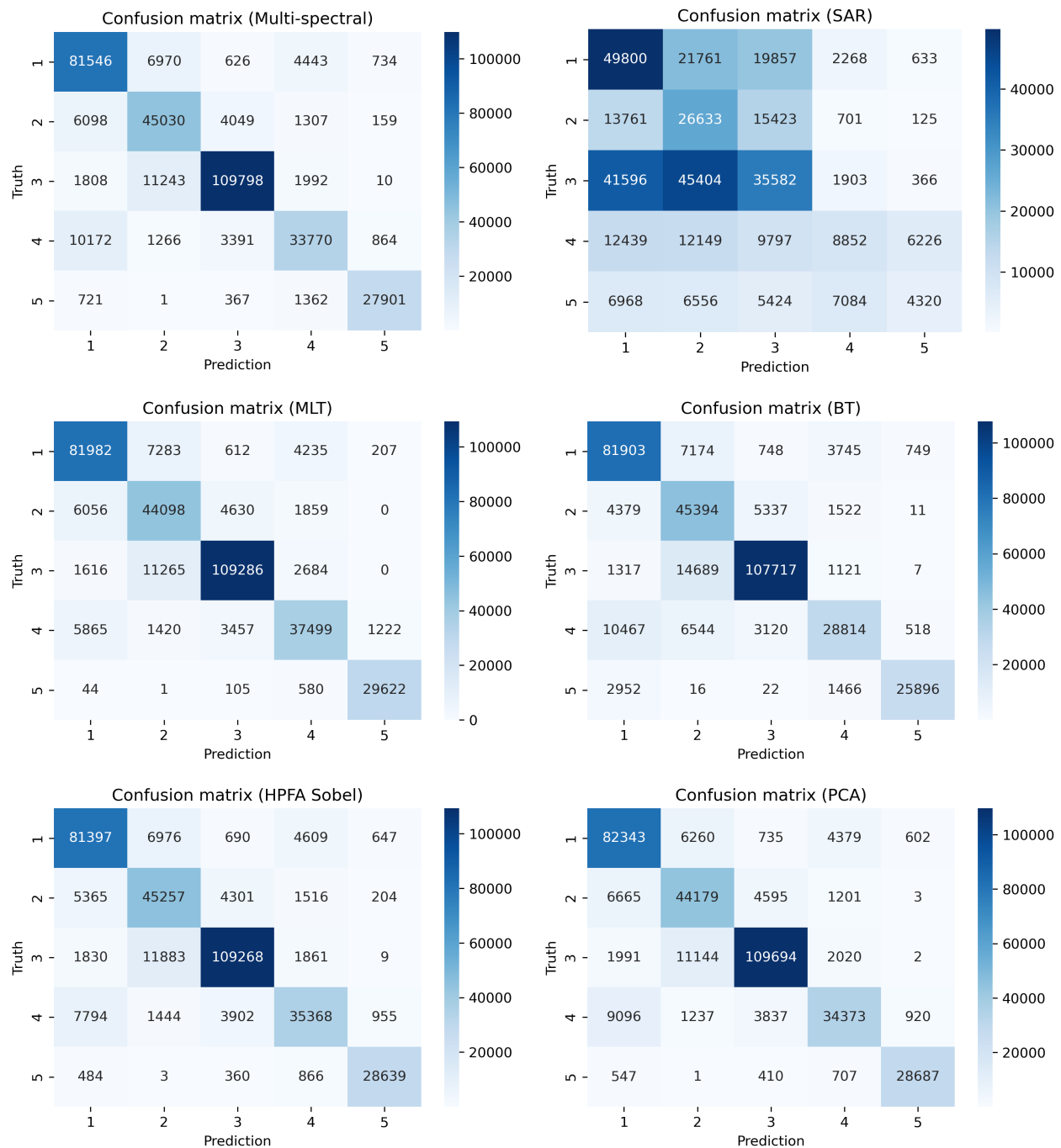


Figure 5.21: Confusion matrix of each classified image.

In terms of total classified area, figure 5.22 shows the participation in the percentage of each class in the entire AOI.

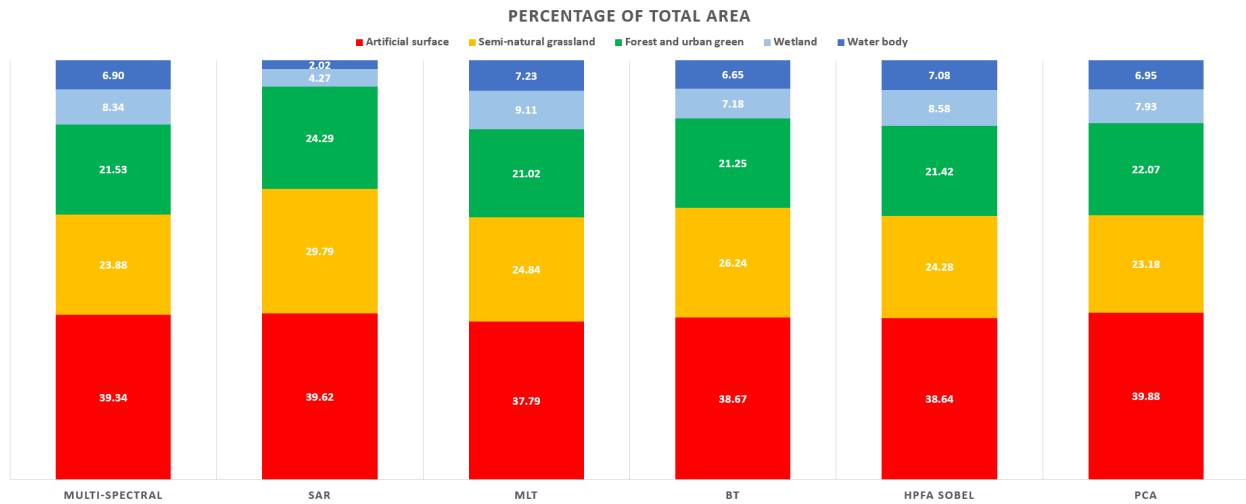


Figure 5.22: Corresponding figure to the table 5.3.

Except for the SAR result, all the rest predicted layers (including the multi-spectral) have similar percentages with each other. Table 5.3 shows the highest standard deviation of all the classes belongs to the semi-natural, wetland, and artificial surface. As mentioned, the forest and water body classes, due to the NIR contribution, are the ones with the lowest standard deviation. The semi-natural grassland class is most confused with the forest and urban green while the artificial surface class with the wetland, just what the visual inspection and confusion matrixes showed.

Class	Multi-spectral	SAR	MLT	BT	HPFA	PCA	SD	SD*
Artificial surface	39.34	39.62	37.79	38.67	38.64	39.88	0.70	0.55
Semi-natural grassland	23.88	29.79	24.84	26.24	24.28	23.18	2.19	0.89
Forest and urban green	21.53	24.29	21.02	21.25	21.42	22.07	1.10	0.19
Wetland	8.34	4.27	9.11	7.18	8.58	7.93	1.59	0.71
Water body	6.90	2.02	7.23	6.65	7.08	6.95	1.85	0.22

Table 5.3: Corresponding table to the figure 5.22, with the addition of SD comparison and the SD* which removes the SAR data-set to compare only with the multi-spectral.

Chapter 6

Discussion

From the results, it's reasonable to derive a definitive answer to the stated hypothesis. The fusion products PlanetScope and ICEYE-X2, under specific pre-processing, fusion, and classification methods, yield higher classification accuracies than the multi-spectral imagery alone. These findings are in line with other literature conclusions that suggest fusion methods deliver superior outcomes in classification applications. In this case, the fusion product has a small margin in the OA and K indicators, which translates into improvements in classifications errors. The constellations of PlanetScope and ICEYE are still small in number, but in time, when they will expand, their fusion capability will provide beneficiary information regarding land cover mapping. The experimental workflow shown in this study can, in theory, be improved and scaled to provide appropriate added-value products.

This work had also major obstacles in the making. For start, the literature of image fusion is limited and difficult to find. Most of the relevant papers assume knowledge of high-level mathematics, image processing techniques, and machine learning workflows. These preconditions make for a tough start for a remote sensing researcher. Besides, these fusion and classification methods are not available in the common GIS softwares. Thus, creating the need to develop algorithms and workflows from the ground up. The programming skills and experience these algorithms require is an addition to the preliminaries needed. There were also two technical difficulties, the image co-registration process, and the fusion evaluation metrics. The correct use of pixel-level fusion requires a strict co-registration process that results in subpixel accuracy. The orthorectification of both products, geometric corrections for SAR, and image-to-image registration with GCPs fixed the problem to an extent, but the inherent geometric properties of SAR sensors are difficult to eliminate. Hence, with full awareness of the three pixels deviation, this study tested the robust nature of fusion algorithms to self-correct in imperfect conditions. As for the evaluation problems, it's already stated that most of these indicators are suited for pan-sharpening techniques. The choice of standard deviation, entropy, and mutual information was based on their ability to evaluate the collective result and not per pixel deviation.

Despite these difficulties, this work delivers a general workflow for land cover map production. However, some limitations stem from the pixel-level fusion methods and the classifier itself. I present some suggestions for future improvements to this workflow. The pixel-level fusion considers the most amount of information, hence operating at a very low complexity level. That requires more computational resources and also mathematical expressions that take place only between pixels. Replacing these methods with decision-level ones gives the ability to apply more complex models with fewer restrictions in the co-registration process. Methods that rely on consensus and Dempster-Shafer theories have proven their usefulness in fusing multi-spectral and SAR data. Another improvement can be achieved in the classification department. Instead of implementing the random forest classifier, a neural network, or even better a convolutional neural network (CNN), would vastly improve the classification accuracy. With enough training data, the CNN would also improve the generalization of the model.

Chapter 7

Conclusions

The main objective of this work is to effectively fuse and classify at the pixel-level the optical PlaneScope and SAR ICEYE-X2 data-sets over the Dar es Salaam. The MLT, BT, HPFA, and PCA fusion methods are compared with each other. For the HPFA, the Sobel operator with $\gamma = 10$ is chosen based on the combination of visual inspection, quality metrics, and classification accuracy. The standard deviation doesn't seem a reliable indicator due to the difference in values scale, however, the entropy and mutual information show prominent results. All the produced layers have double the information (also uncertainty) and their distributions have shifted, indicating different values to the original image. While they offer useful insights into the fused products, these metrics can not reliably determine the optimal fusion method. The appropriate way to assess the quality of the fusion is to consider the classification results.

For the classification, the random forest classifier is used. The best parameters for this application are the number of trees set to 300, entropy as the split criterion, and the minimum sample of leaves set to 16. These parameters produce a forest which is robust to imperfect input data and generally produce homogenous areas. No normalization was attempted because the random forest is invariant to monotonic transformations of individual features, thus feature scaling would not affect the results. Visual inspections in combination with metrics show the MLT fusion as the most accurate depicted result, with $OA = 85.06\%$ and $K = 0.80$. With most of the information taken from the blue and NIR bands, the MLT layer is the most consistent, considering the confusion matrix. Still, neither MLT nor the other fusion results could correctly classify the coastal region. The class area with the highest standard deviation between all classified layers, except SAR, is the semi-natural grassland. The classifier can identify forest, urban green, and waterbody pixels easier rather than artificial surface, wetland, and semi-natural grassland pixels.

This paper attempts to balance the theory of pixel-level fusion, classification, and evaluation, while also including experimental results from the area of Dar es Salaam. The amount of different results this study produces makes their interpretation confusing at times, however, the main findings can be described in four bullet points

- The quality metrics of pixel-level fusion, while insightful, are not able to produce reliable results for the fusion process.
- The random forest classifier, with the right parameters and input bands, gives accurately depicted images with high degrees of OOB, OA, and K.
- Fusion of optical and SAR images gives better classification results than standalone optical or SAR classification.
- PlanetScope and ICEYE-X2 data-sets are compatible for fusion, and in some cases, complement each other's features.

From the beginning, the hypothesis states that a fusion product can outperform in classification accuracy the multi-spectral image. With PlanetScope and ICEYE-X2 as the data, MLT, HPFA, and PCA as the fusion algorithms, and the random forest as the classifier, this hypothesis is valid.

Acknowledgements

I would like to express my sincere gratitude to my supervisor Dr. Issaak Parcharidis for providing guidance and feedback throughout this project. He consistently allowed this paper to be my own work but steered me in the right direction whenever he thought I needed it.

Special thanks to Professor Dr. Efthymios Lekkas for his contribution to PlanetScope data. Without this data-set, this dissertation would not have been possible. The images are obtained with the support of the NSRF research project "Laertes".

Finally, I must express my deep gratitude to my family, colleagues, and friends for providing me with unfailing support and continuous encouragement throughout my postgraduate program and through the process of researching and writing this thesis. Thank you.

Bibliography

- Al-Wassai, F. A., Kalyankar, N. V., & Al-Zuky, A. A. (2011). Arithmetic and frequency filtering methods of pixel-based image fusion techniques. *arXiv preprint arXiv:1107.3348*.
- Abdikan, S., Bilgin, G., Sanli, F. B., Uslu, E., & Ustuner, M. (2015). Enhancing land use classification with fusing dual-polarized TerraSAR-X and multispectral RapidEye data. *Journal of Applied Remote Sensing*, 9(1), 096054.
- Amarsaikhan, D., Blotevogel, H. H., Van Genderen, J. L., Ganzorig, M., Gantuya, R., & Nergui, B. (2010). Fusing high-resolution SAR and optical imagery for improved urban land cover study and classification. *International Journal of Image and Data Fusion*, 1(1), 83-97.
- Benediktsson, J. A., & Swain, P. H. (1992). Consensus theoretic classification methods. *IEEE transactions on Systems, Man, and Cybernetics*, 22(4), 688-704.
- Benediktsson, J. A., & Kanellopoulos, I. (1999). Classification of multisource and hyperspectral data based on decision fusion. *IEEE Transactions on Geoscience and Remote Sensing*, 37(3), 1367-1377.
- Canty, M. J. (2014). *Image analysis, classification and change detection in remote sensing: with algorithms for ENVI/IDL and Python*. Crc Press.
- Carletta, J. (1996). Assessing agreement on classification tasks: the kappa statistic. *arXiv preprint cmp-lg/9602004*.
- Cliche, F. (1985). Integration of the SPOT panchromatic channel into its multispectral mode for image sharpness enhancement. *PE & RS*, 51(3), 311-316.
- Collison, A., & Wilson, N. (2017). Planet Surface Reflectance Product. *Technical White Paper: Version, 1*.

- Dahiya, S., Garg, P. K., & Jat, M. K. (2013). A comparative study of various pixel-based image fusion techniques as applied to an urban environment. *International Journal of Image and Data Fusion*, 4(3), 197-213.
- Filipponi, F. (2019). Sentinel-1 GRD Preprocessing Workflow. In *Multidisciplinary Digital Publishing Institute Proceedings* (Vol. 18, No. 1, p. 11).
- Gangkofner, U. G., Pradhan, P. S., & Holcomb, D. W. (2007). Optimizing the high-pass filter addition technique for image fusion. *Photogrammetric Engineering & Remote Sensing*, 73(9), 1107-1118.
- Gislason, P. O., Benediktsson, J. A., & Sveinsson, J. R. (2006). Random forests for land cover classification. *Pattern Recognition Letters*, 27(4), 294-300.
- Ghose, M. K., Pradhan, R., & Ghose, S. S. (2010). Decision tree classification of remotely sensed satellite data using spectral separability matrix. *International Journal of Advanced Computer Science and Applications*, 1(5).
- Gibril, M. B. A., Bakar, S. A., Yao, K., Idrees, M. O., & Pradhan, B. (2017). Fusion of RADARSAT-2 and multispectral optical remote sensing data for LULC extraction in a tropical agricultural area. *Geocarto international*, 32(7), 735-748.
- ICEYE. (2020). SAR product guide v.3.1. <https://www.iceye.com/downloads/product-documents>
- Jagalingam, P., & Hegde, A. V. (2015). A review of quality metrics for fused image. *Aquatic Procedia*, (4), 133-142.
- James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112, p. 18). New York: springer.
- Jin-Yu, Z., Yan, C., & Xian-Xiang, H. (2009). Edge detection of images based on improved Sobel operator and genetic algorithms. In *2009 International Conference on Image Analysis and Signal Processing* (pp. 31-35). IEEE.
- Kulkarni, S. C., & Rege, P. P. (2020). Pixel Level Fusion Techniques for SAR and Optical Images: A Review. *Information Fusion*.
- Kosztra, B., Büttner, G., Hazeu, G., & Arnold, S. (2017). Updated CLC illustrated nomenclature guidelines. *European Environment Agency: Wien, Austria*, 1-124.
- Kombe, W. J. (2005). Land use dynamics in peri-urban areas and their implications on the urban

- growth and form: the case of Dar es Salaam, Tanzania. *Habitat International*, 29(1), 113-135.
- Kombe, W. J., & Kreibich, V. (2000). *Informal land management in Tanzania*. SPRING Centre, Faculty of Spatial Planning, University of Dortmund, Germany.
- Le Hégat-Masclé, S., Bloch, I., & Vidal-Madjar, D. (1997). Application of Dempster-Shafer evidence theory to unsupervised classification in multisource remote sensing. *IEEE transactions on geoscience and remote sensing*, 35(4), 1018-1031.
- Li, S., Kang, X., Fang, L., Hu, J., & Yin, H. (2017). Pixel-level image fusion: A survey of the state of the art. *Information Fusion*, 33, 100-112.
- Liaw, A., & Wiener, M. (2002). Classification and regression by randomForest. *R news*, 2(3), 18-22.
- Lowe, B., & Kulkarni, A. (2015). Multispectral image analysis using random forest.
- Lu, D., Batistella, M., & Moran, E. (2007). Land-cover classification in the Brazilian Amazon with the integration of Landsat ETM+ and Radarsat data. *International Journal of Remote Sensing*, 28(24), 5447-5459.
- Marcello, J., Medina, A., & Eugenio, F. (2012). Evaluation of spatial and spectral effectiveness of pixel-level fusion techniques. *IEEE Geoscience and remote sensing letters*, 10(3), 432-436.
- Meng, X., Shen, H., Li, H., Zhang, L., & Fu, R. (2019). Review of the pansharpening methods for remote sensing images based on the idea of meta-analysis: Practical discussion and challenges. *Information Fusion*, 46, 102-113.
- Miranda, N., & Meadows, P. J. (2015). Radiometric Calibration of S-1 Level-1 Products Generated by the S-1 IPF. *Technical note*.
- Mudrova, M., & Procházka, A. (2005). Principal component analysis in image processing. In *Proceedings of the MATLAB Technical Computing Conference*, Prague.
- Planet. (2020). Our Constellation. <https://storage.googleapis.com/planet-ditl/day-in-the-life/index.html>
- Pohl, C., & Van Genderen, J. L. (1998). Review article multisensor image fusion in remote sensing: concepts, methods and applications. *International journal of remote sensing*, 19(5), 823-854.
- Pohl, C., & Van Genderen, J. (2016). *Remote sensing image fusion: A practical guide*. Crc Press.

- Richards, J. A., & Jia, X. (1999). *Remote sensing digital image analysis* (Vol. 3, pp. 10-38). Berlin: Springer.
- Rosen W. Jonathan. (2019). This Tanzanian city may soon be one of the world's most populous. Is it ready? *National Geographic: Environment* <https://www.nationalgeographic.com/environment/2019/04/tanzanian-city-may-soon-be-one-of-the-worlds-most-populous/>
- Schowengerdt, R. A. (1980). Reconstruction of multispatial, multispectral image data using spatial frequency content. *Photogrammetric Engineering and Remote Sensing*, 46(10), 1325-1334.
- Shapiro, L. G., & Stockman, G. C. (2000). *Computer Vision*, March 2000, pp. 168.
- Solberg, A. H. (2006). Data fusion for remote sensing applications. *Signal and image processing for remote sensing*, 249-271.
- Small, D., & Schubert, A. (2008). Guide to ASAR geocoding. *ESA-ESRIN Technical Note RSL-ASAR-GC-AD*, 1, 36.
- Tu, T. M., Su, S. C., Shyu, H. C., & Huang, P. S. (2001). A new look at IHS-like image fusion methods. *Information fusion*, 2(3), 177-186.
- Ehlers, M., Klonus, S., Johan Åstrand, P., & Rosso, P. (2010). Multi-sensor image fusion for pansharpening in remote sensing. *International Journal of Image and Data Fusion*, 1(1), 25-45.
- Sajjad, H., & Kumar, P. (2019). Future Challenges and Perspective of Remote Sensing Technology. In *Applications and Challenges of Geospatial Technology* (pp. 275-277). Springer, Cham.
- Xu, M., Watanachaturaporn, P., Varshney, P. K., & Arora, M. K. (2005). Decision tree regression for soft classification of remote sensing data. *Remote Sensing of Environment*, 97(3), 322-336.
- Yin, N., & Jiang, Q. G. (2013). Feasibility of multispectral and synthetic aperture radar image fusion. In *2013 6th International Congress on Image and Signal Processing (CISP)* (Vol. 2, pp. 835-839). IEEE.
- Zhang, Y. (2008). Methods for image fusion quality assessment-a review, comparison and analysis. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37(part b7), 1101-1109.

Appendix

Rasterization workflow

```
from osgeo import ogr, gdal

gdal.UseExceptions()
gdal.AllRegister()

dataset = ogr.Open("../train_samples.shp")
raster_ds = gdal.Open("../stack_image.tif", gdal.GA_ReadOnly)

layer = dataset.GetLayerByIndex(0)
print("The layer is named: {n}".format(n = layer.GetName()))

rows = raster_ds.RasterYSize
cols = raster_ds.RasterXSize
print("Image size is: {r} rows x {c} columns".format(r = rows, c = cols))

proj = raster_ds.GetProjection()
print("Image projection: {proj}".format(proj = proj))

gt = raster_ds.GetGeoTransform()
print("Image geo-transform: {gt}".format(gt = gt))

driver = gdal.GetDriverByName("GTiff")

out_raster_ds = driver.Create("../train_data.tif", cols, rows, 1, gdal.GDT_Byte)
out_raster_ds.SetProjection(proj)
out_raster_ds.SetGeoTransform(gt)

b = out_raster_ds.GetRasterBand(1)
b.Fill(0)

status = gdal.RasterizeLayer(out_raster_ds, [1], layer, None, None, [0], ["ALL_TOUCHED=TRUE", "ATTRIBUTE=id"])
```

```
out_raster_ds, raster_ds = None
```

Basic procedure

```
from osgeo import ogr, gdal, gdal_array
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy import ndimage
from sklearn.decomposition import PCA
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, cohen_kappa_score, confusion_matrix,
    mutual_info_score
from skimage.measure import shannon_entropy
import seaborn as sns
import time

gdal.UseExceptions()
gdal.AllRegister()

# read from disk
img_ds = gdal.Open("../stack.tif", gdal.GA_ReadOnly)
samples_ds = gdal.Open("../train_data.tif", gdal.GA_ReadOnly)
test_ds = gdal.Open("../test_data.tif", gdal.GA_ReadOnly)

# write into array
img = np.zeros((img_ds.RasterYSize, img_ds.RasterXSize, img_ds.RasterCount),
    gdal_array.GDALTypeCodeToNumericTypeCode(img_ds.GetRasterBand(1).DataType))

for band in range(img.shape[2]):
    img[:, :, band] = img_ds.GetRasterBand(band + 1).ReadAsArray()

samples = samples_ds.GetRasterBand(1).ReadAsArray().astype(np.uint8)
test = test_ds.GetRasterBand(1).ReadAsArray().astype(np.uint8)

# split to multi_spectral and sar
m = img[:, :, 0:4]
s = img[:, :, 4]
```

Extra steps for PCA

```
pca_dim = (img.shape[0] * img.shape[1], img.shape[2])
m_ = img.reshape(pca_dim)
m_trans = np.ndarray.transpose(m_)
```

MLT function

```
def MLT(multispectral, sar):

    mlt_fusion = np.zeros_like(multispectral)

    for band in range(multispectral.shape[2]):
        mlt_fusion[:, :, band] = multispectral[:, :, band] * sar

    return mlt_fusion
```

BT function

```
def BT(multispectral, sar):

    brovey_fusion = np.zeros_like(multispectral)

    sum_multispectral = multispectral[:, :, 0] + multispectral[:, :, 1] +
    multispectral[:, :, 2] + multispectral[:, :, 3]

    for band in range(multispectral.shape[2]):
        brovey_fusion[:, :, band] = (multispectral[:, :, band] / sum_multispectral)
        * sar

    return brovey_fusion
```

HPFA functions

```
def HPFA_narrow(multispectral, sar):

    conv_3x3 = np.array([[ -1,  -1,  -1],
                        [ -1,   8,  -1],
                        [ -1,  -1,  -1]])

    highpass_3x3 = ndimage.convolve(sar, conv_3x3)
    narrow_fusion = np.zeros_like(multispectral)

    for band in range(multispectral.shape[2]):
        narrow_fusion[:, :, band] = multispectral[:, :, band] + highpass_3x3

    return narrow_fusion

def HPFA_wide(multispectral, sar):

    conv_5x5 = np.array([[ -1,  -1,  -1,  -1,  -1],
                        [ -1,   1,   2,   1,  -1],
                        [ -1,   2,   4,   2,  -1],
                        [ -1,   1,   2,   1,  -1],
```

```

        [-1, -1, -1, -1, -1]))

highpass_5x5 = ndimage.convolve(sar, conv_5x5)
wide_fusion = np.zeros_like(multispectral)

for band in range(multispectral.shape[2]):
    wide_fusion[:, :, band] = multispectral[:, :, band] + highpass_5x5

return wide_fusion

def HPFA_gauss(multispectral, sar):

    lowpass = ndimage.gaussian_filter(sar, sigma = 3)
    gauss_highpass = sar - lowpass

    gauss_fusion = np.zeros_like(multispectral)

    for band in range(multispectral.shape[2]):
        gauss_fusion[:, :, band] = multispectral[:, :, band] + gauss_highpass

    return gauss_fusion

def HPFA_sobel(multispectral, sar):

    Gx = np.array([[1, 0, -1],
                   [2, 0, -2],
                   [1, 0, -1]])
    Gy = np.array([[1, 2, 1],
                   [0, 0, 0],
                   [-1, -2, -1]])

    Gx_filter = ndimage.convolve(sar, Gx)
    Gy_filter = ndimage.convolve(sar, Gy)

    G = np.sqrt(Gx_filter**2 + Gy_filter**2)

    sobel_fusion = np.zeros_like(multispectral)

    for band in range(multispectral.shape[2]):
        sobel_fusion[:, :, band] = multispectral[:, :, band] + 10 * G

    return sobel_fusion

```

PCA function through sklearn

```

pca_fusion = PCA(n_components = 4)
pca_fusion.fit_transform(m_trans)

```


Implementation of the MLT, BT, and HPFA functions

```
# this example uses the BT function
bt_results = BT(m, s)

# assign new variables for each fused band
bt_red = bt_results[:, :, 0]
bt_green = bt_results[:, :, 1]
bt_blue = bt_results[:, :, 2]
bt_nir = bt_results[:, :, 3]

# flatten the multi-spectral and fused results
red = m[:, :, 0].flatten()
green = m[:, :, 1].flatten()
blue = m[:, :, 2].flatten()
nir = m[:, :, 3].flatten()

bt_red_f = bt_red.flatten()
bt_green_f = bt_green.flatten()
bt_blue_f = bt_blue.flatten()
bt_nir_f = bt_nir.flatten()
```

Implementation of the PCA function

```
# assign new variables for the multi-spectral bands and transpose them
red = m_trans.T[:, 0]
green = m_trans.T[:, 1]
blue = m_trans.T[:, 2]
nir = m_trans.T[:, 3]

# assign new variables for the fused ones and transpose them
pca_red = pca_fusion.components_.T[:, 0]
pca_green = pca_fusion.components_.T[:, 1]
pca_blue = pca_fusion.components_.T[:, 2]
pca_nir = pca_fusion.components_.T[:, 3]

# reshape the pca results
comp = np.ndarray.transpose(pca_fusion.components_)
new_dim = (img.shape[0], img.shape[1], img.shape[2] - 1)
comp_ = comp.reshape(new_dim)
```

Quality metrics

```
# calculate entropy
ent_red = shannon_entropy(red)
ent_green = shannon_entropy(green)
ent_blue = shannon_entropy(blue)
ent_nir = shannon_entropy(nir)
```

```

ent_bt_red = shannon_entropy(bt_red_f)
ent_bt_green = shannon_entropy(bt_green_f)
ent_bt_blue = shannon_entropy(bt_blue_f)
ent_bt_nir = shannon_entropy(bt_nir_f)

# calculate mi
mi_red = mutual_info_score(red, bt_red_f)
mi_green = mutual_info_score(green, bt_green_f)
mi_blue = mutual_info_score(blue, bt_blue_f)
mi_nir = mutual_info_score(nir, bt_nir_f)

# calculate sd
std_red = np.std(red)
std_green = np.std(green)
std_blue = np.std(blue)
std_nir = np.std(nir)

std_bt_red = np.std(bt_red_f)
std_bt_green = np.std(bt_green_f)
std_bt_blue = np.std(bt_blue_f)
std_bt_nir = np.std(bt_nir_f)

```

Visualization process

```

# normalize to range [0, 1] the original and fused bands
minmax_m = np.zeros((m.shape[0] * m.shape[1], m.shape[2]))
minmax_bt = np.zeros((bt_results.shape[0] * bt_results.shape[1], bt_results.shape
[2]))

for band in range(m.shape[2]):
    f = m[:, :, band].flatten()
    ff = bt_results[:, :, band].flatten()
    scaler_f = (f - f.min()) / (f.max() - f.min())
    scaler_ff = (ff - ff.min()) / (ff.max() - ff.min())
    minmax_m[:, band] = scaler_f
    minmax_bt[:, band] = scaler_ff

m_dim = (m.shape[0], m.shape[1], m.shape[2])

m_ = minmax_m.reshape(m_dim)
bt_ = minmax_bt.reshape(m_dim)

# choose which three bands to show (e.g., (4-3-2))
index = np.array([3, 2, 1])
colors = m_[:, :, index].astype(np.float64)
colors_f = bt_[:, :, index].astype(np.float64)

```

```

# apply min and max value for the visualization
max_val = 0.4
min_val = m_.min()

max_val_ = 0.7
min_val_ = bt_.min()

# create a mask of values
colors[colors[:, :, :] > max_val] = max_val
colors[colors[:, :, :] < min_val] = min_val

colors_f[colors_f[:, :, :] > max_val_] = max_val_
colors_f[colors_f[:, :, :] < min_val_] = min_val_

# apply the mask
for b in range(colors.shape[2]):
    colors[:, :, b] = colors[:, :, b] * 1 / (max_val - min_val)
    colors_f[:, :, b] = colors_f[:, :, b] * 1 / (max_val_ - min_val_)

```

Pair-plots

```

# plot the pairplots
col_ms = ["Red", "Green", "Blue", "NIR", "VV"]
df = pd.DataFrame(m_trans.T, columns = col_ms)
df_sample = df.sample(1000)

sns.pairplot(df_sample, diag_kind = "kde", plot_kws = {"s": 4, "linewidth": 0.2})
plt.savefig("../pairplot_band.png", dpi = 300, bbox_inches = "tight")

col_pc = ["PC1", "PC2", "PC3", "PC4"]
df_pc = pd.DataFrame(pca_fusion.components_.T, columns = col_pc)
df_pc_sample = df_pc.sample(1000)

sns.pairplot(df_pc_sample, diag_kind = "kde", plot_kws = {"s": 4, "linewidth":
    0.2})
plt.savefig("../pairplot_pcs.png", dpi = 300, bbox_inches = "tight")

```

Masking process

```

# mask the sample area
n_samples = (samples > 0).sum()
print('We have {n} samples'.format(n = n_samples))

labels = np.unique(samples[samples > 0])
print('The training data include {n} classes: {classes}'.format(n = labels.size,
    classes = labels))

```

```

x = bt_results[samples > 0, :]
x_ = bt_results[test > 0, :]
y = samples[samples > 0]
y_ = test[test > 0]

print('Our X matrix is sized: {sz}'.format(sz = x.shape))
print('Our y array is sized: {sz}'.format(sz = y.shape))

```

RF function with the metrics

```

# create the and fit the classifier
rf = RandomForestClassifier(n_estimators = 300,
                           criterion = "entropy",
                           min_samples_leaf = 10,
                           oob_score = True,
                           random_state = 9,
                           n_jobs = -1)

rf = rf.fit(x, y)

# calculate classification metrics
acc = accuracy_score(y_, rf.predict(x_))
kappa = cohen_kappa_score(y_, rf.predict(x_))

print("OOB: {}".format(np.round(rf.oob_score_ * 100, 2)))
print("OA: {}".format(np.round(acc * 100, 2)))
print("K: {}".format(np.round(kappa, 3)))

bands = [1, 2, 3, 4]

for b, imp in zip(bands, rf.feature_importances_):
    print("Band {b} importance: {imp}".format(b = b, imp = np.round(imp, 2)))

```

Confusion matrix

```

labels = [1, 2, 3, 4, 5]

cm = confusion_matrix(y_, rf.predict(x_))

sns.heatmap(cm, annot = True, fmt = "g", cmap = "Blues")
plt.xlabel('Prediction')
plt.ylabel('Truth')
plt.xticks([0.5, 1.5, 2.5, 3.5, 4.5], labels)
plt.yticks([0.5, 1.5, 2.5, 3.5, 4.5], labels)
plt.title('Confusion matrix (BT)')

plt.savefig("../cm_bt.png", dpi = 300, bbox_inches = "tight")

```

Final layer prediction

```
# reshape, predict, and reshape again the results
start = time.time()

new_shape = (bt_results.shape[0] * bt_results.shape[1], bt_results.shape[2])

bt_as_array = bt_results.reshape(new_shape)
print('Reshaped from {o} to {n}'.format(o = bt_results.shape,
                                         n = bt_as_array.shape))

class_prediction = rf.predict(bt_as_array)
class_prediction = class_prediction.reshape(bt_results[:, :, 0].shape)

class_proba = rf.predict_proba(bt_as_array)
class_proba = class_proba.reshape(img.shape[0], img.shape[1], img.shape[2])

end = time.time()
print("Total runtime: {runtime} min".format(runtime = (end - start) / 60))
```

Area calculation

```
per = np.unique(class_prediction)
k = []

for c in per:
    k.append((class_prediction == c).sum())
    print("Class {} contains {} pixels".format(c, (class_prediction == c).sum()))

for i in range(5):
    print(np.round((k[i] / class_prediction.size) * 100, 2))
```

Save the results

```
# save the predicted layer
[cols, rows] = class_prediction.shape

driver = gdal.GetDriverByName("GTiff")
outDataRaster = driver.Create("../bt_result.tif", rows, cols, 1, gdal.GDT_Byte)
outDataRaster.SetGeoTransform(img_ds.GetGeoTransform())
outDataRaster.SetProjection(img_ds.GetProjection())
outDataRaster.GetRasterBand(1).WriteArray(class_prediction)
outDataRaster.FlushCache()

del outDataRaster, driver, cols, rows
```