

DETECTION OF AGRICULTURAL FIELD BOUNDARIES FROM SENTINEL-2 IMAGES USING FULLY CONVOLUTIONAL NETWORKS AND SUPER RESOLUTION MAPPING

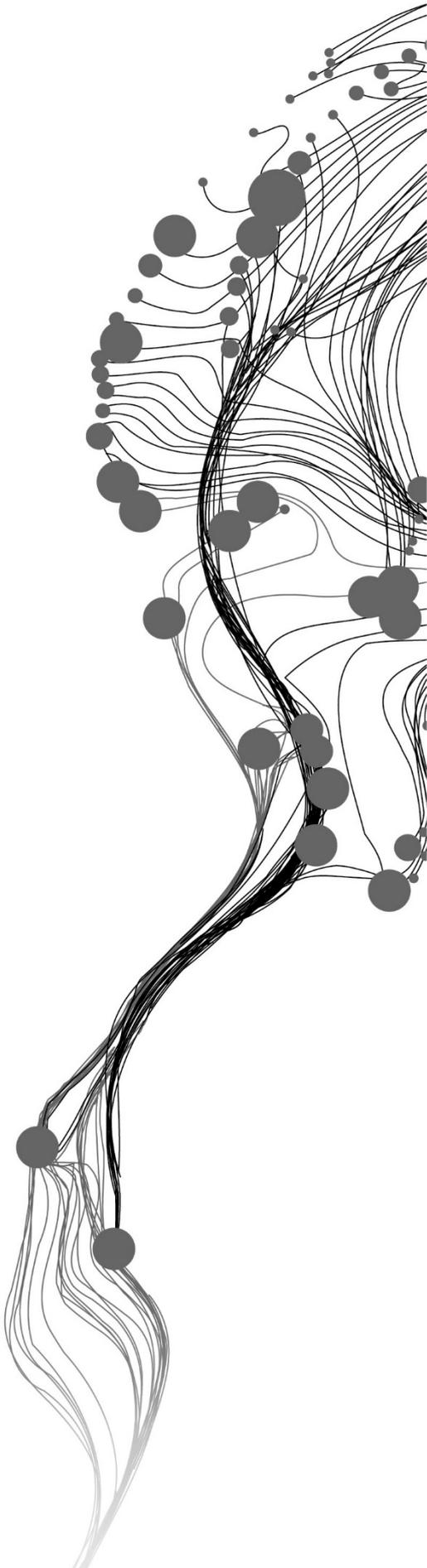
KHAIRIYA MUDRIK MASOUD

February, 2019

SUPERVISORS:

Dr C. Persello

Dr. V.A. Tolpekin



DETECTION OF AGRICULTURAL FIELD BOUNDARIES FROM SENTINEL-2 IMAGES USING FULLY CONVOLUTIONAL NETWORKS AND SUPER RESOLUTION MAPPING

KHAIRIYA MUDRIK MASOUD

February, 2019

Thesis submitted to the Faculty of Geo-Information Science and Earth Observation of the University of Twente in partial fulfilment of the requirements for the degree of Master of Science in Geo-information Science and Earth Observation.

Specialization: Geoinformatics

SUPERVISORS:

Dr. C. Persello

Dr. V.A. Tolpekin

THESIS ASSESSMENT BOARD:

Prof. Dr. ir. A. Stein (Chair)

Dr. M.N. Koeva (External Examiner, University of Twente, ITC-PGM)

DISCLAIMER

This document describes work undertaken as part of a programme of study at the Faculty of Geo-Information Science and Earth Observation of the University of Twente. All views and opinions expressed therein remain the sole responsibility of the author, and do not necessarily represent those of the Faculty.

ABSTRACT

Boundaries of agricultural fields are very important features necessary for defining the agricultural units from which aggregate statistics and other information about agriculture and its associated biophysical characteristics. In this study, we investigated the detection of agricultural fields boundaries (AFB) from open and free available data of Sentinel-2 satellite image of Flevoland province located in the Netherlands using a fully convolutional network (FCN). We investigated the FCN for boundaries detection by developing Deep_FCNet that accepted an input of 8-band combination of the Sentinel-2 at 10 m resolution. The bands are multiresolution, and they were fused outside the network using a bilinear interpolation technique. We also investigated the feasibility of FCN to fuse these multiresolution bands of Sentinel-2 and detect the boundaries within the network by developing an MS-FuseNet. We then, compared these two methods, both produced similar F-Score accuracy, however there is a slight decreased of F-Score for the AFB using MS-FuseNet. Besides, Deep_FCNet outperformed compared to the FCN-DK5 by small increment of F-Score for the AFB. Therefore, Deep_FCNet can be beneficial in detecting boundaries at 10 m resolution from Sentinel-2 image.

Additionally, we developed super resolution mapping technique using FCN to enhance the spatial resolution of the output AFB from 10 m to 5 m. Finally, we designed SRM+LRM-Net that improved the AFB maps by refining the AFB labels from SRM-Net. The results of SRM+LRM-Net are quite similar compared to the results of Deep_FCNet from RapidEye image at 5 m resolution. Based on the results for the AFB from SRM+LRM-Net and considering the complexity of the problem also our accuracy assessment is tolerating any small shift, we concluded that detection of agricultural field boundaries from the Sentinel-2 image using fully convolutional networks and super resolution mapping can produce boundaries at 5 m resolution with the reasonable results.

ACKNOWLEDGMENTS

Firstly, I would like to thank the Almighty God for keeping me healthy and enabling to accomplish this MSc. Secondly, I would like to thank the Netherlands Fellowship Programme for the scholarship to study in ITC faculty at University of Twente.

My special gratitude goes to my supervisors Dr. C. Persello and Dr. V.A. Tolpekin for their valuable advice, support, and patience during my thesis progress. Also, special thanks go to GFM staffs for their teaching and support. My gratitude also goes to my fellow students especially Tanzanians for their support and care.

Thanks to my employer, the State University of Zanzibar (SUZA), for believing in and granting me a study leave. I would also like to thank my colleague at SUZA, Zakaria Khamis for his inspiration to take this MSc.

My deepest gratitude goes to Theresa Burke from student affairs office and Zaid Abubakar for their supporting and helping to complete my study.

Finally, I would like to express my sincere gratitude to my family especially to my lovely husband Mukhtar, daughter, Afrah and son, Omar together with my siblings and friends in Zanzibar for their love, support, and encouragement. Also, special and deep thanks to my mom, dad, and in laws for their love, prayers and keeping my kids safe during the whole period of my study.

TABLE OF CONTENTS

Abstract	3
Acknowledgments	4
List of figures	7
List of tables	9
1. INTRODUCTION.....	10
1.1. Background.....	10
1.2. Research Identification.....	11
1.2.1. Research Objectives.....	11
1.2.2. Research Questions	12
1.2.3. Research Innovation.....	12
2. LITERATURE REVIEW.....	13
2.1. Deep Learning.....	13
2.2. Boundaries Delineation	14
2.3. Image Fusion.....	14
2.4. Super Resolution Mapping (SRM).....	14
3. METHODS.....	16
3.1. Deep FCN-DKs Networks (Deep_FCNet).....	16
3.2. Fusion Networks for Sentinel-2 Multispectral Bands Within the Network (MS-FuseNet).....	17
3.3. Super Resolution Mapping Network (SRM-Net)	18
3.4. Label Refining Modules Network (SRM+LRM-Net).....	19
3.5. Accuracy Assessment.....	20
4. DATASET AND SOFTWARE.....	21
4.1. Study Area.....	21
4.2. Sentinel-2 Satellite Image	21
4.3. RapidEye Satellite Image.....	21
4.4. Reference Data.....	23
4.5. Datasets Preparation	23
4.6. Software	25
5. EXPERIMENTAL ANALYSIS AND RESULTS	28
5.1. Hyperparameters sensitivity analysis.....	28
5.1.1. Training samples	29
5.1.2. Max-pooling layer analysis.....	30
5.1.3. Bands combination analysis	31
5.1.4. Filter size experimental analysis.....	32
5.1.5. Patch size experimental analysis	33
5.1.6. Summary of hyperparameters sensitivity analysis	34
5.2. Architectures experiments	34
5.2.1. Deep_FCNet architectures experiments.....	34
5.2.2. MS-FuseNet architectures experiments.....	36
5.2.3. SRM-Net architectures experiments.....	37
5.2.4. SRM+LRM-Net Architectures Experiments.....	39
5.3. Training Network Hyperparameters.....	40
5.4. Final Experimental Results	40
5.4.1. Results of AFB at 10 m resolution.....	41
5.4.2. Results of AFB at 5 m resolution.....	43
5.5. Performance Analysis of the methods.....	46

5.5.1. Methods comparison at 10 m resolution.....	46
5.5.2. Methods comparison at 5 m resolution.....	47
5.5.3. Output comparison at 10 m and 5 m resolution.....	48
6. DISCUSSION.....	50
6.1. Applicability of the methods.....	50
6.2. Accuracy assessment strategy.....	51
6.3. Limitation of the methods.....	51
6.4. Final remarks.....	51
7. CONCLUSION AND RECOMMENDATION.....	53
7.1. Conclusion.....	53
7.2. Recommendation.....	54
List of references.....	55
List of appendices.....	58

LIST OF FIGURES

Figure 3.1: General Network Architecture for the Deep_FCN-Net: Input image can be any number of channels (bands) of the same resolution. In this study, we use the Sentinel-2 band combination of the 10 m spatial resolution. The output map has the same spatial resolution as an input image. Yellow lines represent the boundaries.	17
Figure 3.2: General Network Architecture for the MS-FuseNet: In this study, we use two inputs. The inputs are the 4-band combination of 10 m and 4-band combination of 20 m resolution from the same image of Sentinel-2. The output map has a spatial resolution of 10 m.....	18
Figure 3.3: SRM-Net Architecture. The input image is a band combination of the 10 m resolution, and the output map is a 5 m resolution.....	19
Figure 3.4: SRM+LRM-Net Architecture.....	19
Figure 4.1: The study area Flevoland, The Netherlands.	21
Figure 4.2: The spatial resolution of Sentinel-2 with their dependent visible, near infrared (VNIR), and short width infrared (SWIR) spectral bands. Source (ESA, 2018b).	22
Figure 4.3: The raw image (a) of the Sentinel-2 containing agricultural and non-agricultural fields. In between the fields showing some objects like road, water, ditches and so on. The ground truth (b) containing white color lines between the fields. These lines represent those such as objects like road, water, ditches.....	23
Figure 4.4: Ground truth sample containing the labels 1, 2, 3 and representing AFB, crop, grass and other respectively.....	24
Figure 4.5: Location of the tiles in Flevoland using Sentinel-2. TR represent tiles for training and TS represent tiles for testing the network.....	26
Figure 4.6: 10 tiles representing the ground truth for both training and testing the network. 5 ground truth used for training represented as GT_TR and 5 ground truth used for testing represented as GT_TS	27
Figure 5.1: The left image is a study area, Flevoland, from the Sentinel-2 with the RGB false colour composite of 843. The red box presents the part of the study area that was analysed by ArcGIS for the clarity of AFB Information. The right top and bottom images from the right side represents single band of 20 m and 60 m spatial resolution respectively. The figure illustrates that, there is no much boundaries information from the 60 m resolution band.	28
Figure 5.2: TR1 (training) and TS1 (testing) are the tiles of 800×800 pixels cropped from Sentinel-2 image. These tiles were used for all preliminary experiments.	29
Figure 5.3: GT-TR1 and GT-TS1 are fully labeled references corresponding to TR1 and TS1 tiles respectively.....	29
Figure 5.4: The AFB maps of TS1 from training sample size analysis: a, b and c are the output maps from the FCN-Net5 using training sample sizes of 500, 1000, and 2000 respectively. Yellow color represents the AFB.....	30
Figure 5.5: AFB reference (a) and output map from FCN-Net of N=5 architecture without incorporating max-pooling layers (b) for the TS1: The map is an output with an F-Score accuracy of 0.41. Yellow colour represents AFB.	31
Figure 5.6: Final AFB Maps at 10 m resolution from 10 m resolution of Sentinel-2 image with their corresponding reference of 10 m resolution: The boundaries are represented by yellow color.....	43
Figure 5.7: Final AFB Maps at 5 m resolution from 10 m resolution of Sentinel-2 image with their corresponding reference of 5 m resolution: The boundaries are represented by yellow color.....	45

Figure 5.8: AFB maps of 10 m spatial resolution from Sentinel-2 image using a Deep_FCNet (a) and MS-FuseNet (b) with their reference map of 10 m resolution (c).....47

Figure 5.9: AFB maps of 5 m spatial resolution from Sentinel-2 image using SRM-Net (a) and SRM+LRM-Net (b) with their reference map of 5 m resolution (c).....47

Figure 5.10: AFB output maps from SRM+LRM-Net (a) and output map from nearest neighbour-based resample of output of Deep_FCNet at 10 m resolution (b).....48

Figure 5.11: The AFB output maps at 10 m and 5 m resolution from Deep_FCNet (a) and SRM+LRM-Net respectively. Oval shape in (b) illustrates precise location compared to oval shape in (a). Rectangle in (b) illustrates better separation of the boundaries with the small distance of approximately 20 m than in (a).....49

LIST OF TABLES

Table 3.1: Components of the F-Score accuracy measure.....	20
Table 4.1: RapidEye Satellite Image Spectral Information	22
Table 4.2: Software.....	25
Table 5.1: Hyperparameters for training sample analysis.....	29
Table 5.2: Hyperparameters for pooling layer analysis.....	31
Table 5.3: Band combination.....	31
Table 5.4: The FCN-Net of $N = 5$ architecture.....	32
Table 5.5: Results of the class AFB for the experiments on band combinations	32
Table 5.6: Hyperparameters for filter size analysis.....	33
Table 5.7: Results of the class AFB for the experiments on filter size	33
Table 5.8: Hyperparameters for patch size analysis	33
Table 5.9: Results of the class AFB for the experiments on Patch Size.....	33
Table 5.10: Optimal hyperparameters	34
Table 5.11: Network architecture of exp1 using FCN-Net of $N = 3$ and $M = 2$	35
Table 5.12: Deep_FCN-Net architectures.....	35
Table 5.13: The F-Score results of AFB class for Deep_FCN-Net architectures	36
Table 5.14: MS-FuseNet architectures	36
Table 5.15: The F-Score results of AFB class for MS-FuseNet architectures.....	37
Table 5.16: SRM-Net architecture for Case-1	38
Table 5.17: The F-Score results of AFB class for SRM-Net architectures Case-1 (a).....	38
Table 5.18: The F-Score results of AFB class for SRM-Net architectures Case-1 (b)	39
Table 5.19: SRM-Net architecture for Case-2.....	38
Table 5.20: The F-Score results of AFB class for SRM-Net architectures Case-2	39
Table 5.21: SRM-LRM-Net architectures	39
Table 5.22: The F-Score results of AFB class for SRM+LRM-Net architectures	40
Table 5.23: F-Score accuracy of the experiment from FCN-DK5 with binary classes.....	41
Table 5.24: Results of the final experiment using Deep_FCN-Net.....	41
Table 5.25: Results of final experiment using MS-FuseNet	41
Table 5.26: Results of the final experiment using Deep_FCN-Net from RapidEye.....	43
Table 5.27: Results of the final experiment using SRM-Net	44
Table 5.28: Results of the final experiment using SRM+LRM-Net.....	44
Table 5.29: F-Score accuracy comparison for Deep_FCN-Net and FCN-DK5.....	46
Table 5.30: The F-Score accuracy comparison for Deep_FCN-Net and MS-FuseNet methods	46
Table 5.31: The F-Score accuracy comparison for SRM-Net and SRM+LRM-Net.....	47
Table 5.32: The F-Score accuracy comparison for Deep_FCN-Net form RapidEye and SRM+LRM-Net48	

1. INTRODUCTION

1.1. Background

Boundaries of agricultural fields are very important features necessary for defining the agricultural units from which aggregate statistics and other information about agriculture and its associated biophysical characteristics can be gathered. This information includes production, size of the field, type of soil, soil fertility and taxation. The agricultural information and statistics are important indicators to monitor agriculture policies and developments; thus they need to be up-to-date, accurate and reliable (Ji, 1996). Mapping the spatial and temporal distribution and characteristics of agricultural fields are paramount for their effective and sound management.

According to the Central Agricultural Policy (CAP) of European Commission, (2017), agricultural area refers to land suitable for agricultural practices, which include arable land, permanent cropland, and permanent grassland. Agricultural field boundaries can be conceptualized as a natural disruption that partition location where the change of crops types occurs, or comparable crops naturally detach (Rydberg & Borgefors, 2001). Traditionally, agricultural field boundaries were established through surveying techniques, and the techniques are laborious, costly and time-consuming.

Currently, the availability of very high-resolution satellite imageries (VHR) and advancement in spatial and computer vision have shown potential advantages for the detection and delineation of agricultural field boundaries (Musyoka, 2018). Notwithstanding the benefits of VHR for the detection and delineation of agricultural field boundaries, these images are expensive, and their analysis requires powerful computers capable of handling big data. While spatial resolutions of the free available satellite imageries (such as Landsat) are coarse, hence they may not suitable for the detection of agricultural field boundaries, the potential of freely available Sentinel-2 which is a medium resolution has been cited in recent studies on classification. Sentinel-2 images have been used for crops classification among other agricultural applications (Sonobe et al., 2018; Belgiu & Csillik, 2018; Lebourgeois et al., 2017). The images convey the state-of-the-art with relatively good spectral information (13 spectral bands) and medium spatial resolution of 10 m (ESA, 2018b). However, their potential for the detection and delineation of agricultural boundaries have not been comprehensively studied.

Spatial contextual features form the basis for the detection of agricultural field boundaries based on spectral similarity and context. Commonly, standard edge detection methods such as a Canny detector, and classification methods such as Object-Based Image Analysis (OBIA) and Random Forest have been applied to extract agricultural fields boundaries (Turker & Kok, 2013; García-Pedrero et al., 2017; Debats et al., 2016). Nevertheless, some of the methods such as Object-Based Image Analysis (OBIA) and Random Forest incorporate spatial contextual features but without automatically learning the features directly from the input by integrating training of the classifier and feature extraction. Unlike such methods, deep learning methods, such as Convolutional Neural Networks (CNN) and Fully Convolutional Network (FCN) are pixel-based methods which can automatically learn the spatial-contextual features from the input images and precisely generate the required outputs (Zhou & Li, 2017). In remote sensing, CNN and FCN have been recently used for land cover land use classification (LCLU) (Yalcin & Razavi, 2016; Andrea et al., 2017; Zhou & Li, 2017). Despite the wide applications of the methods in LULC classification, their capabilities for boundary detection has not been fully explored, and they are limited to VHR images (Musyoka, 2018). This study extends the application of FCN for the detection of agricultural field boundaries using free medium resolution Sentinel-2 image.

Often, the image inputs of FCN architecture have the same spatial resolution. Only recently, Bergado et al. (2018) introduced the FCN that accept VHR image with different spatial resolution (panchromatic and multispectral bands) and fusing them within the network. Similar to this fusion technique within FCN architecture is designed in this study to detect the boundaries of agricultural fields using Sentinel-2 images, however different from Bergado et al. (2018), the architecture fuses selected relevant multispectral bands irrespective of their spatial resolutions (10 meters, 20 meters, and 60 meters) and produce the output for boundary detection at 10 meters spatial resolution. We refer to this new network as MS-FuseNet. We also, develop the boundary detection method (Deep_FCNet) at 10 meters spatial resolution, where the multiresolution bands are fused outside the network. Then, we compare these Deep_FCNet and MS-FuseNet methods.

The resolution of 10 meters is relatively coarse to map the boundaries of the agricultural field. This is because to detect the small and thinner objects like boundaries, using coarse resolution may contribute to loss of the exact location. However, the finer resolution may do the best. Often, super-resolution mapping (SRM) is applied to increase the spatial resolution of the thematic map using different techniques, two-point histogram (Atkinson, 2008), Markov-random-field-based super-resolution mapping (Tolpekin & Stein, 2009), and Hopfield Neural Network (Heltin Genitha & Vani, 2010). In this study, we investigate SRM within the FCN network, and its performance compared with the results obtained from a higher resolution of 5 m from the RapidEye image. Additionally, we apply the contextual spatial regularization to examine the improvement in the spatial resolution of the output maps of 5 m resolution.

1.2. Research Identification

In this study, we address the agricultural field boundary (AFB) detection from open and free Sentinel-2 image as a classification problem of class AFB, crop, grass and other. We develop deep learning network architectures based on FCN to detect boundaries and to identify which bands of Sentinel-2 are relevant for solving this problem. Additionally, we investigate SRM and assessing how much it helps to enhance the spatial resolution of the boundary's maps from 10 m to 5 m. Furthermore, we perform the spatial contextual regularization by reusing the SRM network and refining the AFB labels.

1.2.1. Research Objectives

The main objective of this research is to develop fully convolution networks that detect agricultural field boundaries from the Sentinel-2 image and, to investigate SRM to provide finer-resolution agricultural field boundary maps.

The specific objectives are:

1. To carry out a literature review on deep learning methods (e.g., CNN and FCN), boundary delineation, image fusion, and SRM techniques.
2. To design and implement fully convolutional network architectures for detection of agricultural field boundaries by considering multiresolution multispectral bands of Sentinel-2 image.
3. To investigate SRM for increasing the spatial resolution of the field boundary maps.
4. To explore the possibility of regularizing the obtained agricultural field boundary maps.

1.2.2. Research Questions

The following research questions will facilitate achieving the above-mentioned objectives:

1. How does FCN work based on the available literature in land cover land use (LCLU) classification?
2. (a) Which band(s) of Sentinel-2 image are relevant for agricultural field boundary detection?
(b) How can we properly fuse multiple bands within a single convolutional network trained end to end?
(c) What are the optimal hyperparameter values of the network architectures for AFB detection?
3. (a) How can we design a deep learning network to properly up-sample feature maps at a higher resolution?
(b) What level of accuracy determines the stability of designed SRM network?
4. How can we incorporate spatial regularization in the deep learning network?

1.2.3. Research Innovation

The novelties of this study are:

- to design two methods for detecting agricultural field boundaries
 - An FCN network that uses multiresolution multispectral bands of Sentinel-2 that are fused outside the network.
 - A single FCN framework that fuses different bands from the same scene of multiresolution multispectral Sentinel-2 satellite image and detect agricultural field boundaries simultaneously.
- to investigate SRM using convolutional network to obtain finer-resolution agricultural field boundary maps.
- to regularize the AFB labels from the SRM network.

2. LITERATURE REVIEW

This chapter describes the deep learning methods and their applications in LCLU classification, and it also reviews the available literature on boundary delineation. Additionally, some methods used to fuse the bands of panchromatic and multispectral images are discussed. Finally, this chapter discusses different techniques of SRM mapping that have been explored.

2.1. Deep Learning

Deep learning in image analysis is a collection of machine learning techniques based on algorithms applied to learn and visualize the features such as edges and curves from a given input image. In image analysis, CNN and FCN are well known deep learning methods. Both methods are made up of several layers to optimize the objective function. These layers including convolution, pooling, dropout, batch normalization, and non-linearity. Convolution layers perform the main operation in the network structure. These layers convolve the input to learn its feature representations using learnable filters which are in the dimension of $f \times c \times k$, where by f is the size of the filter kernel, c is the number of the bands of an input and k is the number of filters that are looking for the patterns in the image pixels. Also, these layers contain stride and zero padding parameters as they are controlling the size of the output which is known as feature maps. Another layer is the non-linearity layer which is applied after every convolution layer as it overcomes the problem of a linear classifier. There are different types of non-linearity such as rectified linear unit (ReLU), sigmoid, *tanh* and others. In most cases, ReLU is used because is more robust to train the network compared to the other non-linearity (Maas, Hannun, & Ng, 2013). Pooling layer is usually used to reduce the spatial dimensionality, and it is usually placed between consecutive convolutional layers. Most common pooling layer is a max-pooling which takes the largest value and discard the other values from the defined filter. Additionally, pooling layer can also be an average pooling instead. Pooling layers need spatial extent which is a square filter and stride as their parameters.

Another important layer is batch normalization. This layer is used to reduce covariance shift and speeds up the training process. The shift caused by the input distribution change such that the input for the next layer is an output of the previous layer. Therefore, training the network becoming slower as it is working with the lower learning rates and better parameter initialization (Ioffe & Szegedy, 2015). Dropout layer is an another layer that is used to avoid overfitting and to balance both accuracies obtained after training and testing the network for better performance. Overfitting is a common problem in deep learning that comprising more convolutional layers. To address this problem, it is better to incorporating a dropout as it is a simple and prevent too much co-adaptation by dropping out unit randomly in a neural network (Srivastava et al., 2014).

The main difference between CNN and FCN is that in CNN there are fully connected layers while in FCN fully connected layers are replaced by the convolution layers. The uses of CNN for the automatic extraction of the spatial contextual features investigated in (Bergado et al., 2016). In their study, they analyzed the network hyperparameters. The architecture was designed in a patch-based approach. CNN learned the spatial contextual features from 3D input volume, and the convolution applied 16 filters of size 19 for best accuracy. Therefore, the final output was a one-dimensional vector which was connected to the fully-connected layer. Also, Persello & Stein (2017) applied dilated convolutions (DK) with six convolutional layers to learn the spatial features capturing long-range pixel dependencies. In their research, the FCN-DK6 architecture was used for increasing spatial support and maintaining the numbers of network parameters. The size of the kernel was set equally corresponding to the DK support, and its values were learned in a supervised manner to minimize a loss function. Also, max-pooling using stride factor of 2 was applied to return maximum value within a filter.

2.2. Boundaries Delineation

Feature detection and extraction using remote sense datasets offer several benefits, e.g., automated analysis of remotely sensed images is less expensive and can be applied to cover large areas. Turker & Kok (2013) delineated boundaries of the agricultural field from SPOT4 XS and SPOT5 XS imageries using a Canny edge detector. This is a standard method that only detects intensity discontinuity. García-Pedrero et al. (2017) also delineated small and heterogeneous agricultural land from the WorldView-2 satellite image by applying the segmentation algorithm. Debats et al. (2016) differentiated the mixed agricultural fields from the WorldView-2 imageries by using supervised and non-parametric machine learning random forest classifier.

Turker & Kok (2013) extracted agricultural fields boundaries in the north-west part of Turkey from remote sensing data using perceptual grouping. They applied a Canny edge detector to detect the edge pixels. The challenge was the mismatched of the detected boundaries from SPOT 5 which had higher accuracy than SPOT 4 due to the undetected of some boundaries by SPOT 4. This is because of the coarse resolution of the SPOT 4 and the lower within-field variation than SPOT 5. Agglomerative segmentation is another algorithm applied by García-Pedrero et al. (2017) to delineate the agricultural area in Chilean Central Valley. A high-resolution image of WorldView-2 with 2.4 m spatial resolution used in the study. The algorithm considered over-segmentation and this problem was tackled by dividing the homogeneous regions based on the number of pixels and merged the superpixels to came up with large regions. Debats et al. (2016) also used computer vision and random forest classifier to extract agricultural fields. The algorithm portrayed that, for both panchromatic and multispectral bands, mean area under the receiver operating characteristics curve was 0.91 and square error was 0.02. The study showed that the performance was highly determined by multitemporal than the multispectral bands available in the Worldview-2 imagery. Musyoka (2018) applied FCN in his thesis work to delineate the farm boundaries in the Northern part of Nigeria, Kofa village in Kano state from Very High Resolution (VHR) satellite images WorldView-3. FCN showed better results during the experiment compared with other methods like image segmentation and Canny detection.

2.3. Image Fusion

Image fusion is a pre-processing step in image analysis. The panchromatic has higher spatial resolution than multispectral images. The image fusion allows the spectral information from the multispectral bands with a high spatial resolution of panchromatic to obtain a new dataset. Shao & Cai (2018) practiced pan-sharpening technique with adopted deep learning in image fusion. They introduced two branches of structure and residual learning to achieve the higher-resolution to the multispectral image and retain its spectral resolution. In this method, it was difficult to train the network and inconvenience of performance when the labels do not exist. Bergado et al., (2018) on the other hand fused the panchromatic and multispectral bands of VHR image of Worldview-03 within the FCN.

A Sentinel-2 image has different multiresolution bands in which four 10 m resolution bands considered as fine bands. Wang et al. (2016) applied an extension of pan-sharpening, and area-to-point regression kriging (ATPRK) approaches to fuse four bands of 10m and six 20m resolution from Sentinel-2 image. In extended pan-sharpening, a single band of 10m resolution extracted from four 10m bands. This process synthesized by a linear combination whereby the weights of bands calculated based on a multiple regression model. On the other hand, all four 10m resolution bands were synthesized in ATPRK.

2.4. Super Resolution Mapping (SRM)

Super-resolution mapping (SRM) is a technique applied to increase the spatial resolution of a thematic map from course resolution input. Different techniques explored and applied as SRM. Tolpekin & Stein (2009)

explored Markov-random-field-based super-resolution mapping to overcome the image classification difficulties of fewer spectral bands and high spectral values variation within the class. Also, Atkinson (2008) used a two-point histogram to solve a classification problem at high spatial resolution from both multispectral and panchromatic image at a coarse and intermediate spatial resolution respectively. He performed classification on the coarse resolution image of 30 m (Landsat Thematic Mapper) while the training conducted from a high-resolution image of 4 m (IKONOS).

3. METHODS

For the LCLU classification, CNN and FCN mostly work with raster images from very high-resolution (Bergado et al., 2018; Bergado et al., 2016; Persello & Stein, 2017). In this study, we investigate FCN-DKs from Sentinel-2 multiresolution multispectral image to extract AFB features by developing Deep_FCNet and MS-FuseNet methods. We assess the performance of boundary detection from the Sentinel-2 using these MS-FuseNet and Deep_FCNet methods by fusing multispectral bands of different resolution within the network and outside the network respectively. We develop SRM-Net to investigate SRM within FCN-DKs from Sentinel-2 to come up with a higher resolution of 5 meters. We use the Sentinel-2 image because it has the advantages of being freely available and has high spectral bands. Additionally, we design SRM+LRM-Net to perform contextual spatial regularization for the AFB outputs improvement. In all these methods, we use satellite image of Sentinel-2 to both processes of training and testing the networks. The following sections describe the mentioned methods.

3.1. Deep FCN-DKs Networks (Deep_FCNet)

We organize the FCN layers with dilated kernels sequentially. These layers are used to convert the input image into the output map (AFB). Any number of bands with the same spatial resolution from Sentinel-2 image can be an input data. This data received by the network as patches using a first convolutional layer and they are convolved by the filters of a convolutional layer and added a bias. A convolutional filter is a set of learnable weights which is represented as an array of $f \times c \times k$, where by f is the size of the filter kernel, c is the number of the bands of an input and k is the number of filters. The output from the first convolutional layer is a feature map which is the input in the next convolutional layer. The size of the feature map controlled by the stride factor (s), zero padding (p) and dilation factor (d). The parameter s controls sliding of the filter to convolve the input, in this network, the filters sliding an input via one unit because it is fixed by a factor $s = 1$. The parameter p is using to add zero to the boarder of an input, and parameter d increases the original filter by inserting the spaces between the filter elements so that to keep the feature maps the same size as the input and to enlarge the field of view, therefore, we set p equal to d . In this network, neither down-sampling nor up-sampling are applied. Non-linearity function follow the convolutional layer and batch normalization layer is applied after every convolutional layer. The series of convolutional layers, batch normalization and ReLU operated before the last convolutional layer which is a classification layer. The classification layer produces four (4) feature maps which are the number of classes, and they are fed to the soft-max module for the prediction as an output map.

In general, the architecture of the Deep_FCNet made up of the number of network blocks (N). The N represents the dilation factor for each convolutional layer contained in N . For example, if the network block, $N = 3$, this network contains dilation factor $d = 3$ and the filters for all convolution layers within this block were dilated by a factor of 3. Moreover, each network block (N) can contain sub-blocks (M) that consists convolution layers, with filter size $f \times f$, batch normalization (BN) layers and non-linear activation function (ReLU). We illustrate the general architecture of the Deep_FCNet in a Figure 3.1

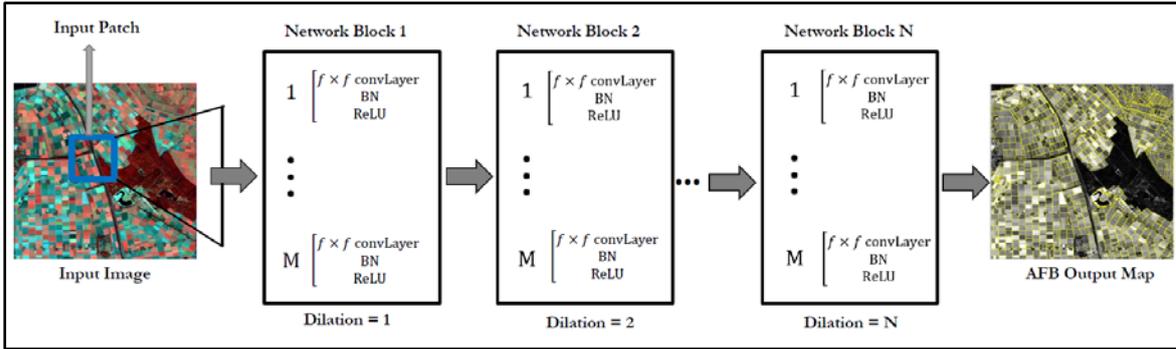


Figure 3.1: General Network architecture for the Deep_FCNet: Input image can be any number of channels (bands) of the same resolution. In this study, we use the Sentinel-2 band combination of the 10 m spatial resolution. The output map has the same spatial resolution as an input image. Yellow lines represent the boundaries.

3.2. Fusion Networks for Sentinel-2 Multispectral Bands Within the Network (MS-FuseNet)

We design fusion network for the multispectral bands of Sentinel-2 (MS-FuseNet) such that to fuse the multiresolution image bands of the Sentinel-2 within the network and performing AFB detection (as the classification problem) at the same time. Both 4 visible and NIR bands and 4 vegetation bands with 10 m and 20 m resolution respectively are used in an MS-FuseNet. Ideally, both input image bands should overlap in the same area. These image bands of the different resolution are received using separate input streams. The input streams are the network blocks of $N = 1$ with several sub-blocks. Both network blocks receive the input using the first convolutional layer as in a Deep_FCNet, then, ReLU and batch normalization layers followed. The series of these layers continue within these network blocks as the sub-blocks of N . Then, we apply a down-sampling operation to the 10 m resolution input such that its spatial dimension of the feature maps matches with the 20 m resolution. Then, these transitional feature maps are concatenated using concatenation layer and then the network blocks (N) with sub-blocks (N) continue to the concatenated feature maps. Lastly before classification, we use up-sampling layer such that the feature maps are up-sample to the highest resolution of 10 m. Figure 3.2 shows the general architecture of the MS-FuseNet from the Sentinel-2 applied on 4-bands of 10 m and 4-bands of 20 m resolution.

Down-Sampling and Up-Sampling Operations in MS-FuseNet

In this study, we use the down-sampling operation to extract and infer the spatial context of 10 m resolution input image by reducing the spatial dimension of the feature maps. The main aim of this down-sampling is to match these feature maps with the ones from the 20 m resolution input image. Therefore, we perform this operation using a layer with a square filter of 2×2 and stride of factor 2 for spatial extraction and spatial size reduction respectively. We perform the down-sampling operation using the common technique which is a max-pooling. We place max-pooling layer once after the first convolutional layer that takes a 10 m resolution as an input image. Springenberg et al. (2015) did experiments and reported that down-sampling could also be performed using a convolutional layer with the stride of the factor greater than one without loss of accuracy. We also, therefore, perform down-sampling to the first convolutional layer. This convolutional layer made by 3×3 filter and stride of factor two.

We use the up-sampling operation to increase the spatial resolution of the input image. Nearest neighbor, bilinear and bicubic interpolation are the ones among the up-sampling operation. However, the neural network cannot learn the features using these operations. A transposed convolution which is also an up-sampling operation is a layer in a neural network. This layer is made up of filter and up-sample factor to learn the spatial context and up-sample the feature maps respectively. In MS-FuseNet, we apply the transposed convolution layer of the filter size 4×4 and an up-sample factor of two to up-sample the feature

maps from 20 m back to 10 m. The up-sample factor of two is a stride of the factor $\frac{1}{2}$. Because transposed convolution has a learnable parameter to learn the spatial contextual features, it should initialize its weight either randomly or using any interpolation method. In this study, we initialize the transposed convolutional weight randomly and using bilinear interpolation method.

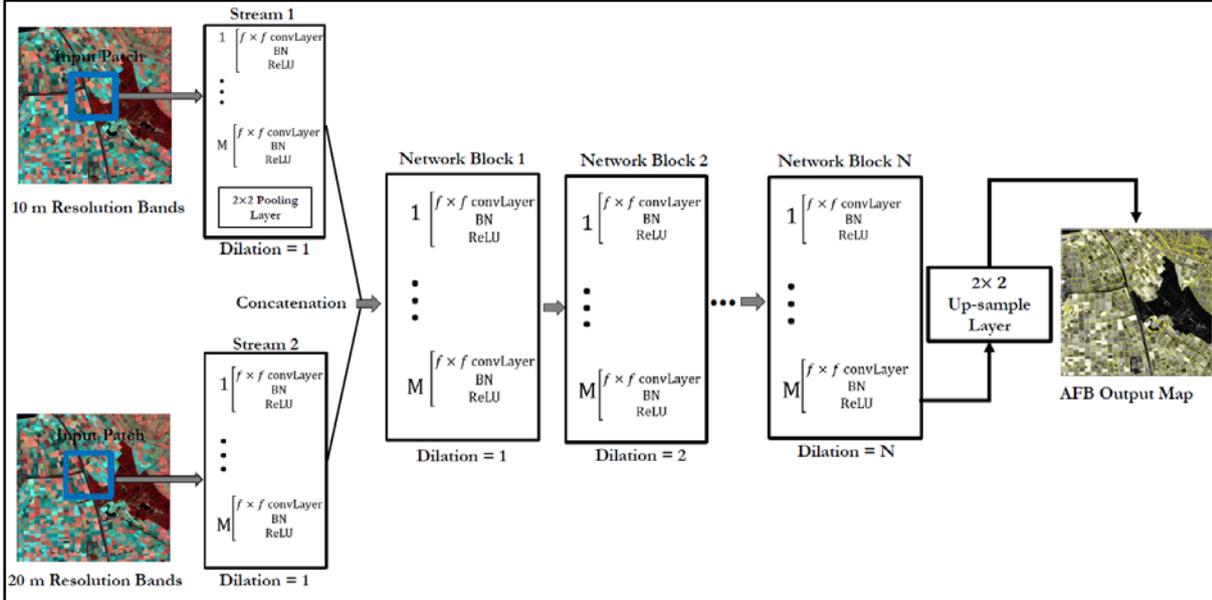


Figure 3.2: General Network Architecture for the MS-FuseNet: In this study, we use two inputs. The inputs are the 4-band combination of 10 m and 4-band combination of 20 m resolution from the same image of Sentinel-2. The output map has a spatial resolution of 10 m.

3.3. Super Resolution Mapping Network (SRM-Net)

The main aim of SRM in this study is to investigate the possibility of enhancing the resolution of the output maps to the 5 m resolution by developing SRM-Net. We apply SRM technique to the medium multispectral Sentinel-2 image at 10 m resolution so that to increase the resolution of the output maps. In this network, the 8-band combination of 10 m resolution from Sentinel-2 are received as input patch using the first convolutional layer, then the series of layers operated as in a Deep_FCNet. The different of this network from the Deep_FCNet is that, in SRM-Net we incorporate an up-sampling layer within the network. We use a transposed convolutional up-sampling layer of the filter size 4×4 and an up-sample factor of two. We initialize the learnable weight parameter of transposed convolution using two techniques; randomly and using bilinear interpolation. We apply this transposed convolution layer to enhance the spatial resolution of the AFB feature maps from the Sentinel-2 input at 10 m resolution. We use this operation, to allow feature learning within the network and enable pixel-wise prediction to the feature maps at 5 m resolution. Figure 3.3 illustrates the architecture of SRM-Net.

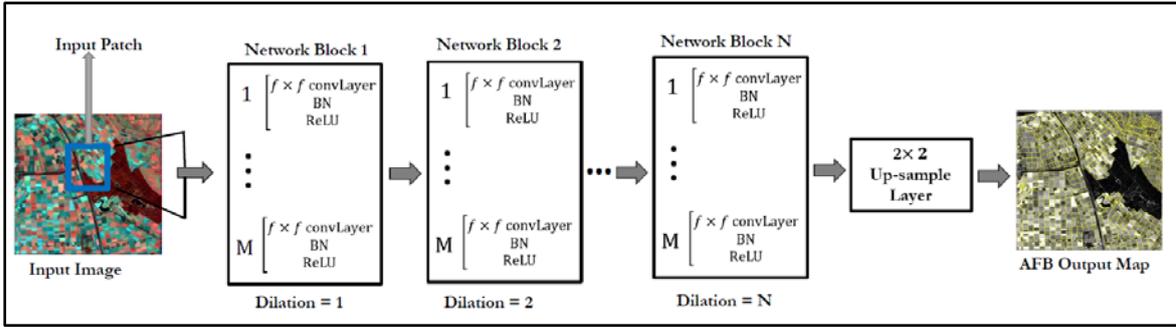


Figure 3.3: SRM-Net architecture. The input image is a band combination at the 10 m resolution, and the output map is at 5 m resolution.

3.4. Label Refining Modules Network (SRM+LRM-Net)

Having noise and mixed pixels between AFB class and other classes is possible during classification. To overcome this problem, we design a network that regularizes the spatial contextual features by refining the AFB labels from SRM-Net. We name this network SRM+LRM-Net. SRM+LRM-Net consists of two predictions instead of making one prediction at the end of the network architecture. The first prediction is obtained from SRM-Net which is the results from learning the contextual features of the input image within the network, and the second prediction obtained after operating additional series of convolution to the SRM-Net output (first prediction). The additional convolutional layers learn the features from label space of AFB from SRM-Net, and hence they filter the noise and increase the capability of separating two parallel boundaries from other fields. Generally, we improve the output maps using this network as it is used to learn and extract the features from both input and label space. Figure 3.4 illustrates the architecture of the SRM+LRM-Net.

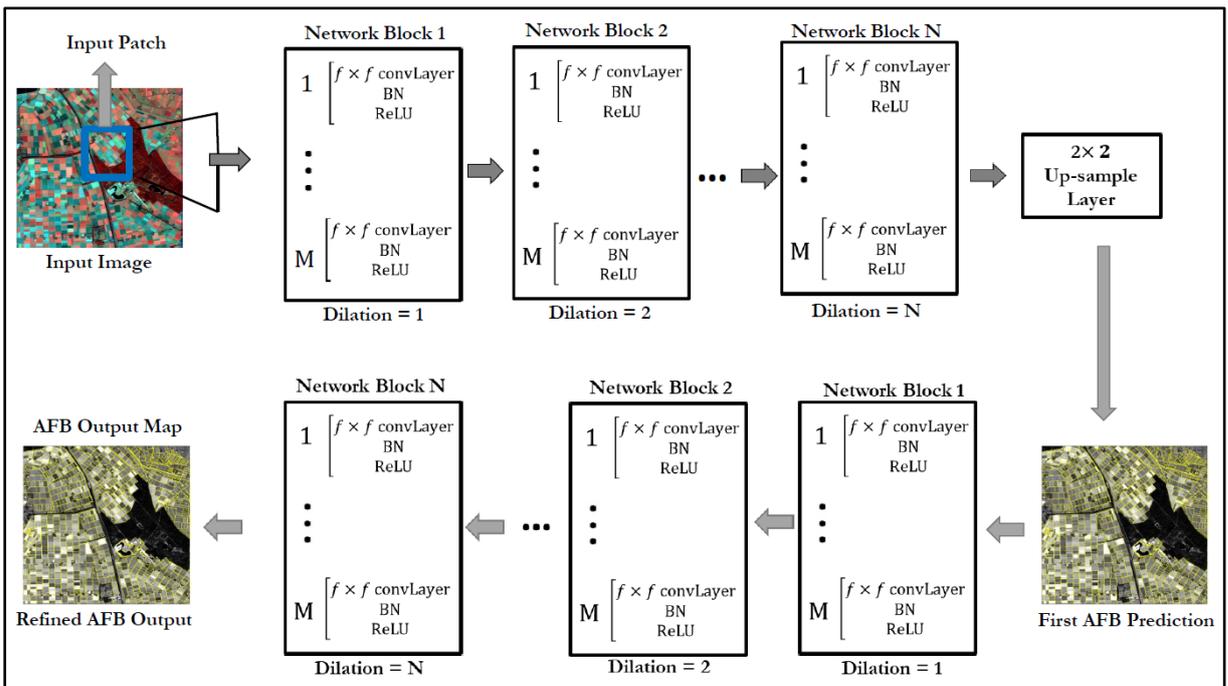


Figure 3.4: SRM+LRM-Net Architecture.

3.5. Accuracy Assessment

In this study, we use the F-Score accuracy measure for all experiments to evaluate the performance of the methods. F-Score is an accuracy measure that computes a harmonic average of precision (p) and recall (r). It ranges from 0 to 1 and tends to be at best when it approaches 1. Precision is a measure of how close the results are to the expected result. It determines the quality of the result. Recall on the other hand is a measure concerned with the level of details that are generated. It looks at the quantitative aspect. In this study we are interested in both the qualitative and quantitative measure therefore we used the harmonic mean, the F-Score. This accuracy assessment measure is used by the network to evaluate the performance of the AFB output maps with their corresponding reference. For the AFB output map, this performance measure calculated based on the four terms which are true positive, true negative, false positive and false negative as described in Table 3.1.

Table 3.1: Components of the F-Score accuracy measure.

F-Score Term	Description of AFB Performance
True Positive (T_{pos})	AFB pixels values correctly predicted
True Negative (T_{neg})	AFB pixels values correctly rejected
False Positive (F_{pos})	Other classes pixels predicted as AFB class pixels which are Type I error (α)
False Negative (F_{neg})	AFB class pixels predicted as other classes pixels which are Type II error (β)

Precision is the number of true positive predicted pixels of AFB divided by the number of all positive pixels returned by the network.

$$p = \frac{T_{pos}}{T_{pos} + F_{pos}}$$

Recall is the number of true positive predicted pixels of AFB divided by all pixels that should have been identified as positive.

$$r = \frac{T_{pos}}{T_{pos} + F_{neg}}$$

Type I Error, and Type II Error represented using α and β equations respectively.

$$\alpha = \frac{F_{pos}}{F_{pos} + T_{neg}}$$

$$\beta = \frac{F_{neg}}{T_{pos} + F_{neg}}$$

F-Score is the harmonic average of precision (p) and recall (r). Therefore, F-Score is expressed by,

$$\text{F-Score} = 2 \frac{p \cdot r}{p + r}$$

4. DATASET AND SOFTWARE

In this chapter, we describe the study area, dataset, and software. Additionally, we describe the pre-processing that is performed to the dataset and how this dataset is prepared.

4.1. Study Area

Flevoland is one of the 12 provinces of the Netherlands (Figure 4.1). This province located at coordinate $52^{\circ} 30'N$ $5^{\circ} 35'E$, covers an area of approximately 1670 square kilometers which was reclaimed from the sea purposely for the agricultural expansion (Mandryk et al., 2015). In this province, the agricultural fields are of different size, however, they are regular in shape. Also, several crops are cultivated in this province including onions seeds, potatoes, wheat, carrot, barley, Japanese oats, maize, and other cereals (PDOK, 2018).

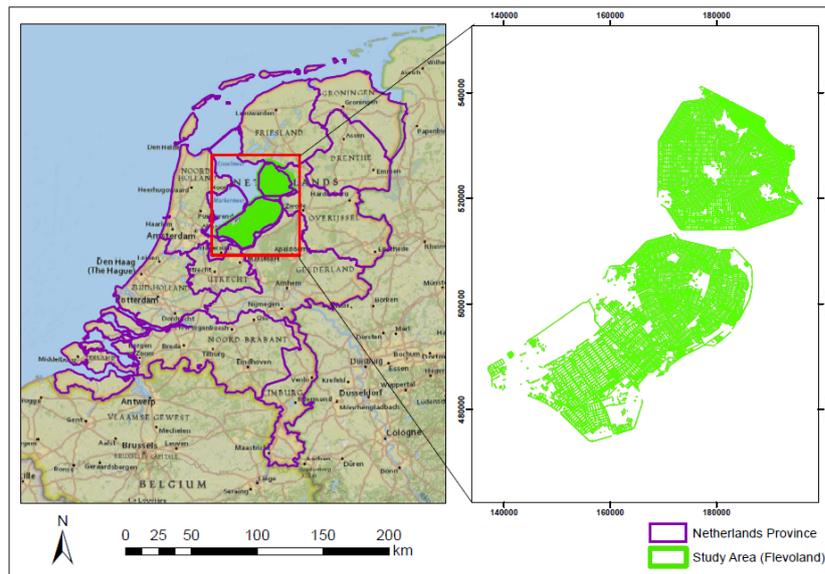


Figure 4.1: The study area Flevoland, the Netherlands.

4.2. Sentinel-2 Satellite Image

We use Sentinel-2 satellite image acquired on 26th September 2016 covering the Flevoland with the geographic reference WGS_1984_UTM_Zone_31N. The data was in level 1C which is not atmospherically corrected. We, therefore, performed an atmospheric correction using SNAP for this data. We generated the level 2A product of Sentinel-2 from the Sen2Cor processor. The process was performed by running the windows installer Sen2Cor-02.05.05-win64 in a command prompt (ESA, 2018a). The image has 13 spectral bands, 4-bands of 10 m (2, 3, 4 and 8 representing Blue, Red, Green and NIR bands respectively), 6 bands of 20 m (5, 6, 7, and 8A representing vegetation bands; and 11 and 12 SWIR bands) and 3 bands of 60 m spatial resolution (bands 1, 9 and 10 for aerosol detection, water vapour and cirrus respectively). Figure 4.2 shows the spatial resolution in meter (m) and spectral bands in nanometre (nm).

4.3. RapidEye Satellite Image

RapidEye data of the level 3B was acquired on 31st August 2016 covering the Flevoland. The purpose of this data is to generate the ground truth of 5 m resolution for accuracy assessment in the SRM network. Also, for checking the stability of the SRM network by comparing the accuracy from SRM+LRM-Net and

Deep_FCNet using RapidEye data. We downloaded RapidEye from the Planet Scope as 16 different tiles. These tiles were already orthorectified and atmospherically, radiometrically and geometrically corrected. Then, we mosaicked the RapidEye tiles using ArcGIS software by performing “blend mosaic operation” from “Mosaic to New Raster Dataset.” In this process we used only 8 relevant tiles to have an image of the area of interest (Flevoland). RapidEye data has 5 bands which are Blue, Red, NIR and Red Edge (RE) and it also has 5 m resolution for each band with ground sampling distance of 6.5 m. Table 4.1 shows the spectral information of the RapidEye image.

Table 4.1: RapidEye Satellite Image Spectral Information

Band Number	Bands	Spectral Range (nm)
Band 1	Blue	440 – 510
Band 2	Green	520 – 590
Band 3	Red	630 – 685
Band 4	Red Edge	690 – 730
Band 5	Near Infrared (NIR)	760 – 850

The spectral information of the RapidEye (Planet, 2016)

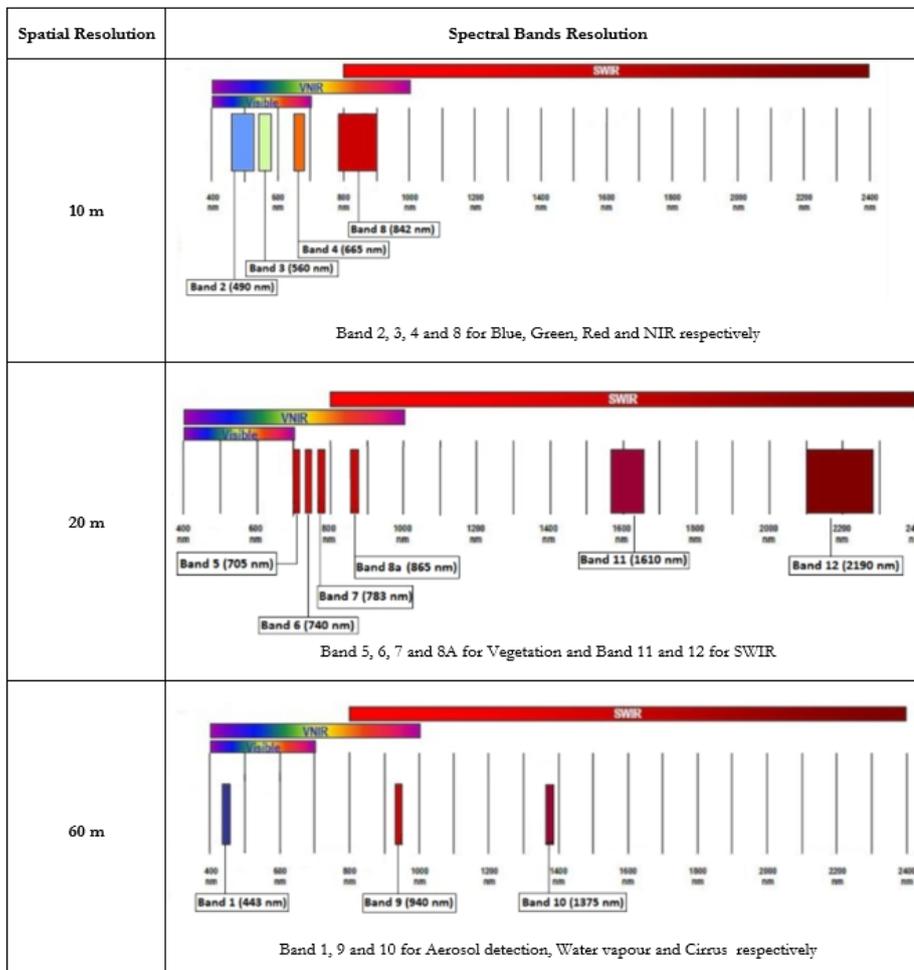


Figure 4.2: The spatial resolution of Sentinel-2 with their corresponding spectral bands. Source (ESA, 2018b).

4.4. Reference Data

We downloaded three datasets from PDOK. PDOK is a Dutch acronym that stands for Public Service On the Card; this is a Dutch government open platform that offers up-to-date geodata. These datasets are agricultural area Netherlands (AAN) shapefile (.shp), basic registration crop parcels (BRP) database (.gdb) and basic registration cadastral (BRK) in a geographic information system (.gml) format. Then, we investigated these datasets using ArcGIS to identify one as a reference dataset in this study. We integrated the Feature Manipulation Engine (FME) platform transformation with the ArcGIS to access gml file (FME, 2018). We transformed the dataset from RD_New to WGS_1984_UTM_Zone_31N geographic reference. Only, the area of interest which is the Flevoland province is clipped from the entire Netherlands for further investigation. For the BRP dataset, the boundaries of the agricultural parcels constructed on the AAN (PDOK, 2018). This BRP dataset fit on the Sentinel-2 data and shows the boundaries with crop types cultivated. Therefore, we selected the BRP as a reference dataset in this study.

4.5. Datasets Preparation

European Commission definition of agricultural area is comprehensive which is within the scope and objectives of this study. Therefore, we redefined the agricultural field as the parcel of land used for crops and flowers cultivation, together with the grass. The grass is a recognized parcel as defined in a PDOK, and it excludes ditches. In this definition of agricultural area, crops and flowers fields are considered as one class, which we classified as crop, this is because they are cultivated in the same seasonal calendar dates and their phenology are somewhat similar. Hence, we contextualized agricultural field boundary (AFB) as the outer extent that define the transition from one agricultural field to another or from one agricultural field to the non-agricultural field which delineates the parcels of agricultural fields from one another. This definition of the boundary does not consider what is on the other side. It is the separation between the agricultural field parcels. Also, in between the agricultural fields, there are some objects like footpaths, roads, ditches and so on as shown in Figure 4.3.

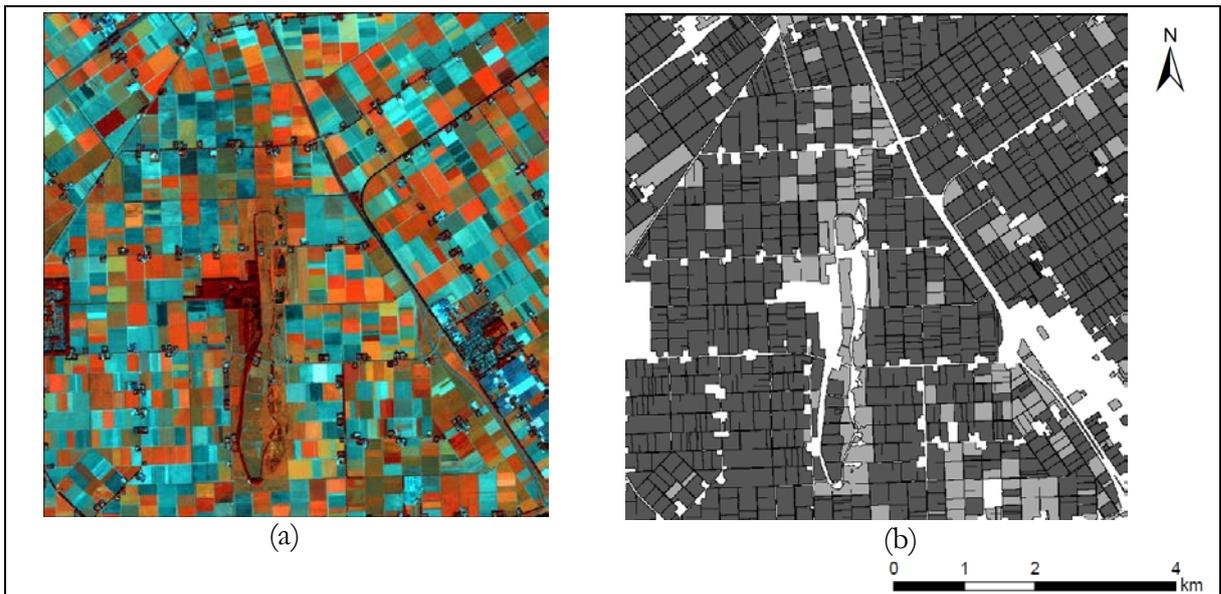


Figure 4.3: The raw image (a) of the Sentinel-2 containing agricultural and non-agricultural fields. In between the agricultural fields showing some objects such as road, water, and ditches. The ground truth (b) containing white color lines between the fields. These lines represent those objects.

We converted reference data, raw BRP dataset from “Polygon To Line” using ArcGIS. We then rasterize the output of BRP data conversion using R-software. In rasterizing, we used 10 m resolution as the finest

resolution of the Sentinel-2 data. The function takes two arguments which are a shapefile (the lines) and a raster (Sentinel-2). It takes the lines and overlays on raster and check which pixel is touched. This process transferred the values associated with the lines spatial data to raster cells that are tangible by their respective lines. Therefore, these pixels are considered as the boundaries and we labelled them as label 1. Through this process, a higher resolution raster data will be able to distinguish two boundaries with a very small distance. On the contrary, if the raster data is low-resolution, it will not be able to distinguish two parallel boundaries if their distance is very small. Moreover, some polygons had holes that represented another feature that limited feature identification. This was solved by excluding the holes using an array technique by taking the first polygon which is bigger and selecting all pixels that are inside. It checks if there is a hole then all pixels inside the hole were given the value 0 otherwise the value 2, 3 or 4 assigned according to the attribute value.

The BRP file consists of five attributes (arable land, grassland, wasteland, natural, and other). The wasteland contains parcels of undetermined lands and uncultivated lands due to the cultivation exemption. Natural land consists parcels of heath which are vegetation but not agricultural field. Another attribute is other that consists of the values of forest which are permanent with replanting obligation and the values of ditches. Arable land consists of crop types and flowers and grassland contain parcels of grass. In the BRP file, grassland attribute was categorized into four types namely; permanent, temporal, agricultural and natural grassland. However, image analysis will not be able to differentiate the grassland that can be used for agricultural and non-agricultural activities, especially for the single date image. Therefore, in this study we categorized all types of these grasses as one class different from crop class and we assigned them different labels. This also will be helpful to know how much of the land is cropland and will count another application apart from AFB detection.

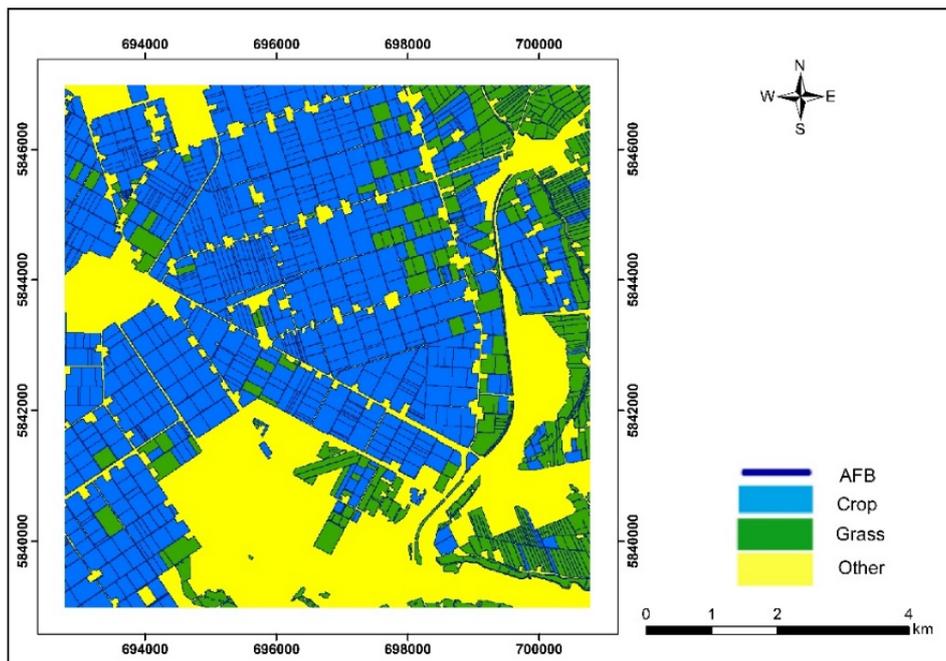


Figure 4.4: Ground truth sample containing the labels 1, 2, 3 and 4 representing AFB, crop, grass and other respectively

In this study, we treat the arable land and grassland attribute as agricultural fields and we gave them values 2 and 3 respectively. Then, we merged wasteland, natural, and other attributes to be one class and named as class other and gave it value 4. In this study, we gave the attribute values to the classes based on the definition of the agricultural field we defined, these values were used for labeling the pixels to train the networks. Therefore, four classes which are agricultural field boundary (AFB), crop, grass, and others with their

corresponding labels 1, 2, 3 and 4 respectively were defined in this study whereas, crop and grass considered as an agricultural field.

We used reference data to create 10 tiles ground truth for training and testing the network. Figure 4.4 shows the ground truth sample, and Figure 4.6 shows 10 tiles of ground truth. Also, from the Sentinel-2 image, we chose ten (10) tiles of the same coverages as the ground truth from the reference data to define the tiles for training and testing the network to detect boundaries from this data. We selected 5 tiles for training and 5 for testing. We cropped the tiles at 10 m and 20 m resolutions with the size of 800×800 and 400×400 pixels, respectively. Figure 4.5 shows the location of the tiles in Flevoland using Sentinel-2. Furthermore, we cropped 10 tiles from the RapidEye with the size of 1600×1600 . We cropped all tiles for ground truth and input images using R software, and ideally, they are overlapping on the same area.

4.6. Software

This section reports the software used as shown in Table 4.2.

Table 4.2: Software

Software	Function
Matlab	FCN implementation using Matconvnet-1.0-beta23 toolbox.
ArcGIS	To prepare the reference data, band combination, to up-sample the Sentinel-2 bands from 20 m to 10 m resolution, to prepare the position of the tiles (top-left scale), to mosaic the tiles of RapidEye image and data presentation (maps).
R software	To prepare the reference dataset using raster packages (Hijmans, 2018), <code>rgdal</code> (Bivand, R., Keitt, T., Rowlingson, 2018) and <code>rgeos</code> (Bivand, R., Rundel, 2018).
SNAP	For atmospheric correction of Sentinel-2 from level 1C to level 2A.

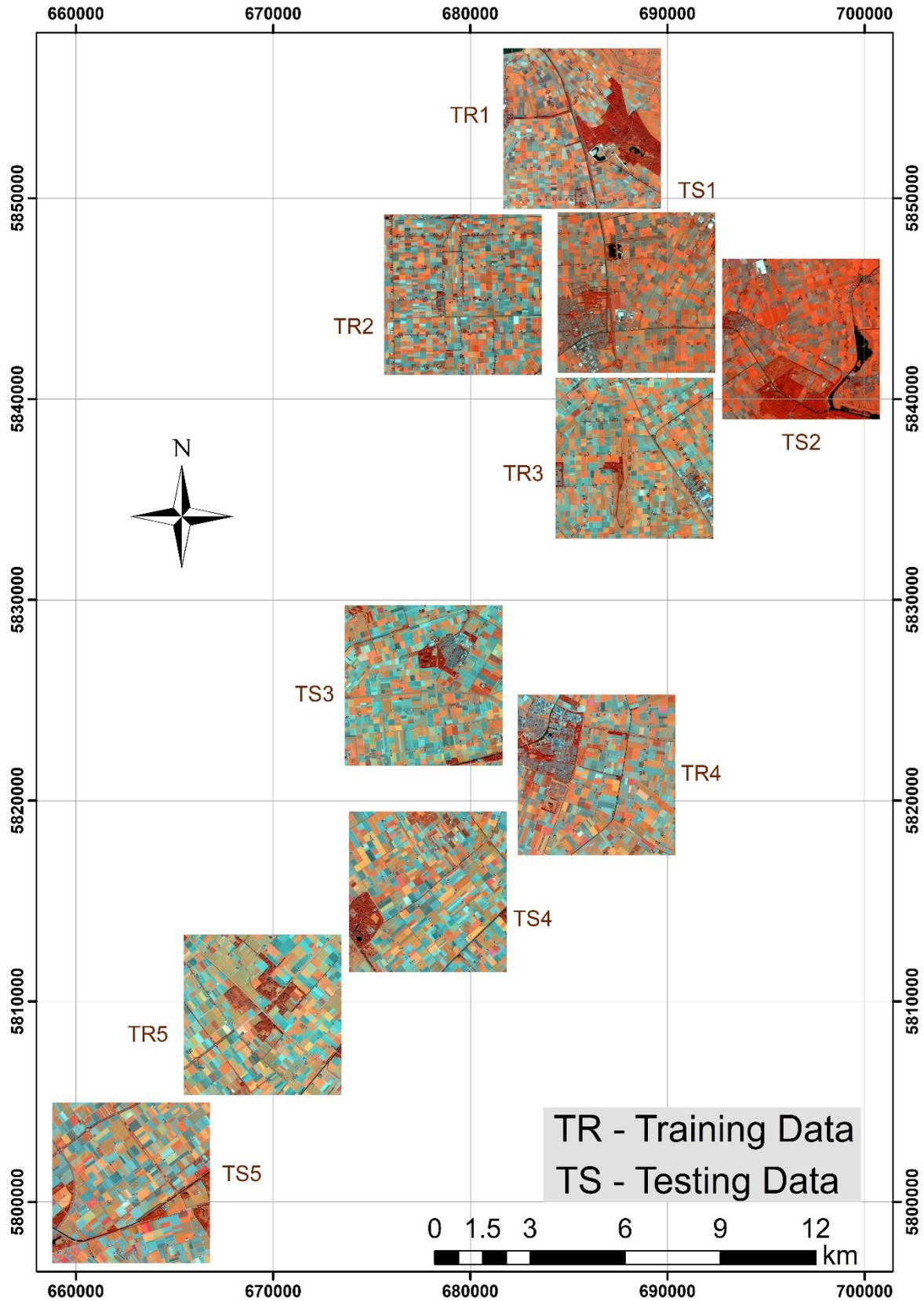


Figure 4.5: Location of the tiles in Flevoland using Sentinel-2. TR represent tiles for training and TS represent tiles for testing the network.

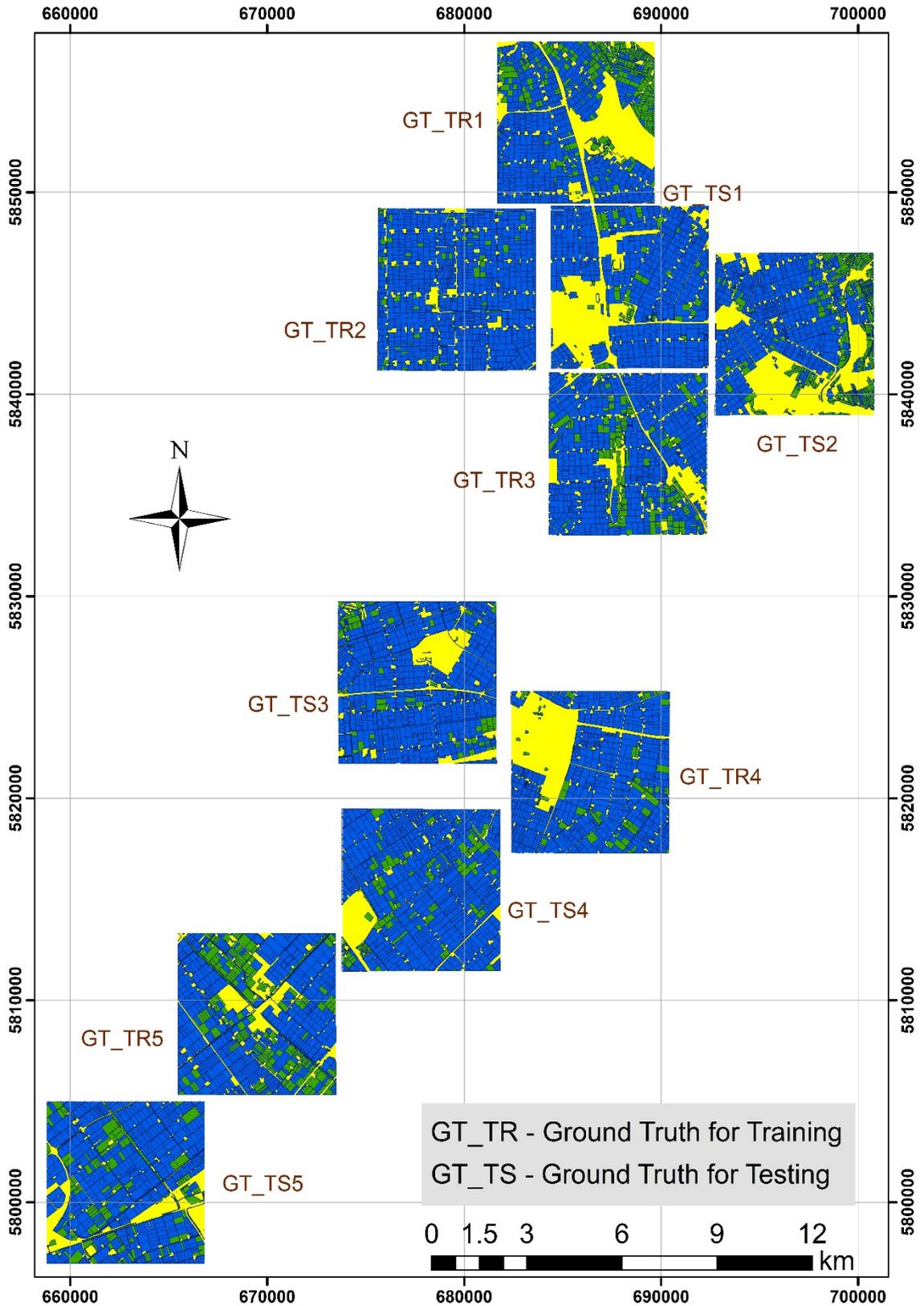


Figure 4.6: 10 tiles representing the ground truth for both training and testing the network. 5 ground truth used for training represented as GT_TR and 5 ground truth used for testing represented as GT_TS

5. EXPERIMENTAL ANALYSIS AND RESULTS

This chapter describes the analysis and results for all experiments we performed. The chapter is divided into five sections. First section describes hyperparameters sensitivity analysis. The second one presents the architectures experiments for the all four methods. The third section describes the training network hyperparameters. The fourth one presents final experimental results and the last section analyses the performance of the methods. Prior to any experimental analysis within the FCN networks, we used ArcGIS to identify which bands of Sentinel-2 image are useful in boundaries detection, and we observed that there is no much information for boundaries at 60 m resolution bands (Figure 5.1). Hence, we used FCN to analyze the band combinations from 10 m and 20 m resolution bands.

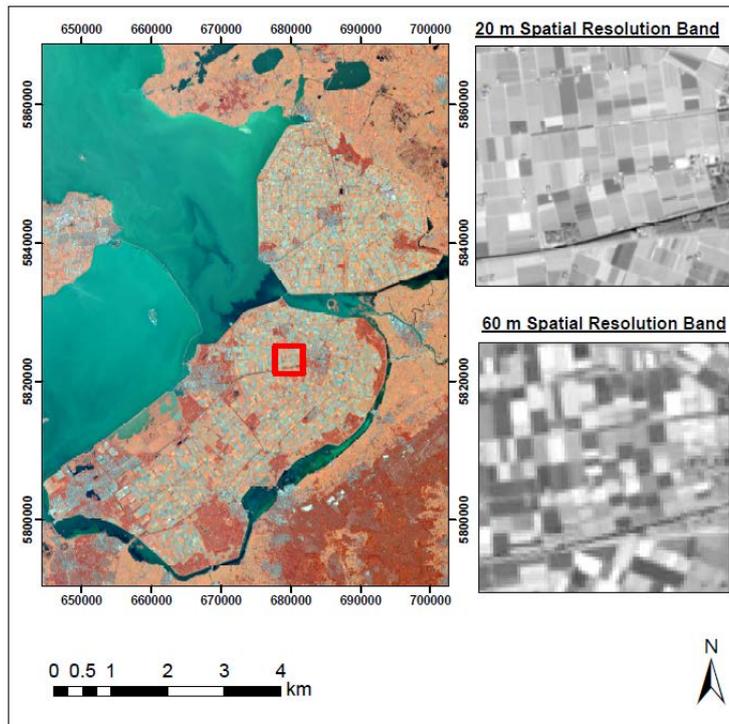


Figure 5.1: The left image is a study area, Flevoland, from the Sentinel-2 with the RGB false colour composite of 843. The red box presents the part of the study area that we analysed using ArcGIS for the clarity of AFB information. The right top and bottom images from the right side represents single band of 20 m and 60 m spatial resolution respectively. The figure illustrates that, there is no much boundaries information at the 60 m resolution band.

5.1. Hyperparameters sensitivity analysis

We started analysis by conducting preliminary experiments using only two tiles. One tile (TR1) for training and another tile (TS1) for testing the network (Figure 5.2). The corresponding ground truths (GT_TR1 and GT_TS1) for TS1 and TS2 respectively are shown in Figure 5.3. We performed the preliminary experiments to tune the hyperparameters to be used for the full dataset, the 5 tiles (TR1, TR2, TR3, TR4, TR5) for training the network and the 5 tiles (TS1, TS2, TS3, TS4, TS5) for testing the network. We assessed the performance of the networks specifically on detecting the boundaries using F-Score of the AFB output maps by comparing pixel by pixel, the actual and predicted value of the AFB class label. We started the experiments using network architecture that we adapted from (Persello & Stein, 2017). We used architecture that consists five convolutional layers with 5×5 filters and patch size of 95 hyperparameters purposively as a baseline to detect AFB from Sentinel-2 satellite image. The following parts describe sensitivity hyperparameters namely; training sample, max-pooling, band combination, filter size and patch size in a sequential process.

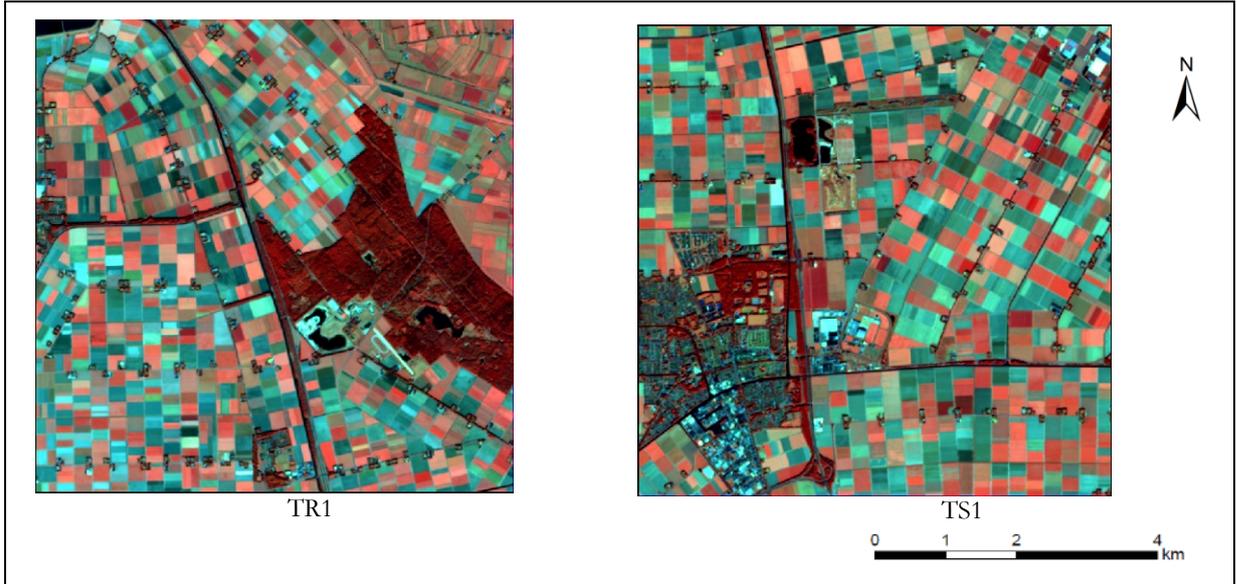


Figure 5.2: TR1 (training) and TS1 (testing) are the tiles of 800×800 pixels cropped from Sentinel-2 image. These tiles were used for all preliminary experiments.

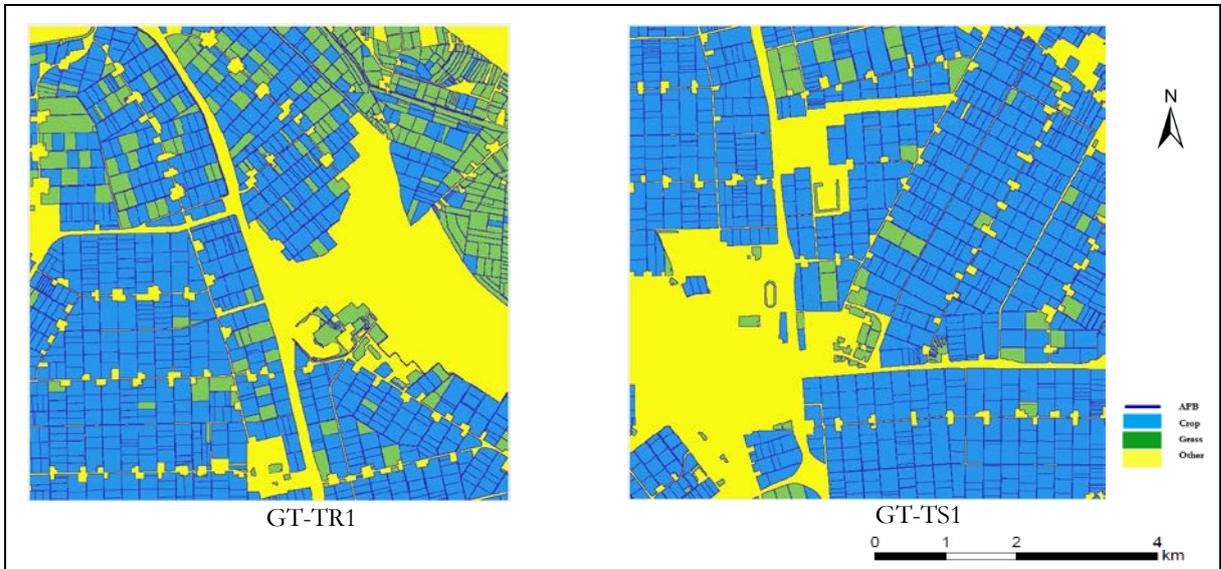


Figure 5.3: GT-TR1 and GT-TS1 are fully labeled references corresponding to TR1 and TS1 tiles respectively

5.1.1. Training samples

We investigated the training samples sizes of 500, 1000 and 2000 from FCN-DK5 and fixing other hyperparameters as shown in Table 5.1. The training samples were taken randomly from fully labeled patches for training the network. In this analysis, we used input image of only 4-band combination (Visible and NIR) at 10 m resolution from Sentinel-2.

Table 5.1: Hyperparameters for training sample analysis

Hyperparameter	Value
Depth of the network	5 convolution layers
Filter size	5×5
Patch size	95
Training sample	500, 1000, 2000

The network provided AFB maps with F-Score accuracy of 0.4%, 6%, and 16% for 500, 1000 and 2000 training sample sizes respectively for TS1 of the class AFB. In this analysis, we observe that, by increasing the number of training samples, the boundaries are better extracted even if many are still missing (Figure 5.4), this is because of the max-pooling which smoothen them. Therefore, we performed the next experiment without incorporate max-pooling layers and training the network using 2000 training sample size.

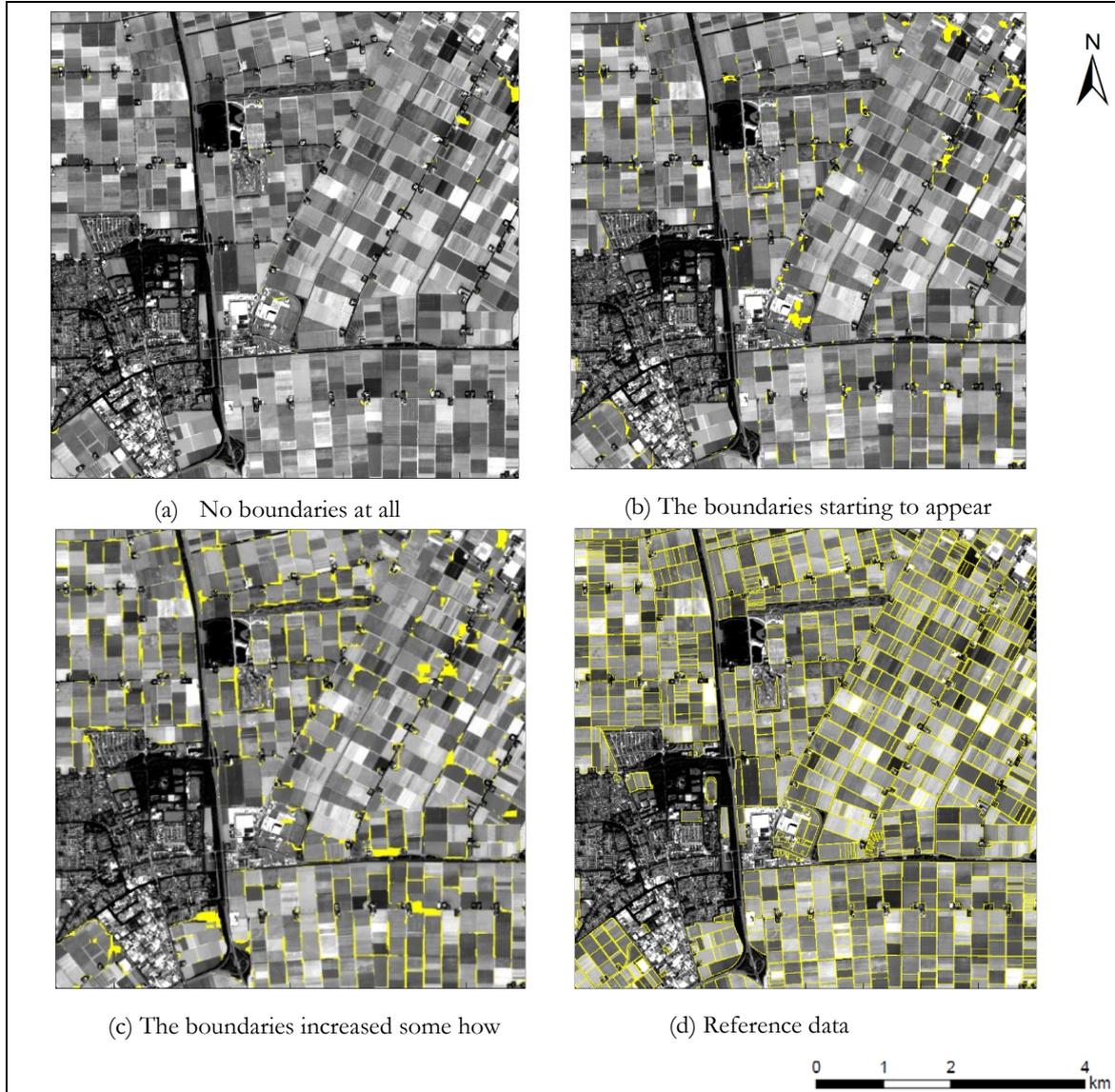


Figure 5.4: The AFB maps of TS1 from training sample size analysis: a, b and c are the output maps from the FCN-Net5 using training sample sizes of 500, 1000, and 2000 respectively. Yellow color represents the AFB.

5.1.2. Max-pooling layer analysis

We investigated the FCN-DK5 by removing max-pooling layer after every non-linearity function. We applied the hyperparameters in the Table 5.2. In this analysis, we renamed the FCN-DK5 to FCN-Net of $N = 5$ and its architecture is illustrated in a Table 5.4. Based on the experimental results (Figure 5.4 and Figure 5.5), the network without max-pooling layers performed better. This is because computing a max-pooling on a large window will basically destroy the exact location especially for the small and thin features like boundaries. Therefore, the AFB completely disappeared and this accounts for the lower

accuracy on it. Hence, we used the network architecture without max-pooling layers in the rest of the experiments.

Table 5.2: Hyperparameters for pooling layer analysis

Hyperparameter	Value
Depth of the network	5 convolution layers
Filter size	5×5
Patch size	95
Training sample	2000



Figure 5.5: AFB reference (a) and output map from FCN-Net of $N=5$ architecture without incorporating max-pooling layers (b) for the TS1: The map is an output with an F-Score accuracy of 0.41. Yellow colour represents AFB.

5.1.3. Bands combination analysis

We conducted five experiments using FCN-Net of $N = 5$ to investigate bands combination (Table 5.3) from Sentinel-2 to get the relevant one for AFB detection. In this analysis, we used hyperparameters in Table 5.2. The bands of 20 m resolution were resampled to 10 m resolution using bilinear technique in ArcGIS. We applied this resampled technique because it is best for resampling continuous data with multiresolution such as Sentinel-2 as it uses weighted average. In this analysis we also analysed the effect of SWIR bands (band 11 and band 12) on the accuracy of AFB.

Table 5.3: Band combination

Name of band combination	Band
4-band combination	2, 3, 4 and 8
8-band combination	2, 3, 4, 5, 6, 7 and 8a
8-mix-band combination	2, 3, 4, 6, 8, 8a, 11 and 12
10-band combination	2, 3, 4, 5, 6, 7, 8, 8a, 11 and 12

Table 5.5 presents the results of only TS1 which show that the addition of vegetation bands to the 4-band combination (8-band combination which was highlighted) improved the F-Score accuracy by 1%, 2%, and 19% to the 4, 10 and 8-mix-band combinations respectively. This increment is attributed to the incorporation of some information of the agricultural boundaries from the vegetation bands. However, the additional of SWIR bands to the 8 and 10-bands combination degraded the accuracy of AFB because SWIR is very good in discriminating soil from vegetation but where there are boundaries between vegetation classes, it cannot easily discriminate, therefore this lowered the accuracy. Hence, the remaining experiments conducted using 8-bands combination highlighted in Table 5.3.

Table 5.4: The FCN-Net of $N = 5$ architecture

Network Block (N)	Layer Name	Weight	Dilation	Stride	Pad
1	convLayer1	$5 \times 5 \times 4 \times 16$	1	1	2
	1ReLU				
2	convLayer2	$5 \times 5 \times 16 \times 32$	2	1	4
	1ReLU				
3	convLayer3	$5 \times 5 \times 32 \times 32$	3	1	6
	1ReLU				
4	convLayer4	$5 \times 5 \times 32 \times 32$	4	1	8
	1ReLU				
5	convLayer5	$5 \times 5 \times 32 \times 32$	5	1	10
	1ReLU				
Classification	convLayer	$1 \times 1 \times 32 \times 4$	1	1	0
	softmax				

Table 5.5: Results of the class AFB for the experiments on band combinations

Band Combination	Type I Error	Type II Error	Precision	Recall	F-Score
4-bands	0.15	0.42	0.32	0.58	0.41
8-bands	0.12	0.47	0.35	0.53	0.42
8-mix-bands	0.18	0.67	0.18	0.33	0.24
10-bands	0.14	0.46	0.32	0.54	0.40

5.1.4. Filter size experimental analysis

We examined the effect of convolutional filter size using FCN-Net of $N = 5$. In this analysis we used 8-band combination and hyperparameters in

Table 5.6. These filter sizes are arranged in a matrix format defined as $f \times f$, with an odd number of f . The higher the f the more the neighbouring pixels to be looked from the input which increases the computational cost in the network. We conducted three experiments to analyse the filter sizes of 7×7 , 5×5 and 3×3 and we observed that the F-Score decreases when we increase the filter size. Therefore, the filter size of 3×3 highlighted in Table 5.7 performed better than the higher ones. This is because the high the filter size smoothen the thinner features (AFB). Hence, the rest of the experiments were performed with convolutional layers of 3×3 filter size

Table 5.6: Hyperparameters for filter size analysis

Hyperparameter	Value
Depth of the network	5 convolution layers
Filter size	7×7, 5×5, 3×3
Patch size	95
Training sample	2000

Table 5.7: Results of the class AFB for the experiments on filter size

Filter Size	Type I Error	Type II Error	Precision	Recall	F-Score
3×3	0.13	0.42	0.36	0.58	0.44
5×5	0.12	0.47	0.35	0.53	0.42
7×7	0.12	0.50	0.33	0.50	0.40

5.1.5. Patch size experimental analysis

FCN receives either a whole image or a subset of an image as an input. A subset of an image is known as a patch. Ideally, a filter convolves the image by looking each patch at a time to find patterns efficiently. In this study, we investigate the patch size of 115, 95, 75, 55, and 35 from 8-band combination of the Sentinel-2 using the hyperparameters in Table 5.8. These patch sizes were purposely selected to find the best one for detecting the AFB.

Table 5.8: Hyperparameters for patch size analysis

Hyperparameter	Value
Depth of the network	5 convolution layers
Filter size	3×3
Patch size	115, 95, 75, 55, and 35
Training sample	2000

The results of these analyses are shown in Table 5.9, and they show that the highlighted patch size of 55 performed better compared to the others. This is because the smaller the patch size, the higher the effectiveness of convolution such that the filters can extract small and thin features like boundaries efficiently. However, a patch size of 35 although small reduced the accuracy because of the possibility of some boundary features missing during the convolution.

Table 5.9: Results of the class AFB for the experiments on Patch Size

Patch Size	Type I Error	Type II Error	Precision	Recall	F-Score
115	0.12	0.44	0.35	0.56	0.43
95	0.13	0.42	0.36	0.58	0.44
75	0.11	0.46	0.37	0.54	0.44
55	0.11	0.43	0.38	0.57	0.46
35	0.12	0.45	0.35	0.55	0.43

5.1.6. Summary of hyperparameters sensitivity analysis

In summary, the network architecture without pooling layer performed better with the optimal hyperparameters shown in Table 5.10. Also, the agricultural field boundaries detection from the 8 bands (2, 3, 4, 5, 6, 7 and 8a) of Sentinel-2 contain detailed information.

Table 5.10: Optimal hyperparameters

Hyperparameter	Value
Filter size	3×3
Patch size	55
Training sample	2000

5.2. Architectures experiments

This section presents all experiments that describes how we obtained the best architecture from every method namely; Deep_FCNet, MS-FuseNet, SRM-Net and SRM+LRM-Net. We performed the experiments by applying the optimal hyperparameters explained in the previous section and tuning other parameters including the depth of the network for every method.

5.2.1. Deep_FCNet architectures experiments.

We performed twelve experiments (exp1-exp12) using Deep_FCNet general architecture as explained in Deep FCN-DKs Networks (Deep_FCNet-Net). We applied 8-band combination and optimal hyperparameters in (Table 5.10) but we varied the network architecture. The general network architecture contains the number of main network blocks (N) and sub-blocks (M). In this analysis, the first experiment (exp1) doubled the filter of 3×3 for each dilation such that there are 3 network blocks $N = 3$ and each block contained two sub-blocks $M = 2$. Thus, this network architecture contains six layers in every block of the network, which are two convolution, two ReLU and two batch normalization layers with a total of 6 convolutional layers (Table 5.11). The second, third and fourth experiments are the extensions of the first experiment. In these experiments, the only sub-blocks of the networks increased to 3, 4 and 5 for the third, fourth and fifth experiment respectively. The experiment two (exp2) has 9 layers for every network block. Each network block has three convolutional layers, three 1ReLU, and three batch normalization. There are 9 convolutional layers in this network architecture which is more in-depth, the more the convolutional layers, the deeper the network. The series of experiments followed the pattern of the first 4 experiments. However, they varied the structure by increasing the number of the main network blocks. The next four experiments implemented with the main network block of $N = 4$ and the last block had $N = 5$.

Table 5.12 summarizes the network architectures for all 12 experiments using the general architecture of the Deep_FCNet-Net. This general architecture contains the number of network blocks (N) that represents dilation factor for each convolutional layer contained in that block as shown in Figure 3.1. For example, network block $N = 3$ is a network block that contains dilation factor $d = 3$. Then, filters for all convolution layers within this block were dilated by a factor of 3. Moreover, each block contained sub-blocks (M) that consists of convolution layers with filter size 3×3 , batch normalization (BN) and non-linear activation function (ReLU). Table 5.13 shows the results of only TS1 from 12 experiments. The results show the deeper the network, the higher the accuracy. However, the more we increase the number of sub-blocks the accuracy decreases as shown in experiments exp4, exp8, and exp12. This observation depicts that the dilation factor within the main-blocks is limited to the four convolutional layers. The accuracy decreases with the

fifth convolutional layer within the main block. In these experiments, the exp11 performed best and hence, we applied the architecture of these experiments to the full dataset.

Table 5.11: Network architecture of exp1 using FCN-Net of $N = 3$ and $M = 2$

Network Block (N)	Layer Name	Weight	Dilation	Stride	Pad
1	convLayer1	$3 \times 3 \times 8 \times 16$	1	1	1
	1ReLU				
	convLayer2	$3 \times 3 \times 16 \times 32$	1	1	1
	1ReLU				
2	convLayer3	$3 \times 3 \times 32 \times 32$	2	1	2
	1ReLU				
	convLayer4	$3 \times 3 \times 32 \times 32$	2	1	2
	1ReLU				
3	convLayer5	$3 \times 3 \times 32 \times 32$	3	1	3
	1ReLU				
	convLayer6	$3 \times 3 \times 32 \times 32$	3	1	3
	1ReLU				
	convLayer	$1 \times 1 \times 32 \times 4$	1	1	0
	softmax				

The batch normalization was performed after every convolutional layer

Table 5.12: Deep_FCNet architectures

Experiment	N	M	NC
exp1	3	2	6
exp2	3	3	9
exp3	3	4	12
exp4	3	5	15
exp5	4	2	8
exp6	4	3	12
exp7	4	4	16
exp8	4	5	20
exp9	5	2	10
exp10	5	3	15
exp11	5	4	20
exp12	5	5	25

These are preliminary experimental architectures for the Deep_FCNet. N and M represent the number of main and sub-network blocks respectively. NC is the total number of convolutional layers within the network. The highlighted shows the best architecture of the Deep_FCNet applied for the full dataset.

Table 5.13: The F-Score results of AFB class for Deep_FCNet architectures

experiment	Type I error	Type II error	Precision	Recall	F-Score
exp1	0.11	0.41	0.40	0.95	0.48
exp2	0.11	0.40	0.41	0.60	0.49
exp3	0.10	0.38	0.43	0.61	0.51
exp4	0.12	0.48	0.17	0.52	0.25
exp5	0.11	0.41	0.39	0.59	0.46
exp6	0.10	0.38	0.42	0.62	0.50
exp7	0.11	0.39	0.40	0.61	0.49
exp8	0.12	0.67	0.12	0.33	0.17
exp9	0.12	0.49	0.16	0.51	0.25
exp10	0.11	0.41	0.41	0.60	0.48
exp11	0.10	0.39	0.44	0.61	0.51
exp12	0.11	0.59	0.15	0.41	0.22

These are the results of the 12 experimental architecture of the Deep_FCNet. The highlighted is the results of the best Deep_FCNet architecture.

5.2.2. MS-FuseNet architectures experiments.

We performed several experiments using MS-FuseNet (Fusion of multispectral bands from Sentinel-2 within the network). We conducted the experiments using the optimal hyperparameters (Table 5.10) In these experiments we varied the depth of the networks. Also, we applied the max-pooling layer to down-sampling the image from 10 m to 20 m for the second input stream such that to have the same spatial dimensional features maps. Table 5.14 and Table 5.15 show nine experiments using MS-FuseNet and their F-Score results respectively. We observed that MS-FuseNet architectural design of acyclic causes the fluctuation in F-Score accuracy. Hence, it is not easy to interpret the pattern of the accuracy results

Table 5.14: MS-FuseNet architectures

Experiment	N	M	NC
exp1	3	3	11
exp2	3	4	14
exp3	3	5	17
exp4	4	3	14
exp5	4	4	18
exp6	4	5	22
exp7	5	3	17
exp8	5	4	22
exp9	5	5	27

These are preliminary experimental architectures for the MS-FuseNet. N and M are the number of main sub-network blocks respectively. The number of convolution layers (NC) calculated by multiplying N and M; and adding 2 convolution layers from both input streams. The highlighted shows the best architecture of the MS-FuseNet applied for the full dataset.

Table 5.15: The F-Score results of AFB class for MS-FuseNet architectures

experiment	Type I error	Type II error	Precision	Recall	F-Score
exp1	0.16	0.40	0.31	0.60	0.41
exp2	0.17	0.41	0.30	0.59	0.40
exp3	0.17	0.42	0.30	0.58	0.40
exp4	0.17	0.43	0.29	0.57	0.39
exp5	0.15	0.39	0.33	0.61	0.43
exp6	0.17	0.40	0.31	0.59	0.40
exp7	0.16	0.29	0.32	0.61	0.42
exp8	0.15	0.42	0.32	0.58	0.41
exp9	0.17	0.40	0.31	0.60	0.41

These are the results of the 9 experimental architecture of the MS-FuseNet. The highlighted is the results of the best MS-FuseNet architecture.

Additionally, we used the stride factor of $f = 2$ in a convolutional layer instead of the max-pooling layer for the first input stream. To use this technique, we repeated only one experiment (exp5) because it performed best than the other experiments (Table 5.15). Similarly, we repeated exp5 to investigate max-pooling as a down-sampling and transposed convolution without bilinear function for weight initialization as its corresponding up-sampling. Another experiment with the same network architecture was carried out using the stride factor of $f = 2$ in a convolutional layer as a down-sampling and transposed convolution without bilinear function as its corresponding up-sampling. We observed that, down-sampling using max-pooling performed marginal better than down-sampling using a stride of a factor 2 by slightly increasing the F-Score accuracy. This slight increment in the accuracy for AFB attributed to the transposed convolution with the bilinear interpolation for weight initialization.

5.2.3. SRM-Net architectures experiments.

We investigated SRM by conducting several experiments and divided these experiments into two cases. The first case was placing the up-sampling layer at the end of the convolutional series. In this case, we examined the transposed convolution with and without bilinear (random) function for weight initialization. Table 5.16 shows the structure of the experiments for the first case with and without bilinear function for weight initialization respectively. The results of the first case with bilinear for weight initialization in transposed convolution are shown in Table 5.17. However, for the experiments without bilinear, the sub-block network of network block $N = 4$ increased to seven for further investigation, and we stopped at $M = 7$ because the results started to decrease (Table 5.18).

We carried out the experiments for the second case whereby the series of network blocks were operated after the up-sampling layer. In this case, we applied the transposed convolution with randomly weight initialization. Table 5.19 and Table 5.20 shows the structure of the preliminary architectures for the second case and their results respectively. Based on all experiments, the best results were obtained from exp7 (Table 5.18).

Table 5.16: SRM-Net architecture for Case-1

Experiment	N	M	NC
exp1	3	3	9
exp2	3	4	12
exp3	3	5	15
exp4	4	3	12
exp5	4	4	16
exp6	4	5	20
exp7	4	6	24
exp8	4	7	28
exp9	5	3	15
exp10	5	4	20
exp11	5	5	25

All experiments were done using transposed convolution that initialized its weight using both randomly and bilinear function. The later skipped two experiments (exp7 and exp8). Case-1 presents the architecture that placed the transposed convolution at the end of the convolutional layers within the network.

Table 5.17: The F-Score results of AFB class for SRM-Net architectures Case-1 (a)

experiment	Type I error	Type II error	Precision	Recall	F-Score
exp1	0.10	0.63	0.20	0.37	0.26
exp2	0.10	0.61	0.21	0.40	0.28
exp3	0.10	0.63	0.20	0.38	0.26
exp4	0.09	0.64	0.21	0.36	0.27
exp5	0.09	0.64	0.20	0.36	0.25
exp6	0.10	0.63	0.19	0.37	0.25
exp9	0.09	0.61	0.21	0.39	0.27
exp10	0.10	0.60	0.21	0.41	0.28
exp11	0.10	0.69	0.17	0.32	0.22

These are the results corresponding to the SRM-Net architecture explained in Table 5.17. Case-1 (a) presents the transposed convolution layer that was set to initialize its weight using bilinear interpolation method.

Table 5.18: SRM-Net architecture for Case-2

Experiment	N	M	NC	Up-sampling layer position
exp1	4	4	16	After N=3
exp2	5	5	25	After N=4
exp3	6	3	18	After N=3
exp4	6	4	24	After N=4
exp5	6	4	24	After N=3
exp6	6	5	30	After N=3

All experiments were done using transposed convolution that initialized its weight randomly. Case-2 presents that, these architectures added series of network blocks after transposed convolution layer.

Table 5.19: The F-Score results of AFB class for SRM-Net architectures Case-1 (b)

experiment	Accuracy	Type I error	Type II error	Precision	Recall	F-Score
exp1	0.88	0.10	0.56	0.23	0.44	0.30
exp2	0.88	0.09	0.55	0.24	0.45	0.31
exp3	0.87	0.10	0.58	0.22	0.42	0.28
exp4	0.87	0.10	0.56	0.22	0.44	0.30
exp5	0.88	0.10	0.56	0.23	0.44	0.30
exp6	0.88	0.10	0.55	0.23	0.45	0.31
exp7	0.88	0.10	0.54	0.24	0.46	0.32
exp8	0.87	0.11	0.57	0.21	0.43	0.28
exp9	0.87	0.10	0.56	0.22	0.44	0.30
exp10	0.87	0.10	0.55	0.22	0.45	0.30
exp11	0.87	0.10	0.56	0.22	0.44	0.29

These are the results corresponding to the SRM-Net architecture explained in Table 5.17. Case-1 (a) presents the transposed convolution layer that was set to initialize its weight randomly. The highlighted represents the results of the best SRM-Net architecture.

Table 5.20: The F-Score results of AFB class for SRM-Net architectures Case-2

experiment	Type I error	Type II error	Precision	Recall	F-Score
exp1	0.10	0.58	0.21	0.42	0.28
exp2	0.10	0.63	0.20	0.37	0.26
exp3	0.09	0.58	0.22	0.42	0.29
exp4	0.10	0.60	0.21	0.40	0.28
exp5	0.10	0.56	0.22	0.45	0.30
exp6	0.10	0.62	0.19	0.38	0.25

The results for SRM using Transposed convolution that was placed between convolutional layers. The weight initialized randomly for the transposed convolutional layer.

5.2.4. SRM+LRM-Net Architectures Experiments.

We performed label refinement on the best SRM-Net architecture (exp7) from Table 5.18. In this analysis, we added convolutional layers after the prediction of the SRM-Net, and we estimated the second prediction. Table 5.21 shows some experiments implemented for refining the SRM output, and their corresponding results are shown in Table 5.22.

Table 5.21: SRM-LRM-Net architectures

Experiment	N	M	NC
exp1	1	6	6
exp2	2	6	12
exp3	1	7	7
exp4	1	8	8
exp5	1	9	9
exp6	1	10	10
exp7	1	11	11
exp8	1	12	12

These are the preliminary experiments for the SRM+LRM-Net architectures. N and M are the main and sub-networks blocks after the first prediction. NC is the additional number of layers to the SRM label refinement. The highlighted is the best architecture of the SRM+LRM-Net.

Table 5.22: The F-Score results of AFB class for SRM+LRM-Net architectures

experiment	Type I error	Type II error	Precision	Recall	F-Score
exp1	0.10	0.56	0.24	0.44	0.31
exp2	0.10	0.55	0.24	0.45	0.31
exp3	0.10	0.52	0.24	0.49	0.32
exp4	0.09	0.55	0.24	0.45	0.31
exp5	0.09	0.55	0.25	0.45	0.32
exp6	0.09	0.54	0.25	0.46	0.32
exp7	0.09	0.56	0.24	0.44	0.31
exp8	0.09	0.55	0.25	0.45	0.32

These are the results of the 8 experimental architecture of the SRM+LRM-Net. The highlighted is the results of the best SRM+LRM-Net architecture.

5.3. Training Network Hyperparameters

To train the networks in all architecture, we fixed the stochastic gradient descent with a momentum of 0.9 and batch size of 32 and analyzed the learning rate and epoch size by varying their values. Initially, we trained Deep_FCNet-Nets in two stages, the first stage using 150 number of epochs with 10^{-4} learning rate and the second stage using 20 number of epochs with 10^{-5} learning rate using 170 number of epochs with the 10^{-5} learning rate. The objective function did not converge well, so, using the same learning rate, we increased the number of epochs to 170 and 30 to the first and second stages respectively.

For FuseNet architecture, we trained the network using 150, 170 and 200 epochs with the learning rate of 10^{-5} . Also, we trained the network in two stages as in a Deep_FCNet-Net. Additionally, we examined 170 and 200 number of epochs with the learning rate using log space between 10^{-4} and 10^{-5} . Since 170 number of epochs with the learning rate of 10^{-5} performed better, these training hyperparameters were used for all FuseNet experiments.

Also, we investigated training network hyperparameters (the number of epochs and learning rate) in the SRM-Net for first preliminary experiments. Initially, we trained the networks using 170 epochs with the learning rate of 10^{-4} , and an objective curve did not converge well, so, we added 30 number of epochs with the learning rate 10^{-5} as a second phase. Lastly, we trained the network using 200 number of epochs with the learning rate of log space between 10^{-4} and 10^{-5} . The second and third cases converge the objective function in a very suitable manner compared to the first one. However, the second investigation shows better performance than the last and hence, we used these training parameters values to train all the experiments of SRM-Net and SRM+LRM-Net.

5.4. Final Experimental Results.

We applied all four methods to the full dataset using 4 classes (AFB, crop, grass and other) and binary classes (AFB and non-boundary). All four methods with binary classes outperformed by increasing the F-Score of the AFB class to approximately 10%. This performance increment can be attributed to the merging of similar classes (crop, grass and other) which may have decreased the errors during the classification. This is because, significant number of training samples that were misclassified from the 4 classes are reduced. This

section presents the results of the final experiments from all four methods with binary classes. See Appendix 1 for the F-Score accuracy with the 4 classes. The results are F-Score of accuracy assessment of only class boundary (AFB) for all tested tiles (TS1, TS2, TS3, TS4 and TS5).

5.4.1. Results of AFB at 10 m resolution

This section presents the results at 10 m resolution. Table 5.23, Table 5.24, Table 5.25 shows the F-Score accuracy of AFB from Deep_FCNDK, MS-FuseNet and FCN-DK respectively and Figure 5.6 show the output maps of AFB at 10 m resolution with their corresponding reference from Deep_FCNDK.

Table 5.23: F-Score accuracy of the experiment from FCN-DK5 with binary classes

Tile	Type I error	Type II error	Precision	Recall	F-Score
TS1	0.04	0.44	0.62	0.56	0.59
TS2	0.06	0.55	0.58	0.45	0.51
TS3	0.05	0.49	0.62	0.51	0.56
TS4	0.06	0.39	0.64	0.61	0.62
TS5	0.06	0.47	0.62	0.53	0.57

These results of the five tested tiles from the FCN-DK-5 architecture adapted from (Persello & Stein, 2017) for the boundary detection. The architecture used 125 patch size and 5×5 filters but we removed max-pooling layers.

Table 5.24: Results of the final experiment using Deep_FCNDK

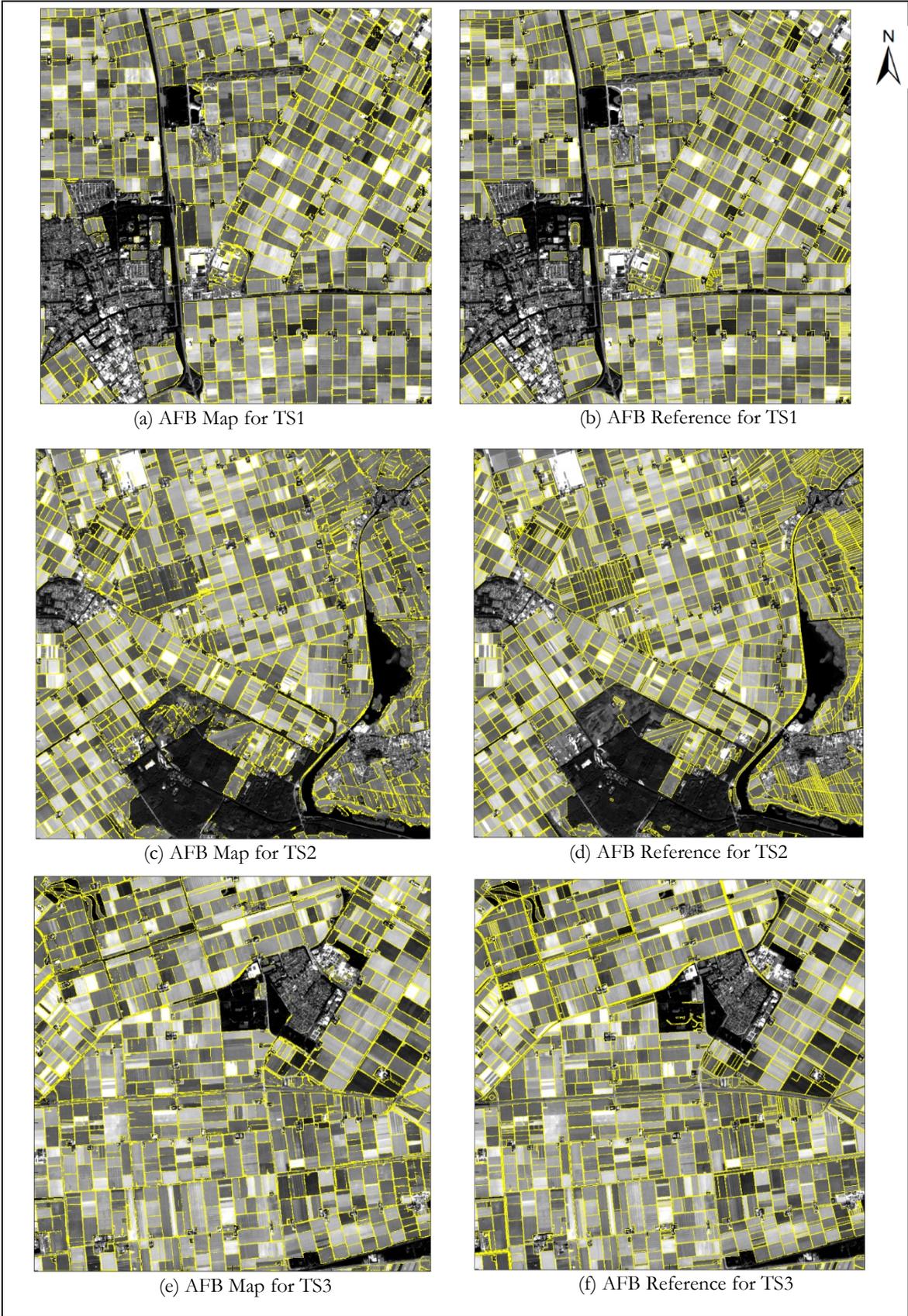
Tile	Type I error	Type II error	Precision	Recall	F-Score
TS1	0.04	0.36	0.66	0.64	0.65
TS2	0.05	0.44	0.64	0.56	0.60
TS3	0.05	0.40	0.66	0.60	0.63
TS4	0.05	0.34	0.69	0.66	0.67
TS5	0.05	0.42	0.67	0.58	0.62

The results obtained from the Deep_FCNDK architecture that contains 5 main blocks of network $N = 5$ and 4 sub-blocks $M = 4$, with a total of 20 convolutional layers. The filter size for each convolutional layer is 3×3 . A patch size of 55 from 8-bands of 10 m resolution and the 5000 training samples, 1000 from each training tile.

Table 5.25: Results of final experiment using MS-FuseNet

Tile	Type I error	Type II error	Precision	Recall	F-Score
TS1	0.05	0.35	0.63	0.65	0.64
TS2	0.05	0.47	0.62	0.53	0.57
TS3	0.05	0.42	0.64	0.58	0.61
TS4	0.05	0.37	0.68	0.63	0.66
TS5	0.05	0.43	0.65	0.57	0.61

The results obtained from the MS-FuseNet architecture that contains one network block ($N = 1$) and one sub-block ($M = 1$) for each input stream, and $N = 4$ with $N = 4$ after concatenation, with a total of 18 convolutional layers. The filter size for each convolutional layer is 3×3 . A patch size of 28 and 56 from 4-bands of 10 m and 20 m resolution and the 5000 training samples, 1000 from each training tile.



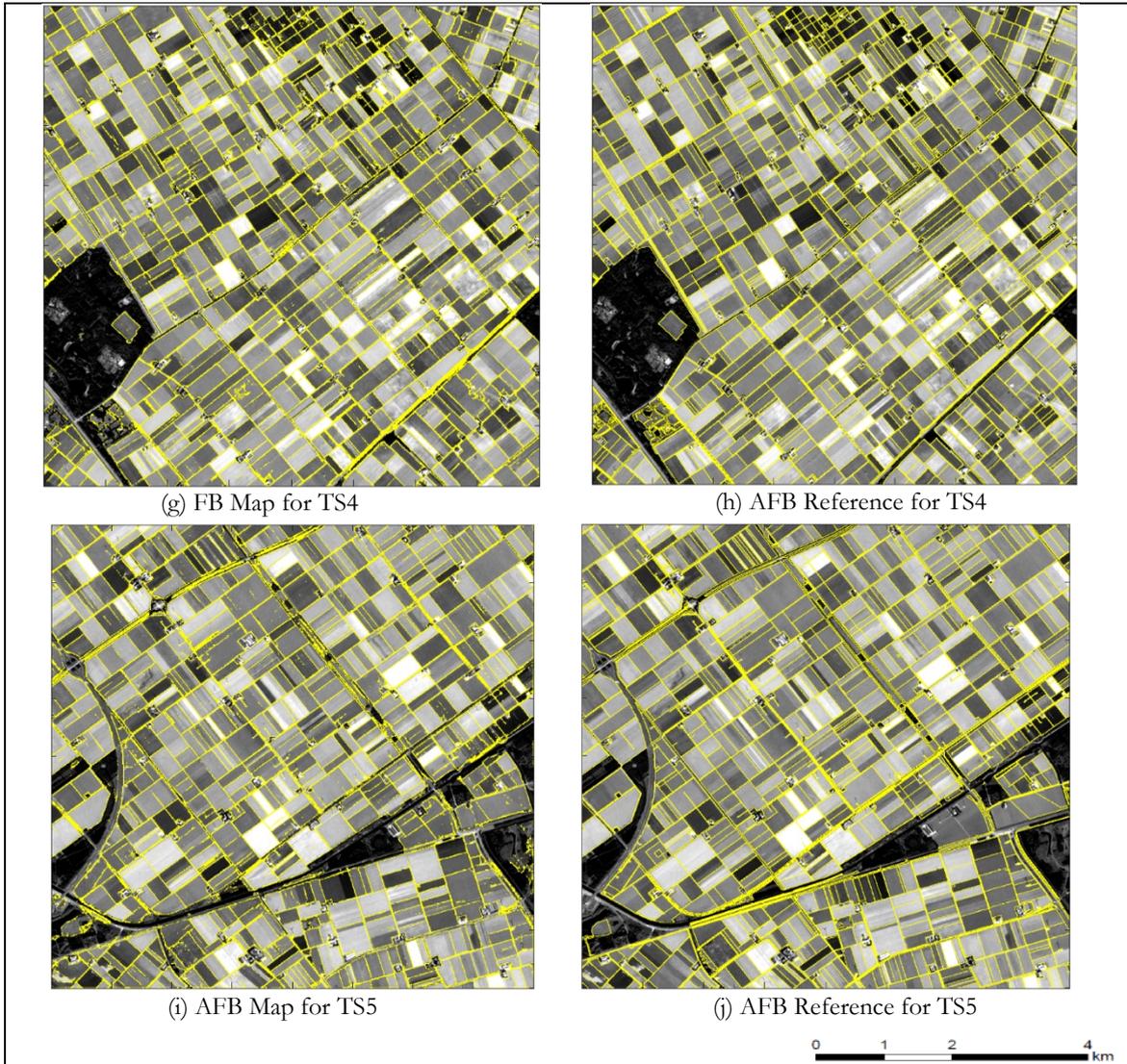


Figure 5.6: Final AFB Maps at 10 m resolution from 10 m resolution of Sentinel-2 image with their corresponding reference of 10 m resolution: The boundaries are represented by yellow color.

5.4.2. Results of AFB at 5 m resolution

This section presents the results at 5 m resolution. Table 5.26, Table 5.27, Table 5.28 shows the F-Score accuracy of AFB from Deep_FCNDK from RapidEye, SRM-Net and SRM+LRM-Net respectively and Figure 5.7 show the output maps of AFB from SRM+LRM-Net with their corresponding reference.

Table 5.26: Results of the final experiment using Deep_FCNDK from RapidEye

Tile	Type I error	Type II error	Precision	Recall	F-Score
TS1	0.03	0.53	0.52	0.47	0.50
TS2	0.04	0.58	0.50	0.42	0.45
TS3	0.03	0.58	0.53	0.42	0.47
TS4	0.04	0.49	0.54	0.51	0.53
TS5	0.03	0.57	0.57	0.43	0.49

The results obtained from the Deep_FCNDK architecture using RapidEye that contains 5 main blocks of network $N = 5$ and 4 sub-blocks $M = 4$, with a total of 20 convolutional layers. The filter size for each convolutional layer is 3×3 . A patch size of 55 from 8-bands of 10 m resolution and the 5000 training samples, 1000 from each training tile.

Table 5.27: Results of the final experiment using SRM-Net

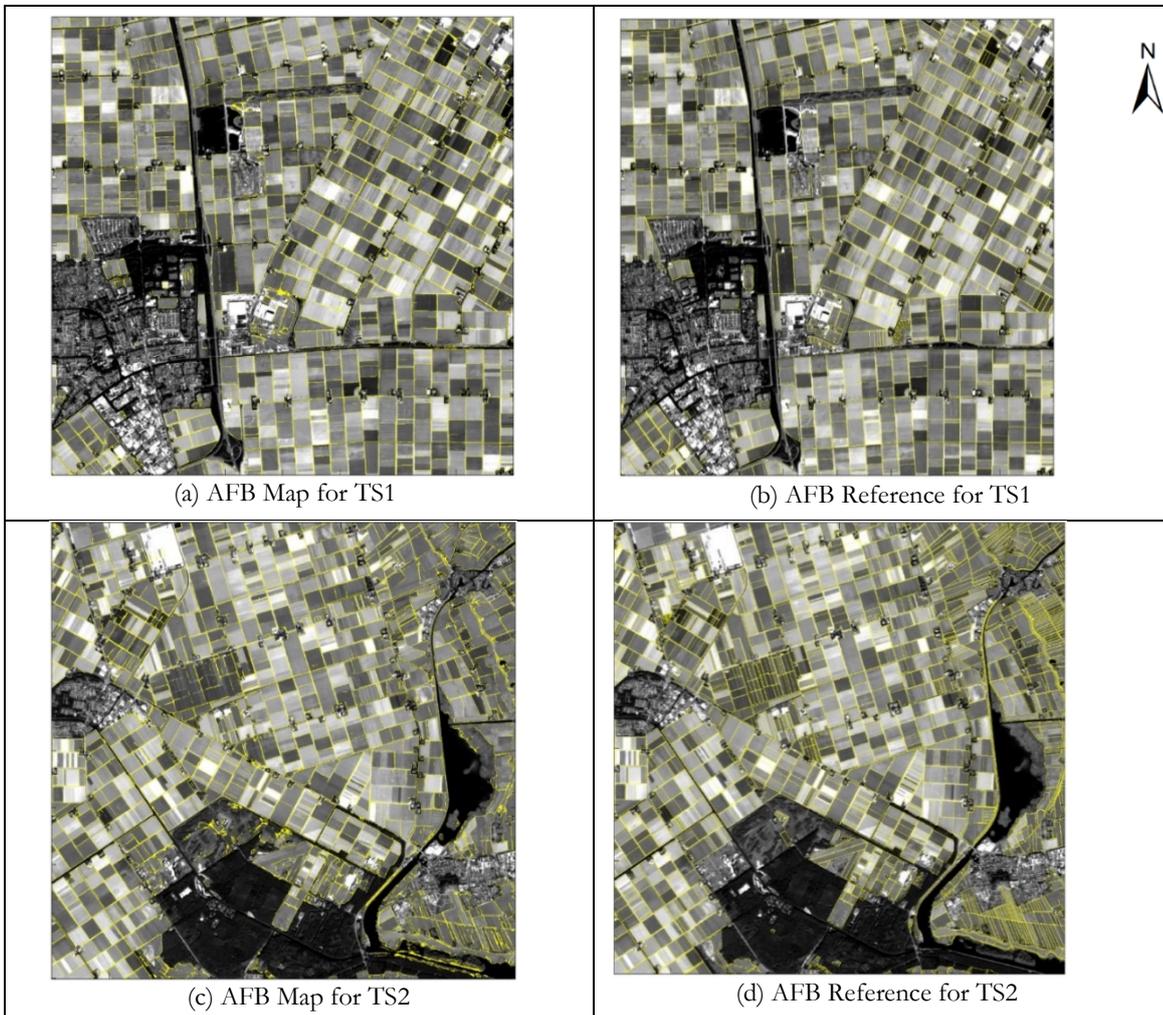
Tile	Type I error	Type II error	Precision	Recall	F-Score
TS1	0.04	0.61	0.41	0.39	0.40
TS2	0.04	0.63	0.44	0.37	0.40
TS3	0.04	0.65	0.43	0.35	0.39
TS4	0.05	0.58	0.45	0.42	0.43
TS5	0.04	0.63	0.46	0.37	0.41

The results obtained from the SRM-Net architecture that contains 24 convolutional layers.

Table 5.28: Results of the final experiment using SRM+LRM-Net

Tile	Type I error	Type II error	Precision	Recall	F-Score
TS1	0.03	0.61	0.43	0.39	0.41
TS2	0.04	0.63	0.46	0.37	0.41
TS3	0.04	0.64	0.45	0.36	0.40
TS4	0.04	0.59	0.46	0.41	0.43
TS5	0.04	0.64	0.47	0.36	0.41

The results obtained from the SRM+LRM-Net architecture that contains 24 and convolutional layers with additional of 10 convolutional layers applied after first prediction



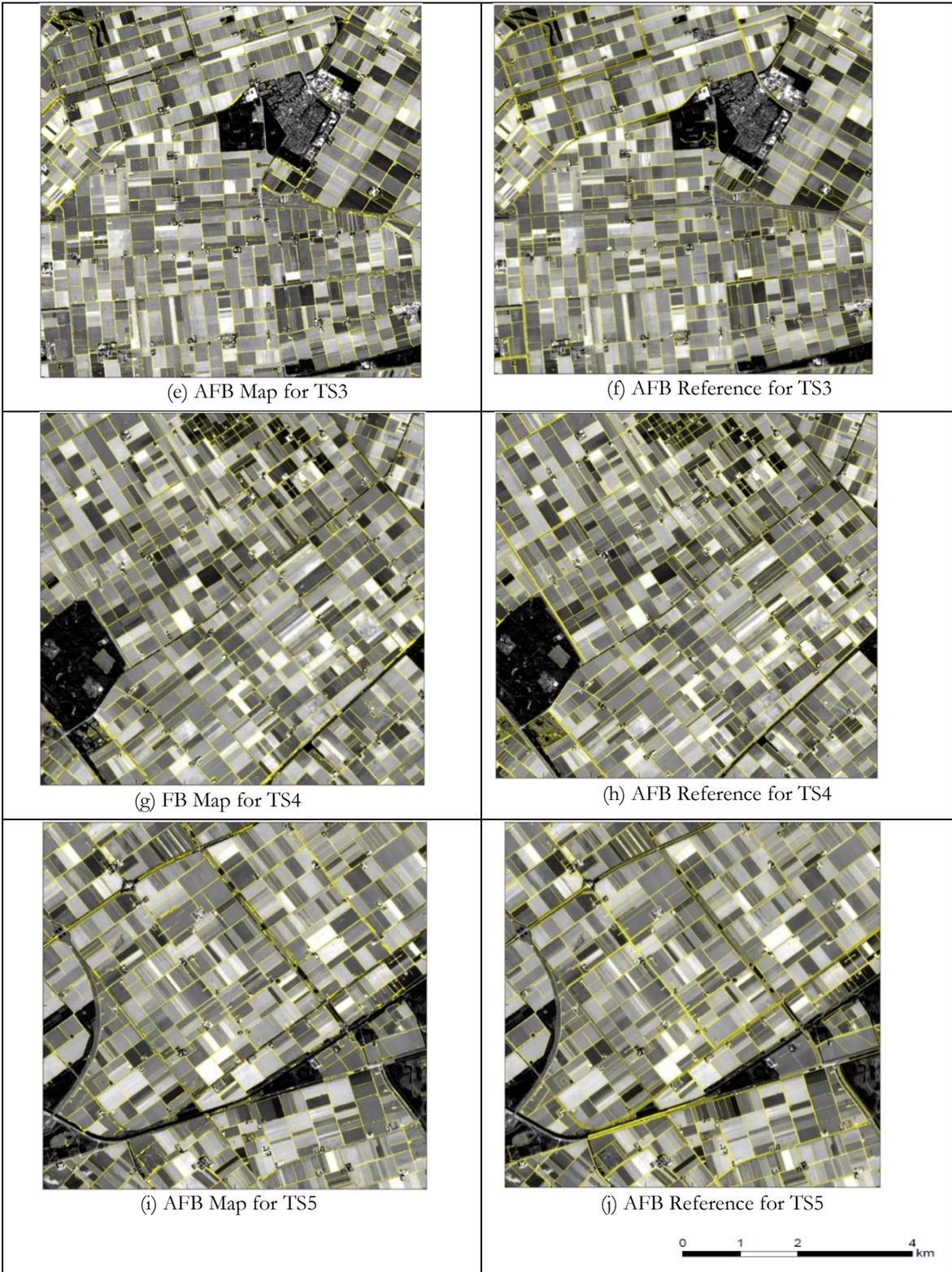


Figure 5.7: Final AFB Maps at 5 m resolution from 10 m resolution of Sentinel-2 image with their corresponding reference of 5 m resolution: The boundaries are represented by yellow color.

5.5. Performance Analysis of the methods.

This section presents the comparison analysis of the methods applied in this study based on the output. We analysed these methods both visually and using the F-Score accuracy measure as described in Accuracy Assessment. We assessed all methods only for the class boundary from the binary classes (boundary and non-boundary).

5.5.1. Methods comparison at 10 m resolution

Deep_FCNet versus FCN-DK5

Based on the results from the analysis, we compared the performance of Deep_FCNet with FCN-DK5 inspired by Persello & Stein, (2017), and in general, we observed that, Deep_FCNet performed better than FCN-DK5 by an increment of 0.064 F-Score of AFB class (Table 5.29).

Table 5.29: F-Score accuracy comparison for Deep_FCNet and FCN-DK5

Method/Tile-F-Score	TS1	TS2	TS3	TS4	TS5
Deep_FCNet	0.65	0.60	0.63	0.67	0.62
FCN-DK5	0.59	0.51	0.56	0.62	0.57

The F-Score results of 10 m spatial resolution for the AFB class show that the Deep_FCNet performed better than the FCN-DK5.

Deep_FCNet versus MS-FuseNet

Table 5.30 presents the F-Score of the AFB class using all tested tiles for the 10 m spatial resolution output from the Sentinel-2 image of the Flevoland. The F-Score results indicate that, the Deep_FCNet performed better than MS-FuseNet by slight difference of 0.016 which is equal to 1.2%. This slight increment on the Deep_FCNet can be attributed to the deepness of the network, such that it has larger number of convolutional layers which are 20 in total compared to the MS-FuseNet which are 18 in total. The difference of the depth of the network is based on the architectural design of these two methods, whereby, the Deep_FCNet used simple chain of blocks order, on the other hand, the MS-FuseNet used DagNN (directed acyclic graph neural network) where its layers are connected to the variables and vice-versa in topological order. Based on the architectural design of MS-FuseNet, it is not easy to interpret the pattern of the F-Score accuracy results from network depth analysis. Besides, the Deep_FCNet produced the maps with less noise compared to the MS-FuseNet as shown in Figure 5.8.

Table 5.30: The F-Score accuracy comparison for Deep_FCNet and MS-FuseNet methods

Method/Tile-F-Score	TS1	TS2	TS3	TS4	TS5
Deep_FCNet	0.65	0.60	0.63	0.67	0.62
MS-FuseNet	0.64	0.57	0.61	0.66	0.61

The F-Score results of 10 m spatial resolution for the AFB class show that the Deep_FCNet performed better than the MS-FuseNet.

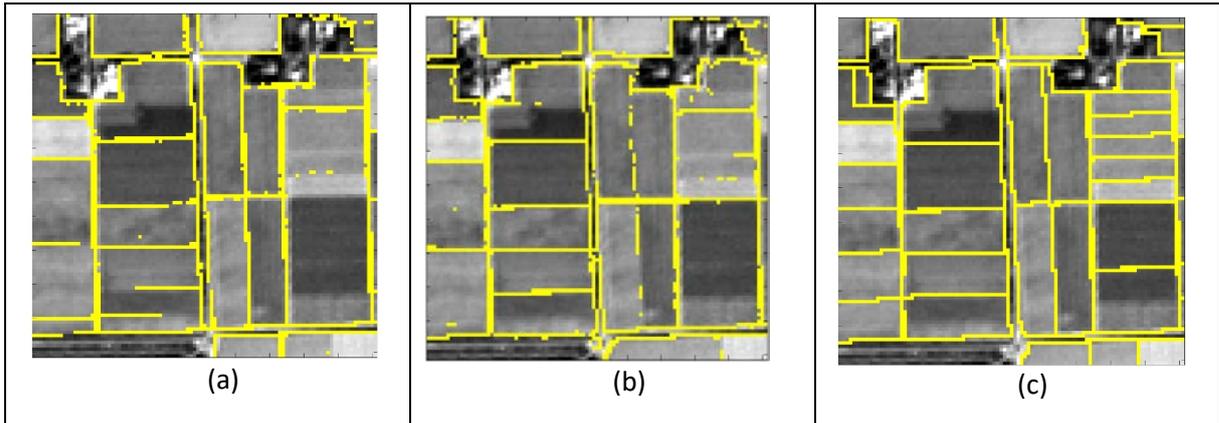


Figure 5.8: AFB maps of 10 m spatial resolution from Sentinel-2 image using a Deep_FCNet (a) and MS-FuseNet (b) with their reference map of 10 m resolution (c).

5.5.2. Methods comparison at 5 m resolution

SRM-Net versus SRM+LRM-Net

Table 5.31 presents the F-Score accuracy for the AFB output of 5 m spatial resolution using SRM-Net which the enhancement network of the AFB maps from 10 m to 5 m resolution and the SRM+LRM-Net which the refine version of the SRM-Net. The SRM+LRM-Net does not affect significantly the accuracy of AFB from SRM-Net. This is because the F-Score of SRM+LRM-Net increased slightly by 0.006 which is equal to 6%. However, the maps visually depict that the SRM+LRM-Net improved the AFB by filtered the noise and increased the capability to discriminate different field as shown in Figure 5.9

Table 5.31: The F-Score accuracy comparison for SRM-Net and SRM+LRM-Net

Method/Tile-F-Score	TS1	TS2	TS3	TS4	TS5
SRM-Net	0.40	0.40	0.39	0.43	0.41
SRM+LRM-Net	0.41	0.41	0.40	0.43	0.41

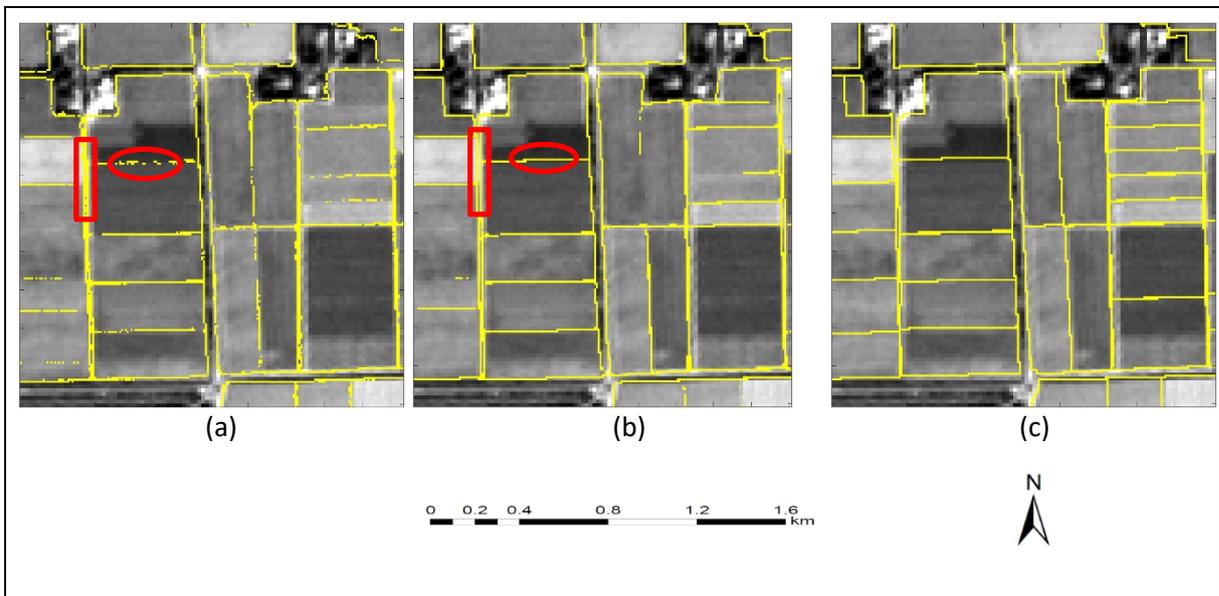


Figure 5.9: AFB maps of 5 m spatial resolution from Sentinel-2 image using SRM-Net (a) and SRM+LRM-Net (b) with their reference map of 5 m resolution (c).

SRM+LRM-Net versus Deep_FCN-Net from RapidEye

Table 5.32 presents the F-Score accuracy for the AFB output of 5 m spatial resolution using Deep_FCN-Net from RapidEye as a baseline, and SRM+LRM-Net. The Deep_FCN-Net using RapidEye performed better than SRM+LRM-Net by 0.076 which is equal to 7.6%. This is because Deep_FCN-Net was applied directly to the 5 m spatial resolution of the RapidEye data without up-sampling therefore the effects such as loss of fine details and visual artefacts associated with up-sampling was limited.

Table 5.32: The F-Score accuracy comparison for Deep_FCN-Net form RapidEye and SRM+LRM-Net

Method/Tile-F-Score	TS1	TS2	TS3	TS4	TS5
Deep_FCN-Net for RapidEye	0.50	0.45	0.47	0.53	0.49
SRM+LRM-Net	0.41	0.41	0.40	0.43	0.41

SRM+LRM-Net versus nearest neighbour interpolation-based resample

Lastly, based on the results from the analysis, we also compared the performance of SRM+LRM-Net by performing nearest neighbor-based resample on Deep_FCN-Net output at 10 m resolution. Generally, SRM+LRM-Net produced the boundaries maps with the precise location compared to the nearest neighbor-based resample that we applied to output of Deep_FCN-Net at 10 m resolution (Figure 5.10).

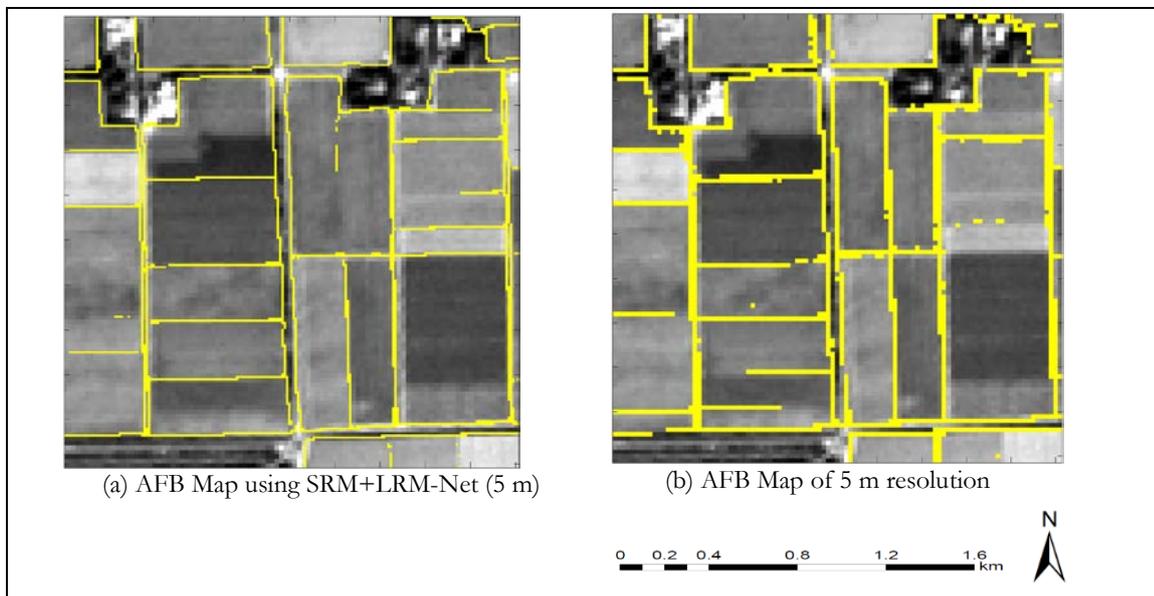


Figure 5.10: AFB output maps from SRM+LRM-Net (a) and output map from nearest neighbour-based resample of output of Deep_FCN-Net at 10 m resolution (b).

5.5.3. Output comparison at 10 m and 5 m resolution

Deep_FCN-Net versus SRM+LRM-Net

In Figure 5.11, it shows that, the AFB at 10 m resolution from Deep_FCN-Net cannot separate two parallel boundaries with the distance of 20 m, on the contrary, the AFB map at 5 m resolution from the SRM+LRM-Net can separate them. Furthermore, in the output of 10 m resolution, we see that, identification of exact location of boundaries is a challenge because some boundaries are located within the agricultural parcel while the boundaries in 5 m resolution output located precisely in somehow.

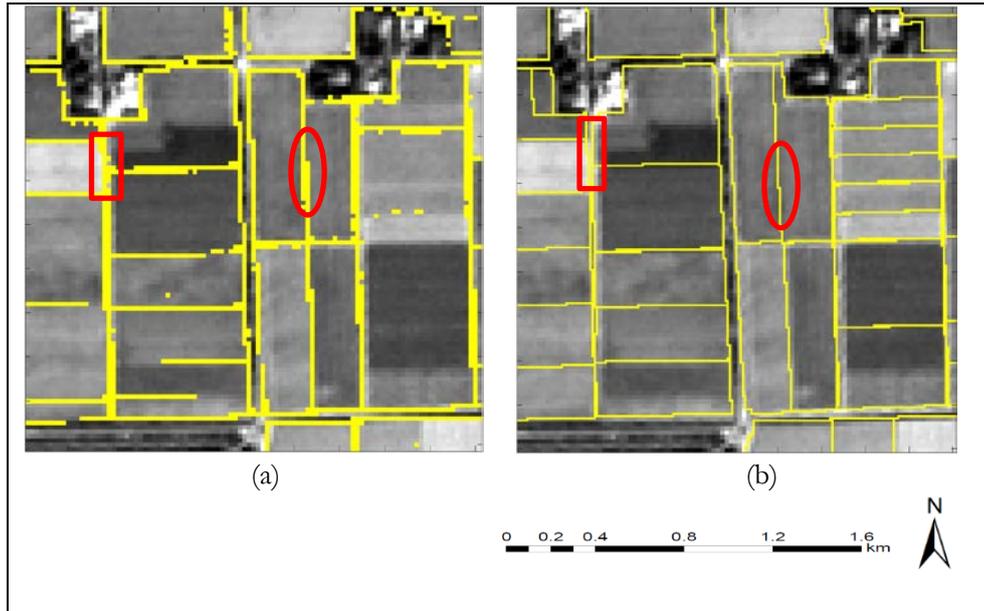


Figure 5.11: The AFB output maps at 10 m and 5 m resolution from Deep_FCNet-Net (a) and SRM+LRM-Net respectively. Oval shape in (b) illustrates precise location compared to oval shape in (a). Rectangle in (b) illustrates better separation of the boundaries with the small distance of approximately 20 m than in (a).

6. DISCUSSION

In this chapter, we discuss the results from all novel methods we developed. We divide discussion into four sections. We first describe the applicability of the methods followed by accuracy assessment strategy. Then we describe the limitations of the methods and finally we present the final remarks on the developed methods with their respective results.

Identification of the boundaries is a difficult problem, especially when the boundaries are detected from medium spatial resolution data such as Sentinel-2 images. Based on the definition of boundaries (AFB) that was given in this study, AFB is not necessarily associated to the object. We infer the AFB as the transition from one agricultural field to another, or from agricultural field to non-agricultural field. While roads and ditches can be possible boundaries present in the image, in this study we classified them as the non-agricultural land cover types rather than the boundaries themselves. Taking the above criteria in consideration, we therefore mapped the boundaries as the outer extent of agricultural fields that do not have specific size.

6.1. Applicability of the methods

In reference to the accuracy assessment presented in section 5.4, the methods are considered reliable based on the complexity of the problem, beside this accuracy assessment is also tolerating any small shift. Therefore, both Deep_FCN-Net and SRM+LRM-Net methods can be applied to detect boundaries from the regular agricultural fields also in cadastral survey. This also will be helpful in quantifying how much of the land is cropland.

Application of Deep_FCN-Net and MS-FuseNet

In this study, we have taken the FCN-DK5 network inspired by (Persello & Stein, 2017). We improved and adapted this network to the Deep_FCN-Net for boundary detection. Based on the accuracy generated, Deep_FCN-Net performed better than the FCN-DK5 by an increment of 10%. This is because, at the methodological level, the Deep_FCN-Net is deeper, and it uses 3×3 while FCN-DK5 uses 5×5 filters. The increasing of the filter size smoothens the output, therefore, thin features such as boundaries are eliminated. We then, compared the results by implementing these two methods without max-pooling layers. Additionally, we developed MS-FuseNet which is also a novelty inspired by Bergado et al., (2018) that fuses the panchromatic and multispectral bands. Furthermore, we adapted and improved this technique by developing a network that fuses multiresolution multispectral bands of Sentinel-2 within the network and detecting the boundaries at the same time. However, this method does not perform well compared to the Deep_FCN-Net based on the accuracy results for the AFB class as shown in comparison section from the previous chapter. This can be attributed to both complexity of its architectural design and complexity of the detection of the agricultural field boundaries.

Application of SRM-Net and SRM+LRM-Net

We designed SRM-Net to automatically detect boundaries at higher resolution of 5 m by applying the transposed convolution layer within FCN. Then, we designed SRM+LRM-Net that uses the contextual information in the label space. The importance of SRM+LRM-Net is that it refines the output from SRM-Net by reducing the noise and increasing the capability of separating different fields. The results on the stability assessment of SRM+LRM-Net showed that, this network is applicable in detecting boundaries as its performed similar to the Deep_FCN-Net from RapidEye with the slightly decrease in the F-Score for the class boundary as presented in the sub-section 5.5.2. Therefore, this study presents an opportunity of using open and free data of Sentinel-2 to automatically detect boundaries at higher resolution of 5 m.

Deep learning methods developed in this study (Deep_FCN-Net and the SRM+LRM-Net) are robust for detecting the agricultural fields. The methods are replicable and scalable (upscale) to the whole province of

Flevoland. This is because the training data used for training the networks were good representative (Figure 4.5) for the whole province of Flevoland. Therefore, it is feasible to produce the map over the large area such as the entire province of Flevoland.

6.2. Accuracy assessment strategy

F-Score is an accuracy assessment method commonly used to assess land cover classification maps. We use the F-score accuracy assessment test to assess the performance of our methods in detecting the agricultural field boundaries. Arguably, F-score best to assessment the accuracy of multi-class classification through analysing the accuracy per specific class (Hossin & Sulaiman, 2015), rather than assessing the overall aggregate accuracy for all classes. In this study, our aim was to assess the methods performance in detecting the agricultural field boundaries, we considered the boundaries as one class against the rest. This is because there are unbalanced classes whereas the class boundary is much smaller than the class rest. Thus, in this specific case, the F-score was ideal and reliable accuracy assessment method such that it balances the size of the two classes.

In this study, we assess the accuracy of the methods outputs against reference data. The reference data was rasterized from its original format vector to allow for the automatic accuracy analysis from FCN method. The FCN method employs the pixel by pixel comparison to assess for the accuracy of classification results from the reference data. Despite the wide application of this method and its automated performance, the pixel by pixel approach requires that both (classification output and reference) are in raster format, thus forces for the rasterization of the reference data. Ordinarily, rasterization of vector data, especially polyline vector, results to the loss of data quality because of the stair-like structure of the rasterized lines. Such quality loss may damage the accuracy assessment. Using vector data as the reference data for accuracy assessment of the boundaries would be ideal. Arguably, vector format for reference data increase the classification output accuracy, especially when we consider buffering the feature. This technique will allow us to include location tolerance, hence increasing the assessment accuracy.

6.3. Limitation of the methods

In this study, all four methods we developed (Deep_FCNet, MS-FuseNet, SRM-Net and SRM+LRM-Net) are time consuming. This is because, the methods are deep as they comprise large number of convolutional layers. Therefore, the methods need more processing power for their operational. Also, all methods produced fragmented boundary maps. These outputs depict that, the boundary detection problem using FCN is a challenge because the networks learn small and thin features such as lines. This limitation also presented in the Musyoka, (2018) work.

6.4. Final remarks

Considering the complexity of the problem (detection of boundaries from Sentinel-2) and because our F-Score accuracy assessment is tolerating any small shift, then, the results at 10 m and 5 m resolution from Deep_FCNet and SRM+LRM-Net respectively are reasonable results. These results present a good opportunity of using higher multispectral of Sentinel-2 image which is also open and freely available. The potential advantages of Sentinel-2 image used because VHR images are commercial. Although, Sentinel-2 having the low spatial resolution of 10 m and limited when we use them in the FCN because of mismatch of spatial resolution of its bands. Using 10 m spatial resolution of Sentinel-2 image, we produced the AFB with the higher spatial resolution of 5 m similar to using high resolution image which is commercial such as RapidEye with a slight decrease in the F-score as explained in sub-section 5.5.2. Besides, there was a negligible shift in the boundaries' locations between input image and reference data. This shift is not fully systematic because there are some parts having shifts and others not. The shift could be as a result of uncertainty on reference dataset that does not fully agree with the satellite image. It is probable that the

vector dataset was extracted from VHR image. Although, rasterization corrected this problem but not fully and this may have contributed to the low AFB results as this study did not set any buffering between boundaries as a tolerance.

7. CONCLUSION AND RECOMMENDATION

7.1. Conclusion

In this study, we developed two FCN methods namely Deep_FCNet and MS-FuseNet for detecting the agricultural field boundaries (AFB) from free and open data of Sentinel-2 satellite image. Additionally, we investigated the limitation of less detailed information from 10 m resolution by enhancing this resolution to 5 m by designed SRM-Net. Furthermore, we designed SRM+LRM-Net to examine the contextual spatial regularization by refining the labels on the SRM output.

We answer the research questions presented in Section 1.2.2 based on the findings from the sensitivity analysis and discussion.

Specific Objective 1.

1. How does FCN work based on the available literature in land cover land use (LCLU) classification?

In Chapter 2, Section 2.1 we discussed the deep learning methods including CNN and FCN. Both methods work with raster images to classify raster image into land use classes such as build-up. In this study, we chose FCN with dilated convolution, and we looked on specifically detecting the agricultural field boundaries.

Specific Object 2.

1. Which band(s) of the Sentinel-2 image are relevant for agricultural field boundary detection?

We chose the 8-band combination of visible bands, NIR, and 4-bands of vegetation from Sentinel-2 to detect the AFB. These bands combination produced impressive performance than the other combination as described in Section 5.13.

2. How can we properly fuse multiple bands within a single convolutional network trained end to end?

We designed a network with two input streams (convolutional layers of 3×3 filter sizes) that accepts multispectral bands of different spatial resolution (10 m and 20 m), 16 feature maps with the spatial dimension of 10 and 20 m resolution produced from the input streams that accepted 10 m and 20 m resolution images respectively. The max-pooling operation was used for down-sampling the feature maps of 10 m resolution to 20 m resolution such that to match the feature maps of 20 m resolution. Then, we applied a concatenation layer that concatenated these feature maps and continues with the series of convolution to these concatenated 32 feature maps as an input. Before using the classification layer, we applied the transposed convolution to up-sampled the feature maps of 20 m back to the 10 m resolution which is the highest spatial resolution of Sentinel-2.

3. What are the optimal hyperparameters values of the network architectures for AFB detection?

A patch size of 55 from 8-bands combination, training sample of size 5000 and 20 number of convolutional layers of 3×3 filter sizes produced best results in terms of F-Score for AFB detection from Deep_FCNet. For the best MS-FuseNet, we used patch sizes of 56 and 28 taken from 4-bands (visible and NIR) and 4 vegetation bands of 10 m and of 20 m resolution respectively. Also, we used 18 convolutional layers of 3×3 filters, and then trained the network using training samples of 5000.

Specific Object 3.

1. How can we design a deep learning network to properly up-sample feature maps at a higher resolution?

We designed SRM-Net that was up-sample the feature maps from 10 m to 5 m resolution from Sentinel-2 by arranging the network in sequential order. Convolutional layer followed by a non-linear rectifier and we incorporated the batch normalization between these two layers. In total, there are 24 convolutional layers. Then, we placed the transposed convolutional layer at the end of the convolutional layers. This layer initialized its weight randomly and its used to enhance the spatial dimension of feature maps from 10 m to 5 m by using an up-sample factor two.

2. What level of accuracy determines the stability of designed SRM-Net?

We assessed the stability of the SRM-Net by comparing its results in terms of F-Score with the Deep_FCNet using RapidEye satellite image. Ideally, both images of Sentinel-2 and RapidEye overlap in the same area. However their dates are different by one month, but there is no significant difference between these two data. A Sentinel-2 was taken on 26th September 2016 and RapidEye was taken on 31st August 2016.

Specific Object 4.

1. How can we incorporate spatial regularization in the deep learning network?

We incorporated spatial regularization by refining the labels of SRM output. We did this by performing the convolutional operations on the prediction of the SRM-Net and estimate the second prediction later.

7.2. Recommendation

In the future work, we can develop the strategies to produce segmentation where fragmented boundaries are connected to obtain close contours. We also can perform statistical tolerance analysis on Deep_FCNet and SRM_LRM-Net. Also, we could compare Deep_FCNet with other methods such as using global Pb and e-Cognition. Additionally, we could apply the Deep_FCNet on the other study areas than Flevoland that have irregular fields Finally, we could develop a boundary detection method from Sentinel-2 using SegNet technique.

LIST OF REFERENCES

- Andrea, C.-C., Mauricio Daniel, B. B., & Jose Misael, J. B. (2017). Precise Weed and Maize Classification through Convolutional Neuronal Networks. In *2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)* (pp. 1–6). Salinas: IEEE. <https://doi.org/10.1109/ETCM.2017.8247469>
- Atkinson, P. M. (2008). Super-Resolution Mapping Using the Two-Point Histogram and Multi-Source Imagery. In *geoENV VI – Geostatistics for Environmental Applications* (pp. 307–321). Dordrecht: Springer Netherlands. https://doi.org/10.1007/978-1-4020-6448-7_26
- Belgiu, M., & Csillik, O. (2018). Sentinel-2 Cropland Mapping Using Pixel-Based and Object-Based Time-Weighted Dynamic Time Warping Analysis. *Remote Sensing of Environment*, *204*, 509–523. <https://doi.org/10.1016/J.RSE.2017.10.005>
- Bergado, J. R., Persello, C., & Gevaert, C. (2016). A Deep Learning Approach to the Classification of Sub-Decimetre Resolution Aerial Images. In *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)* (pp. 1516–1519). Beijing: IEEE. <https://doi.org/10.1109/IGARSS.2016.7729387>
- Bergado, J. R., Persello, C., & Stein, A. (2018). Recurrent Multiresolution Convolutional Networks for VHR Image Classification. *IEEE Transactions on Geoscience and Remote Sensing*, *56*(11), 1–14. <https://doi.org/10.1109/TGRS.2018.2837357>
- Bivand, R., Keitt, T., Rowlingson, B. (2018). Bindings for the “Geospatial” Data Abstraction Library [R package rgdal version 1.3-9]. Comprehensive R Archive Network (CRAN). Retrieved from <https://cran.r-project.org/web/packages/rgdal/index.html>
- Bivand, R., Rundel, C. (2018). Interface to Geometry Engine - Open Source (‘GEOS’) [R package rgeos version 0.4-2]. Comprehensive R Archive Network (CRAN). Retrieved from <https://cran.r-project.org/web/packages/rgeos/index.html>
- Debats, S. R., Luo, D., Estes, L. D., Fuchs, T. J., & Caylor, K. K. (2016). A Generalized Computer Vision Approach to Mapping Crop Fields in Heterogeneous Agricultural Landscapes. *Remote Sensing of Environment*, *179*, 210–221. <https://doi.org/10.1016/J.RSE.2016.03.010>
- ESA. (2018a). Sen2Cor | STEP. Retrieved October 1, 2018, from <http://step.esa.int/main/third-party-plugins-2/sen2cor/>
- ESA. (2018b). Spatial - Resolutions - Sentinel-2 MSI - User Guides - Sentinel Online. Retrieved August 24, 2018, from <https://earth.esa.int/web/sentinel/user-guides/sentinel-2-msi/resolutions/spatial>
- European Commission. (2017). *Cap Explained. Direct payments for farmers 2015-2020*. <https://doi.org/10.2762/572019>
- FME. (2018). FME | Data Integration Platform | Safe Software. Retrieved September 28, 2018, from <https://www.safe.com/how-it-works/>
- García-Pedrero, A., Gonzalo-Martín, C., & Lillo-Saavedra, M. (2017). A Machine Learning Approach for Agricultural Parcel Delineation through Agglomerative Segmentation. *International Journal of Remote Sensing*, *38*(7), 1809–1819. <https://doi.org/10.1080/01431161.2016.1278312>
- Heltin Genitha, C., & Vani, K. (2010). Super Resolution Mapping of Satellite Images using Hopfield Neural Networks. In *Recent Advances in Space Technology Services and Climate Change 2010 (RSTS & CC-2010)* (pp. 114–118). Chennai: IEEE. <https://doi.org/10.1109/RSTSCC.2010.5712813>
- Hijmans, R. J. (2018). Geographic Data Analysis and Modeling [R package raster version 2.8-19]. Comprehensive R Archive Network (CRAN). Retrieved from <https://cran.r-project.org/web/packages/raster/index.html>

- Ioffe, S., & Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, *53*(3), 279–284. <https://doi.org/10.1007/s13398-014-0173-7.2>
- Ji, C. Y. (1996). Delineating agricultural field boundaries from TM imagery using dyadic wavelet transforms. *ISPRS Journal of Photogrammetry and Remote Sensing*, *51*(6), 268–283. [https://doi.org/10.1016/0924-2716\(95\)00017-8](https://doi.org/10.1016/0924-2716(95)00017-8)
- Lebourgeois, V., Dupuy, S., Vintrou, É., Ameline, M., Butler, S., & Bégué, A. (2017). A Combined Random Forest and OBIA Classification Scheme for Mapping Smallholder Agriculture at Different Nomenclature Levels Using Multisource Data (Simulated Sentinel-2 Time Series, VHRS and DEM). *Remote Sensing*, *9*(3), 259. <https://doi.org/10.3390/rs9030259>
- Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013). Rectifier Nonlinearities Improve Neural Network Acoustic Models. *Semantic Scholar*, *30*(1), 3. [https://doi.org/10.1016/0010-0277\(84\)90022-2](https://doi.org/10.1016/0010-0277(84)90022-2)
- Mandryk, M., Reidsma, P., Kartikasari, K., van Ittersum, M., & Arts, B. (2015). Institutional constraints for adaptive capacity to climate change in Flevoland’s agriculture. *Environmental Science & Policy*, *48*, 147–162. <https://doi.org/10.1016/J.ENVSCI.2015.01.001>
- Musyoka, G. M. (2018). *Automatic Delineation of Small Holder Agricultural Field Boundaries Using Fully Convolutional Networks*. Enschede: University of Twente Faculty of Geo-Information and Earth Observation (ITC).
- PDOK. (2018). Introduction - PDOK. Retrieved October 1, 2018, from <https://www.pdok.nl/introductie?articleid=1948958>
- Persello, C., & Stein, A. (2017). Deep Fully Convolutional Networks for the Detection of Informal Settlements in VHR Images. *IEEE Geoscience and Remote Sensing Letters*, *14*(12), 2325–2329. <https://doi.org/10.1109/LGRS.2017.2763738>
- Rydberg, A., & Borgefors, G. (2001). Integrated Method for Boundary Delineation of Agricultural Fields in Multispectral Satellite Images. *IEEE Transactions on Geoscience and Remote Sensing*, *39*(11), 2514–2520. <https://doi.org/10.1109/36.964989>
- Shao, Z., & Cai, J. (2018). Remote Sensing Image Fusion with Deep Convolutional Neural Network. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *11*(5), 1656–1669. <https://doi.org/10.1109/JSTARS.2018.2805923>
- Sonobe, R., Yamaya, Y., Tani, H., Wang, X., Kobayashi, N., & Mochizuki, K. (2018). Crop classification from Sentinel-2-derived vegetation indices using ensemble learning. *Journal of Applied Remote Sensing*, *12*(02), 1. <https://doi.org/10.1117/1.JRS.12.026019>
- Springenberg, J. T., Dosovitskiy, A., Brox, T., & Riedmiller, M. (2015). Striving for Simplicity: The All Convolutional Net. In *3rd International Conference on Learning Representations (ICLR2015)*. San Diego. Retrieved from <http://arxiv.org/abs/1412.6806>
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958. Retrieved from <http://jmlr.org/papers/v15/srivastava14a.html>
- Tolpekin, V. A., & Stein, A. (2009). Quantification of the Effects of Land-Cover-Class Spectral Separability on the Accuracy of Markov-Random-Field-Based Superresolution Mapping. *IEEE Transactions on Geoscience and Remote Sensing*, *47*(9), 3283–3297. <https://doi.org/10.1109/TGRS.2009.2019126>
- Turker, M., & Kok, E. H. (2013). Field-Based Sub-Boundary Extraction from Remote Sensing Imagery using Perceptual Grouping. *ISPRS Journal of Photogrammetry and Remote Sensing*, *79*, 106–121. <https://doi.org/10.1016/j.isprsjprs.2013.02.009>
- Wang, Q., Shi, W., Li, Z., & Atkinson, P. M. (2016). Fusion of Sentinel-2 images. *Remote Sensing of Environment*, *187*, 241–252. <https://doi.org/10.1016/J.RSE.2016.10.030>

- Yalcin, H., & Razavi, S. (2016). Plant Classification using Convolutional Neural Networks. In *2016 Fifth International Conference on Agro-Geoinformatics (Agro-Geoinformatics)* (pp. 1–5). Tianjin: IEEE. <https://doi.org/10.1109/Agro-Geoinformatics.2016.7577698>
- Zhou, Z., & Li, S. (2017). Peanut Planting Area Change Monitoring from Remote Sensing Images Based on Deep Learning. In *2017 4th International Conference on Systems and Informatics (ICSAI)* (pp. 1358–1362). IEEE. <https://doi.org/10.1109/ICSAI.2017.8248497>

LIST OF APPENDICES

Appendix 1: The final experimental results with the 4 classes (AFB, crop, grass, other) from all four methods namely; Deep_FCNet, MS-FuseNet, SRM-Net and SRM+LRM-Net.

Results of the final experiment using Deep_FCNet

Tile	Type I error	Type II error	Precision	Recall	F-Score
TS1	0.09	0.36	0.47	0.64	0.54
TS2	0.15	0.44	0.38	0.56	0.45
TS3	0.10	0.40	0.50	0.60	0.54
TS4	0.11	0.34	0.51	0.66	0.57
TS5	0.13	0.40	0.45	0.61	0.52

Results of the final experiment using MS-FuseNet

Tile	Type I error	Type II error	Precision	Recall	F-Score
TS1	0.09	0.37	0.46	0.63	0.53
TS2	0.13	0.50	0.39	0.50	0.44
TS3	0.09	0.41	0.51	0.59	0.55
TS4	0.10	0.37	0.52	0.63	0.57
TS5	0.13	0.44	0.43	0.56	0.49

Results of the final experiment using SRM-Net

Tile	Type I error	Type II error	Precision	Recall	F-Score
TS1	0.08	0.59	0.24	0.41	0.30
TS2	0.13	0.66	0.18	0.34	0.24
TS3	0.09	0.63	0.26	0.37	0.30
TS4	0.11	0.59	0.26	0.41	0.32
TS5	0.11	0.61	0.25	0.39	0.30

Results of the final experiment using SRM+LRM-Net

Tile	Type I error	Type II error	Precision	Recall	F-Score
TS1	0.09	0.58	0.23	0.42	0.30
TS2	0.13	0.63	0.20	0.37	0.26
TS3	0.09	0.63	0.25	0.37	0.30
TS4	0.11	0.59	0.25	0.41	0.31
TS5	0.12	0.62	0.23	0.38	0.29