satellite classification by deep learning

Kiarash Borooshan[1*]

Abstract

Deep-learning has ability to detect or predict each feature of natural resources. To achieve this goal, the network needs optimum amount of data that has given from a few years ago of objects. After all, the gathered data, trained for once and makes a unique model. Each object has specific model and can to decryption object without any train data. This paper has proceeded deep learning method to detect the water, vegetation (agriculture) and bare land. Because author did not have any spatial data to train the network. Additionally, water, vegetation (agriculture) and bare land can detect easily by NDVI[2]. First of all row data-set has made by NDVI's threshold, and balanced it. Then separated to train, test and validation data. Next step, different machine and deep-learning method applied and accuracy has calculated. The MODEL has built by applying machine and deep-learning method to all balanced data without separate data. Finally the MODEL has acquired, applied to two case studies. Case study small area and case study big area. All results seem good and probably it will better with precise data. Most of incorrect classify has located in the border, between two different features.

Keywords :

deep learning, machine learning, regression, satellite image classification,

1 Master Degree in remote sensing at science and research university of Tehran
* Correspondence : kiarash.borooshan@gmail.com
2 Normalized Difference Vegetation Index

1. Introduction

Detection and classification is the most important task that a human is doing every moment to make a better decision. The present environmental problem of our planet is growing and future sustainability, needs spurred great interest in monitoring the main variation of the earth system [1]. Since all forces interacting to shape the world landscape are intrinsically dynamic, the earth environment has always been affected by the change [2]. Very good results have been made for detection of images taken by the camera. For example face, car and other detection. That method has mentioned, can use for satellite images.

The common definition of remote sensing is “science and some extent art of acquiring information about the Earth’s surface without any direct contact”. A machine learning approach with implications for both art and science to emphasize the concept of the “System of Systems Engineering” to accommodate an all inclusive capability of sensing, monitoring, modeling and decision making to mitigate the human induced stress on the environment [3].

2. Materials and Methods

To understand the information to extract from remote sensing data is crucial to understanding characteristic of electromagnetic radiation in term of their wavelength and frequency. Figure 1 hows Remote sensing brief. Part (A) describe it in seven part: (A-A)Energy source or illumination, (A-B)Radiation and Atmosphere, (A-C)Interaction with the target, (A-D)Recording of Energy by the Sensor, (A-E)Transmission, Reception and Processing, (A-F) Interpretation and Analysis and (A-G)Application.

Energy source (Electromagnetic radiation) illuminated to the target and emitted to the sensor. This energy is form of electromagnetic radiation (B) and consist of an electrical field and magnetic field. Both these fields travel at the speed of light. The length of one wave cycle called wavelength and represented by Greek letter lambda(ƛ).

And Frequency refers to the number of cycles of wave passing a fixed point per unit of time. The electromagnetic spectrum according to their ranges of frequency or wavelength divided several region and named different bands. Such as x-ray, visible, infrared, microwave, etc. (D) includes a series of spectral signatures for various covers [1].

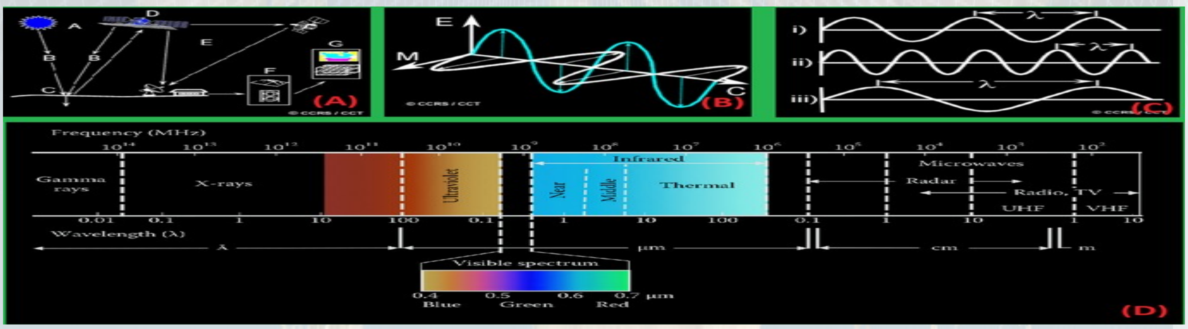


Figure 1: Fundamental of remote sensing

3. Case study (small and big)

Lake Urmia Figure 2 s located between West and East Azerbaijan provinces. The largest and endorheic lake in the Middle East and has a wealth of archaeological sites going back to the Neolithic period. Artifact settlements have found from about 7000 BCE and later. 200 thousand years Vegetation and lake level recorded by palynological investigation on long core from Urmia Lake[3].

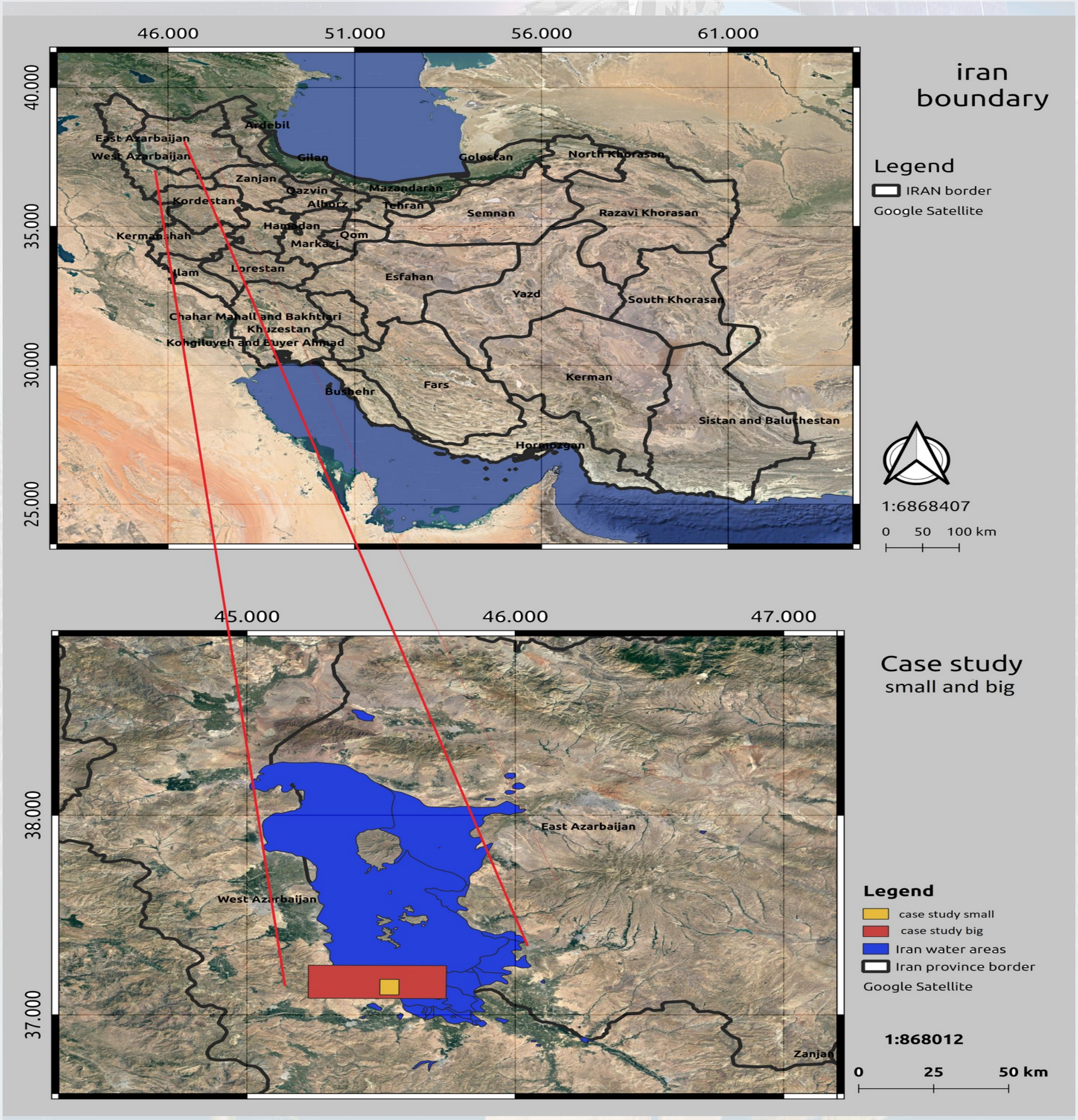


Figure 2: Lake Urmia

3 https://en.wikipedia.org/wiki/Lake_Urmia

4. Data and Process

The manner's interact between solar radiation and the Earth's surface, can tell us much about surface's properties such as chemical and physical. Spectral reflectance signature or spectral signature refers to the reflectance behavior of an object over various wavelengths of the electromagnetic spectrum [2]. From the basis spectral signatures Figure 3 describe objects from remote sensing measurements in the solar region of the EM spectrum. But a problem is, these spectral signatures are not constant for each cover. Atmospheric component, land cover variation, soil and geological substrate, solar illumination, slope and aspect are some external conditions can affect spectral signatures.

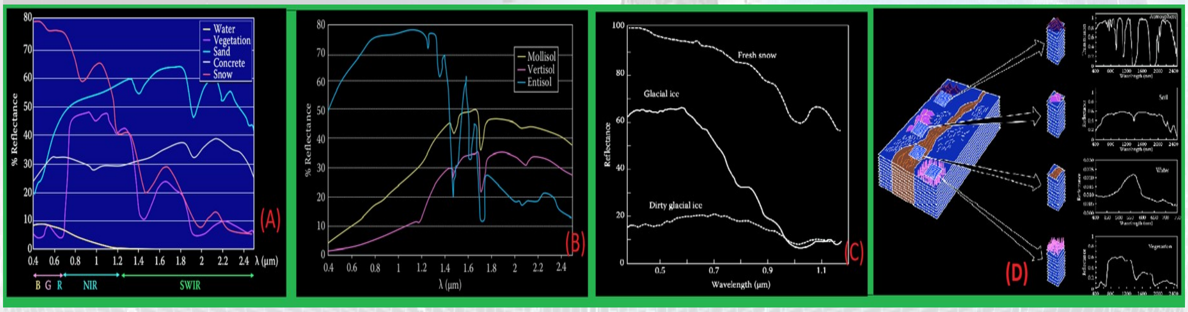


Figure 3: spectral signatures

4.1. Spectral signature Library

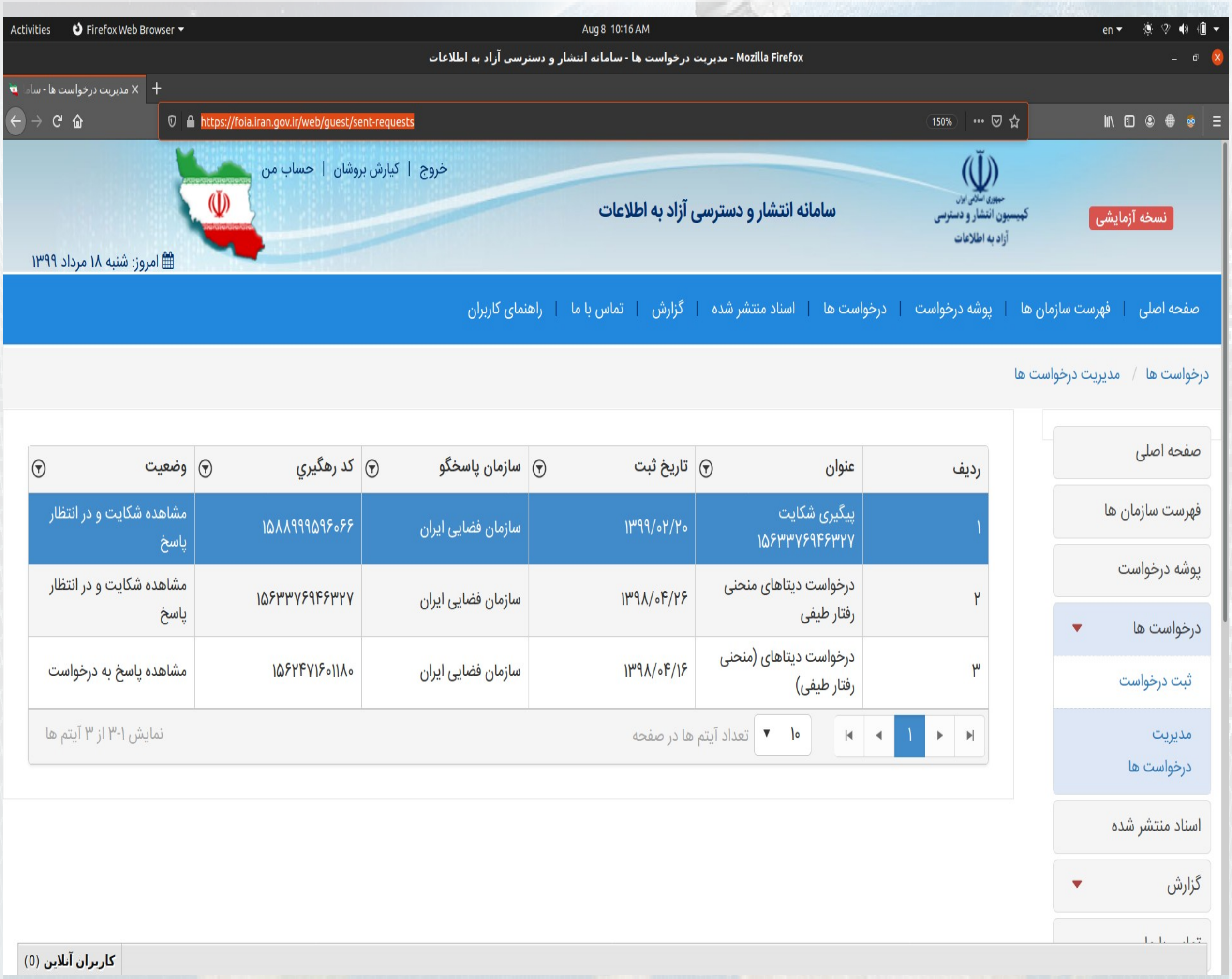
The spectral signature library is a collection of variable of spectral reflectance signatures according to the wavelength of discrete materials such as agriculture, minerals geology, natural resources, etc. Spectral signature is [1] produced by the spectrometer Figure 4. Common spectrometers [4] are ASD FieldSpec® series, Geophysical Environmental Research (GER) series, Spectral Evolution series, SpectraScan® series and Ocean optical (USB2000) series. Each one has different Spectral ranges, resolution, number of bands, FOV[4].

Figure 4: Spectrometer, (A): SpectrScan, (B): GER, (C): ASD, (D): Spectra Evolution

4 Field of view (Degree)

The USGS[5], ECOSIS[6], JPL[7] are some of sites that shared totally more than hundred thousand free spectral signature library included agriculture, Artificial materials, Coating, Liquids, organic compounds, soil and mixture vegetation, mineral, man-made and etc. God bless them. More than 48000 spectral signature was between 0.4 ~ 2.5 µm.

Iranian space agency (ISA) [8] has twelve hundred spectral signature data. The author, according freedom of information (FOI)[9] laws Figure 5 has requested this data on Jun 2019, but ISA didn't give any data even one. Instead the geniuses ISA, answered "you can use the spectral signature library as (.JPG) format" Figure 6. The author couldn't find any article to progress how to use the spectral signature as (.JPG) format. If anyone knows about this fabulous and unique knowledge, please contact with the author.



ردیف	عنوان	تاریخ ثبت	سازمان پاسخگو	کد رهگیری	وضعیت
۱	پیگیری شکایت ۱۵۶۳۳۷۶۹۴۶۳۲۷	۱۳۹۹/۰۲/۲۰	سازمان فضایی ایران	۱۵۸۸۹۹۹۵۹۶۰۶۶	مشاهده شکایت و در انتظار پاسخ
۲	درخواست دیتاهای منحنی رفتار طیفی	۱۳۹۸/۰۴/۲۶	سازمان فضایی ایران	۱۵۶۳۳۷۶۹۴۶۳۲۷	مشاهده شکایت و در انتظار پاسخ
۳	درخواست دیتاهای (منحنی رفتار طیفی)	۱۳۹۸/۰۴/۱۶	سازمان فضایی ایران	۱۵۶۲۴۷۱۶۰۱۱۸۰	مشاهده پاسخ به درخواست

Figure 5: Freedom of information of IRAN

5 https://www.usgs.gov/
6 https://ecosis.org/
7 https://www.jpl.nasa.gov/
8 http://isa.ir/
9 https://foia.iran.gov.ir/

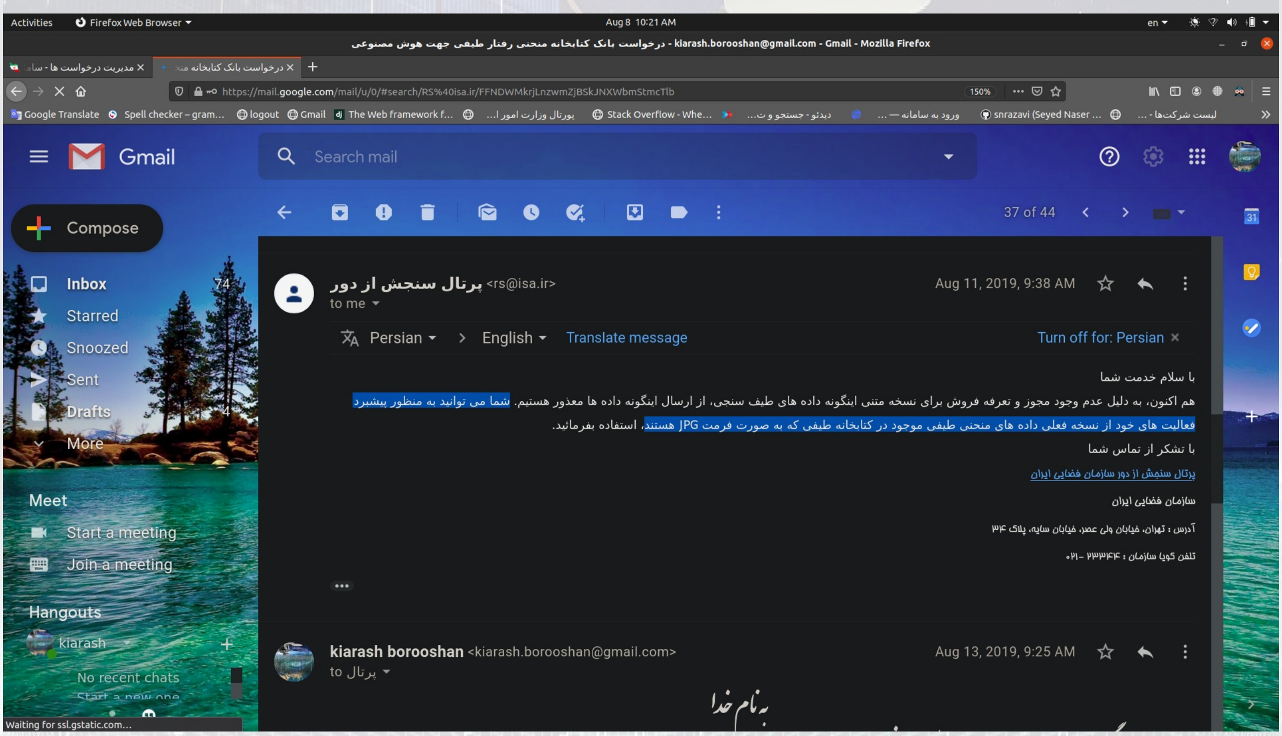


Figure 6: ISA answered: "use screenshot format of spectral library"

4.2. Satellite Images

The sensors in the optical wavelength range are divided multi-spectral Figure 7 -(A) and hyper-spectral Figure 7 -(B). These sensors have included aspect major differences [4]. The sensors can acquire image data in a few hundred narrow and contiguous spectral bands called hyper-spectral whereas multi-spectral measures image data in a few wide and discrete spectral ranges.

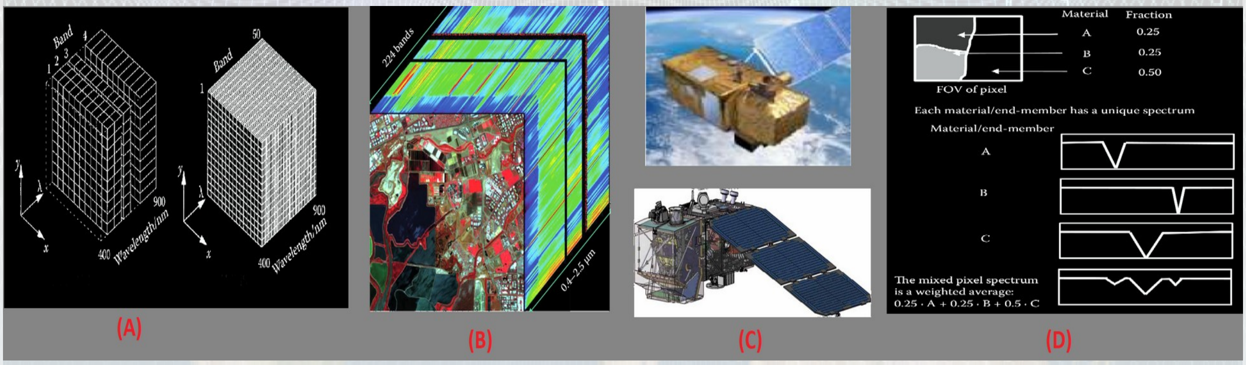


Figure 7: visualization (multi(A), hyper(B) - spectral), Sentinel(C), mixture pixel(D)

he Copernicus Open Access[10] provides complete, free and open access to Sentinel 1, 2, 3 and 5P user product. The Sentinel-2 Figure 7-(C) comprise twin polar-orbiting, wide-swath, high-resolution, multi-spectral imaging and will monitor variations in land surface every 5 days under cloud-free condition. The measured responses for each band of sentinel-2A and

10 https://scihub.copernicus.eu/dhus/#/home

Sentinel-2B multi-spectral instrument illustrated in Table 1.

The 8a band at 865 NM[11] in NIR[12] designed narrowness to avoid water vapor contamination but still sensitive to iron oxide content for soil. Band 1 at 443 NM in the Blue domain configured to precise aerosol correction of acquired data. Thin cirrus detection, managed using spectral band at 1375 NM (band 10) [5].

Table 1: measured spectral responses for each band of Sentinel-2

Band	(nm)Center (λ)	Spatial (m)	Purpose
1	443	60	Atmospheric correction (aerosol scattering) / costal aerosol
2	490	10	Sensitive to vegetation senescence, carotenoid, browning and soil background; atmospheric correction (aerosol scattering) / Blue
3	560	10	Green peak, sensitive to total chlorophyll in vegetation / Green
4	665	10	Maximum chlorophyll absorption / Red
5	705	20	Position of red edge; consolidation of atmospheric corrections / fluorescence baseline. / vegetation red edg
6	740	20	Position of red edge, atmospheric correction, retrieval of aerosol load. / vegetation red edg
7	783	20	Leaf Area Index (LAI), edge of the Near-Infrared (NIR) plateau. / vegetation red edg
8	842	10	LAI / NIR
8a	865	20	sensitive to total chlorophyll, biomass, LAI and protein; water vapour absorption reference; retrieval of aerosol load and type. / Narrow NIR
9	945	60	Water vapour absorption, atmospheric correction. / Water vapor
10	1375	60	Detection of thin cirrus for atmospheric correction. / SWIR cirus
11	1610	20	Sensitive to lignin, starch and forest above ground biomass. Snow/ice/cloud separation. / SWIR-1
12	2190	20	Assessment of Mediterranean vegetation conditions. Distinction of clay soils for the monitoring of soil erosion. Distinction between live biomass, dead biomass and soil, e.g. for burn scars mapping. / SWIR-2

The Sentinel-2 products two level granules of a fixed size. For level -0, -1A and -1B that are 25 km across track and 23 km along track in size. For ortho rectified product (level -1C and -2A) consist of 100 km by 100 km squared ortho images. Level 1B and 1C are top-of-atmosphere radiance in sensor geometry and Level 2A is Bottom-of-atmosphere reflectance in cartographic geometry.

5. Prepare satellite data

Deep-learning methods need labeled data. Author didn’t have any spatial data. So has made label data by supervising cluster method. NDVI, KNN[13] and Spectral clustering. This paper has used two case study and named case study small and case study big. Case study small is part of case study big. NDVI formula Illustration 2 has used for this paper Figure 10 (case study small) and Figure 11 (case study big). in fact 14 classes has detected by KNN method Figure 8 with distortion = 5000 (Elbow method) in case study small scene.

11 nano meter
12 Near infra-red
13 K-nearest neighbor

To made labeled data-set, NDVI value has Re-classed Figure 12 and Figure 13. Then labeled by threshold to 3 classes as Table 2 for both case study (small and big). Deep-learning methods needs balanced data-set. It means the number of labeled data-set should be same. The number of vegetation (agriculture) data was less. So the random same number of other labeled data (water and bare land) was picked. Then data-set separated to train, test and validation data. Next step, different machine and deep-learning method applied and accuracy has calculated. The MODEL has built by applying that methods to all balanced data without separate data. Finally the MODEL has acquired applied to two case studies (small and big)

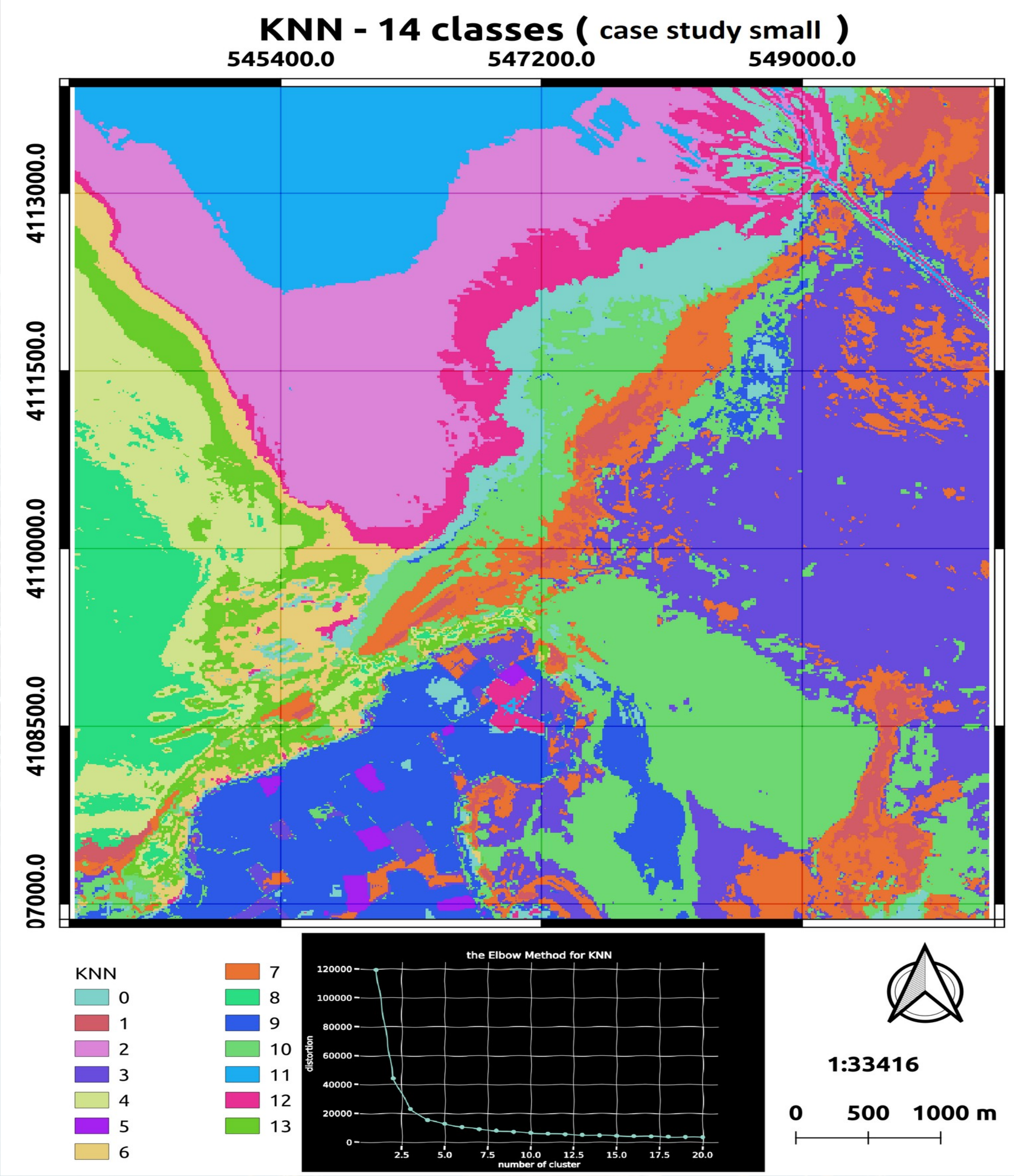


Figure 8: KNN for train area with the Elbow method

5.1. Convert Digital number to Reflectance

Radiance is somewhat depend on reflectance and is what is measured at sensor. Sentinel-2 images (Level-1C) are already provided in Top Of Atmosphere (TOA) Reflectance, scaled prior to output (ESA, 2015). top-of-atmosphere conversion[14] has showed Illustration 1.

$$\rho_k(i,j) = \frac{\pi \times CN_{k,NTDI}(i,j)}{A_{k,NTDI} \times E_s \times d(t) \times \cos(\theta_s(i,j))}$$

Illustration 1: Top of Atmosphere conversion

where:

- CNk,NTDIis the equalized numeric digital count of the pixel (i,j) with NTDI, the number of SENTINEL-2 TDI lines
- Es is the equivalent extra-terrestrial solar spectrum and depends on the spectral response of the SENTINEL-2 bands
- The component *d(t)* is the correction for the sun-Earth distance variation (see Equation 2). It utilises the inverse square law of irradiance, under which, the intensity (or irradiance) of light radiating from a point source is inversely proportional to the square of the distance from the source.

5.2. NDVI (Normalized Difference Vegetation Index)

NDVI formula has showed in Illustration 2 and applied to case study small and big. Then NDVI, Re-classed and labeled according Table 2. as mentioned deep-learning methods needs balanced data-set. It means the number of labeled data-set should be same. The number of vegetation (agriculture) data was less. So the random same number of other labeled data (water and bare land) was picked.

$$NDVI = \frac{(NIR - \mathrm{Red})}{(NIR + \mathrm{Red})}$$

Illustration 2: NDVI

Table 2: NDVI’s threshold

NDVI’s threshold	String label	Numeric label
Value < 0	Water	1
0 < Value < 0.2	bare land	2
Value > 0.2	agriculture	3

twelve bands of Sentinel, scatter plot for 10000 random data has showed in Figure 9 (band 10 has removed).

14 https://sentinel.esa.int/web/sentinel/technical-guides/sentinel-2-msi/level-1c/algorithm

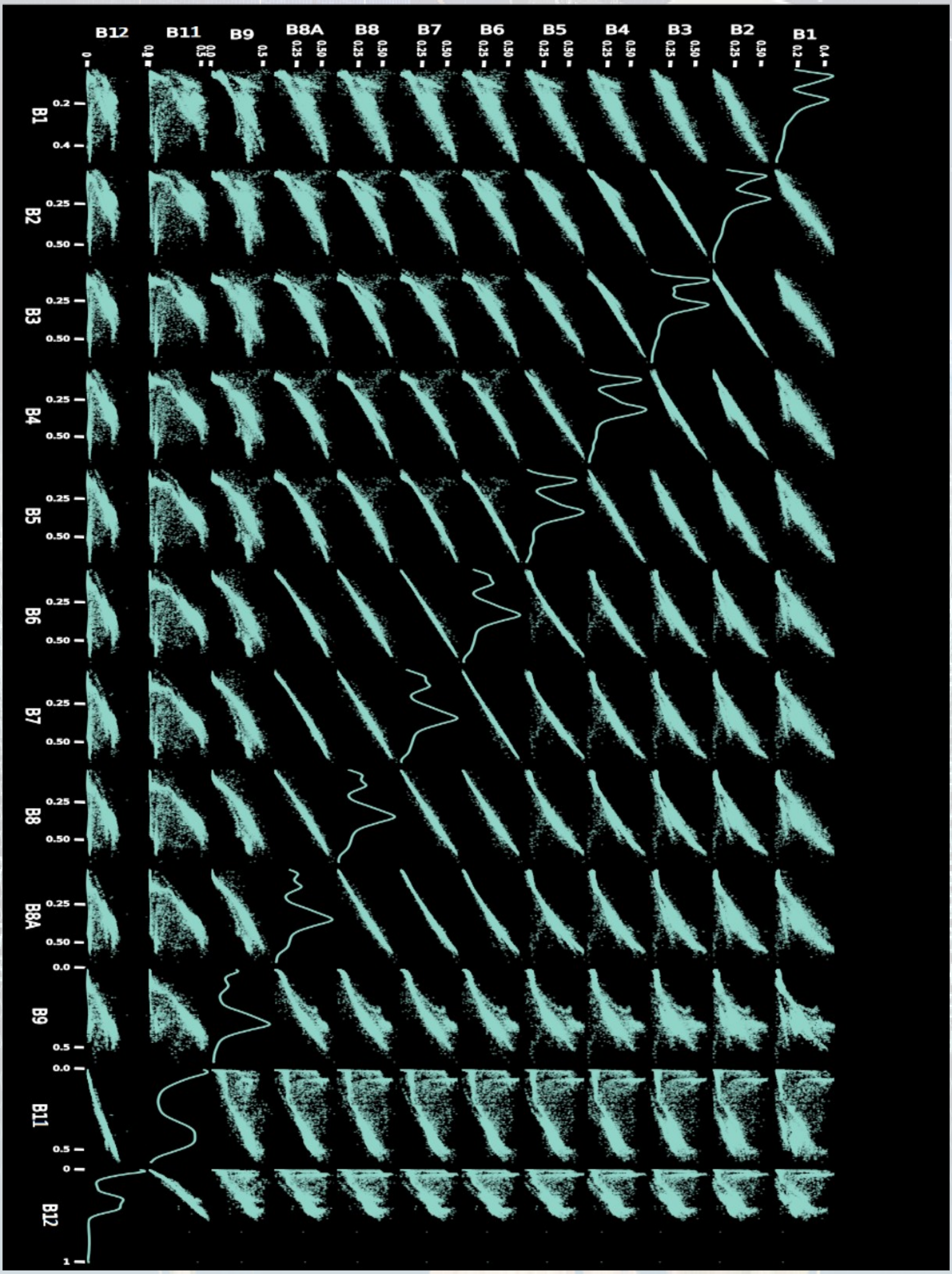


Figure 9: sentinel2 satellite bands scatter plot for 10000 random data

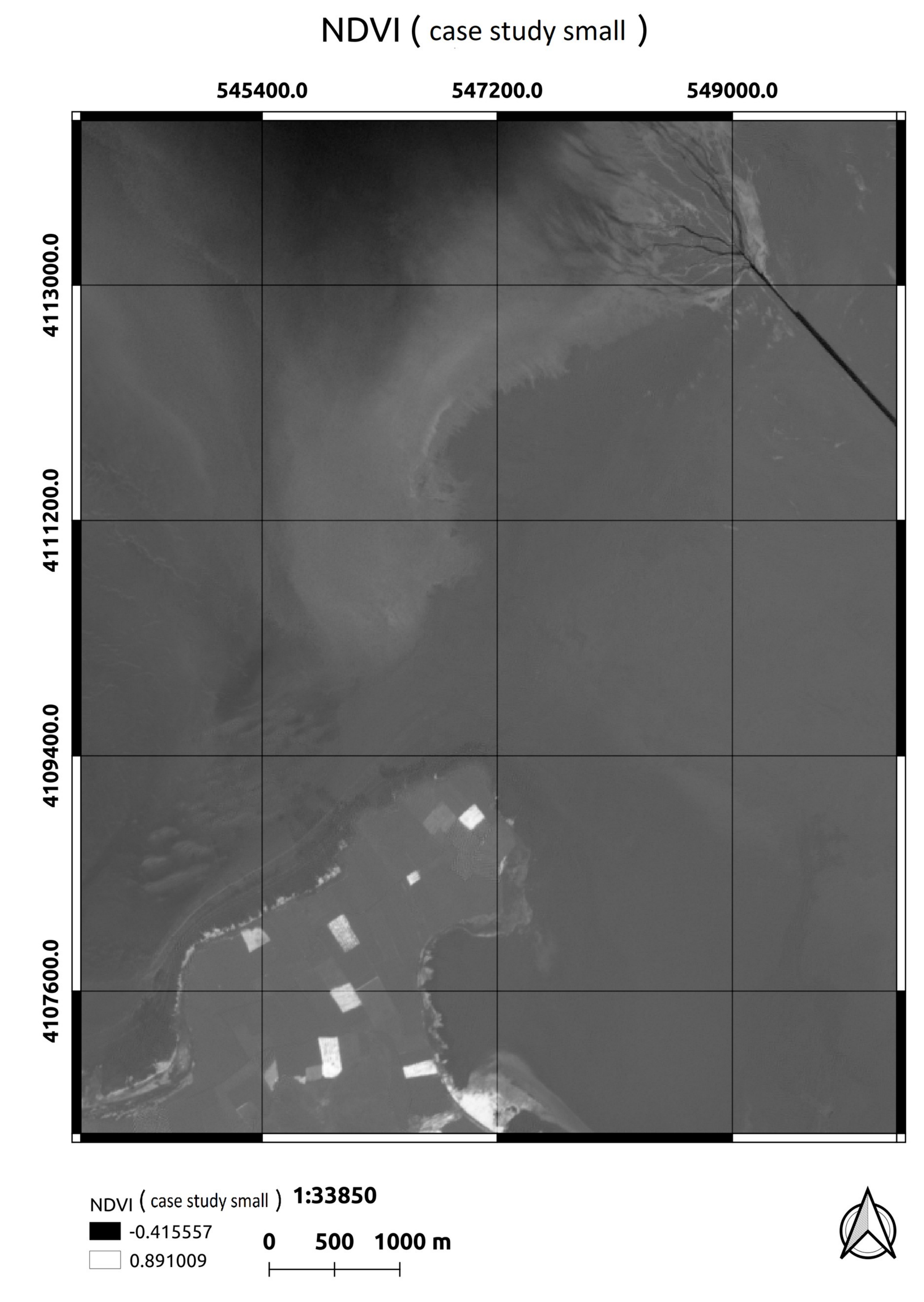


Figure 10: NDVI (case study small)

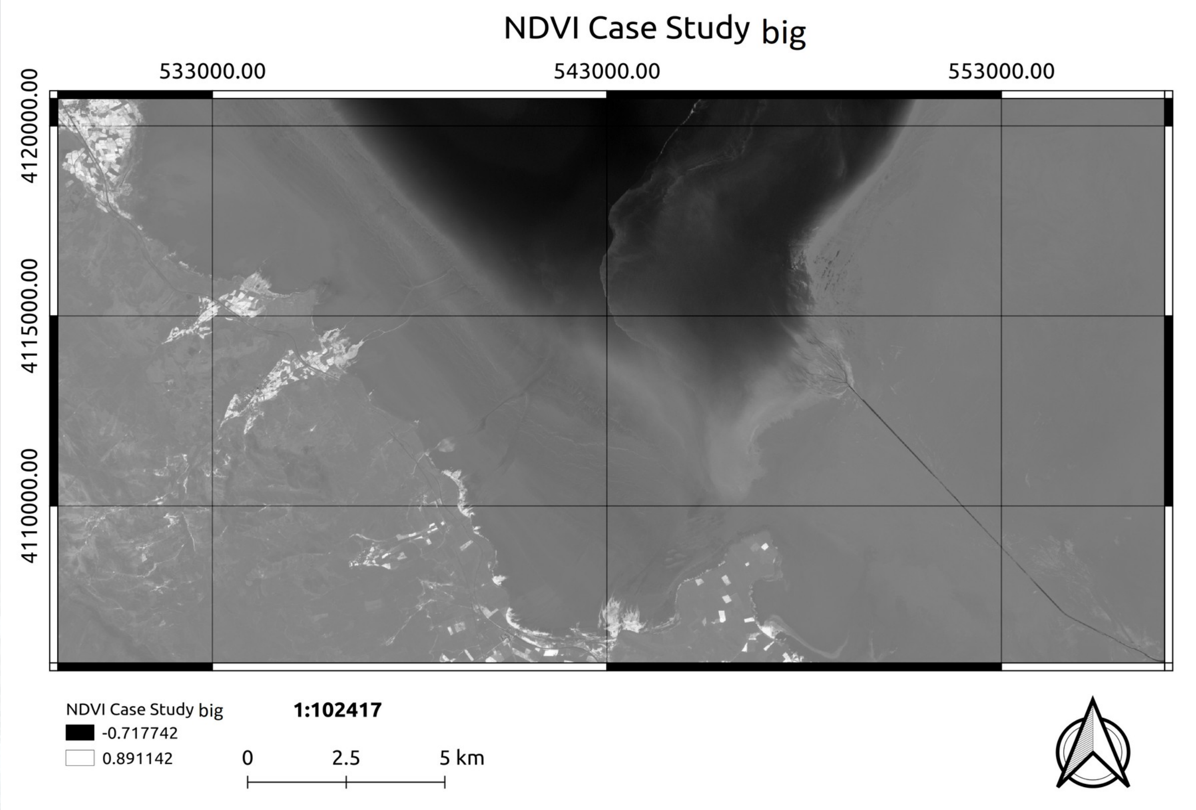


Figure 11: NDVI (case study big)

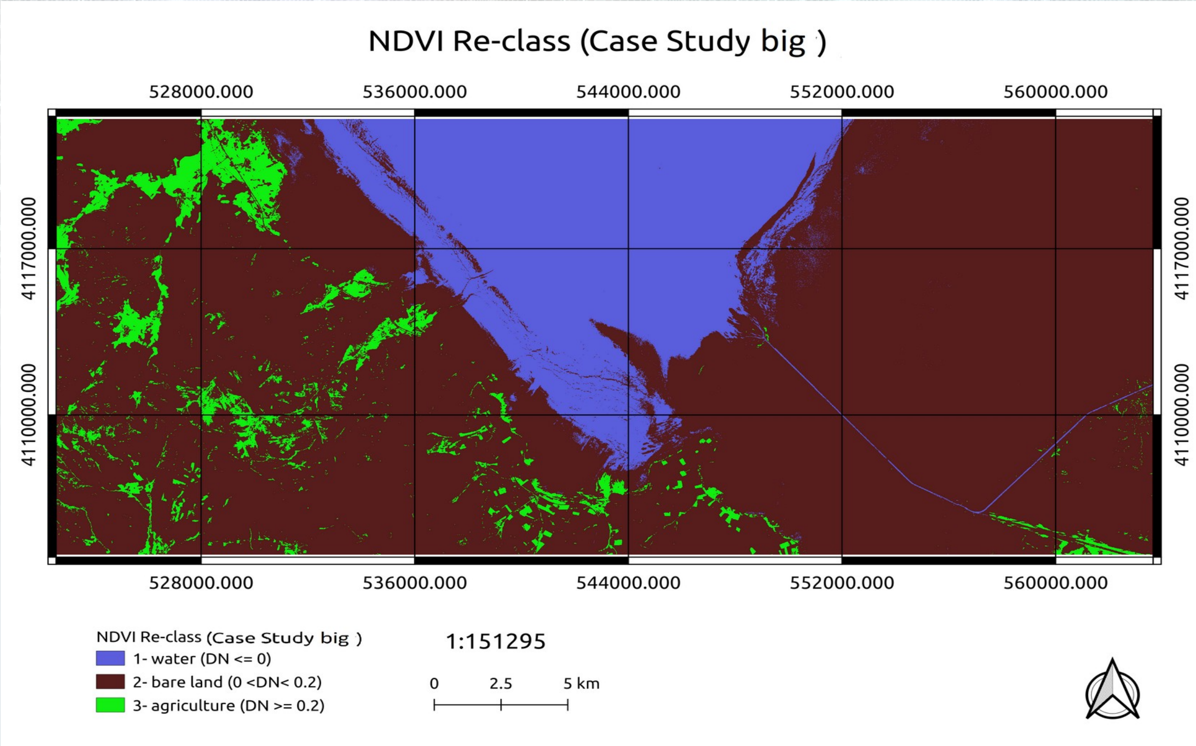


Figure 12: NDVI Re-class (case study big)

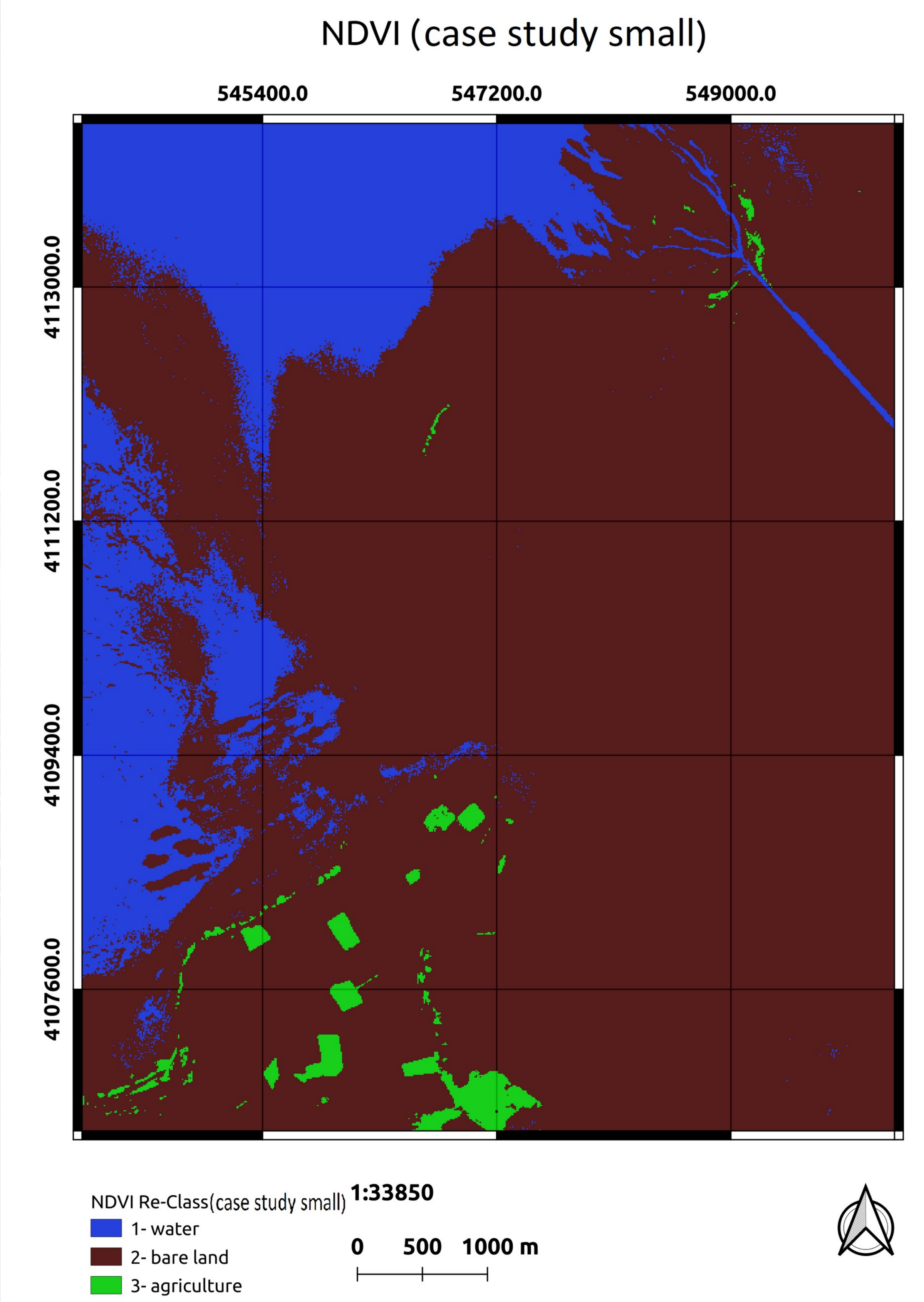


Figure 13: NDVI Re-class (case study small)

5.3. Spectral cluster

As mentioned the author didn't have any spatial data. So has used NDVI method to labeled data. But in regular way, you should to know how many classes you are using.

When the structure of your data is highly non-convex or measure of the center and spread of the cluster is not suitable description of complete cluster, Spectral Cluster is very useful[15]. Figure 14 shows characteristics of different clustering algorithms on data-sets that are still in 2D. As you mentioned, Spectral Cluster produced a good clustering result.

But Spectral clustering algorithm has ~ O(n^3) time complexity, and a fairly bad space complexity, since you are running out for memory with 16 GB RAM to process a ~0.8 GB data-set (10000x10000 array, assuming 64bit floats). It is therefore not suitable for large data-sets[16]. Your affinity matrix is a 57,600 by 57,600 by 64 (assuming that you use np.float64). Therefore, this is accounting for 26.5 GB. When calling the Spectral Clustering, the Laplacian matrix will be computed on a dense matrix creating again a matrix of 26.5 GB[17]. Author faced this message: "Memory Error: Unable to allocate array with shape (123328, 123328) and data type float64".

Instead the author has used KNN method and performed the Elbow method to find how many optimum classes are Figure 8 14 classes are the optimum number for case study small seance with distortion = 5000.

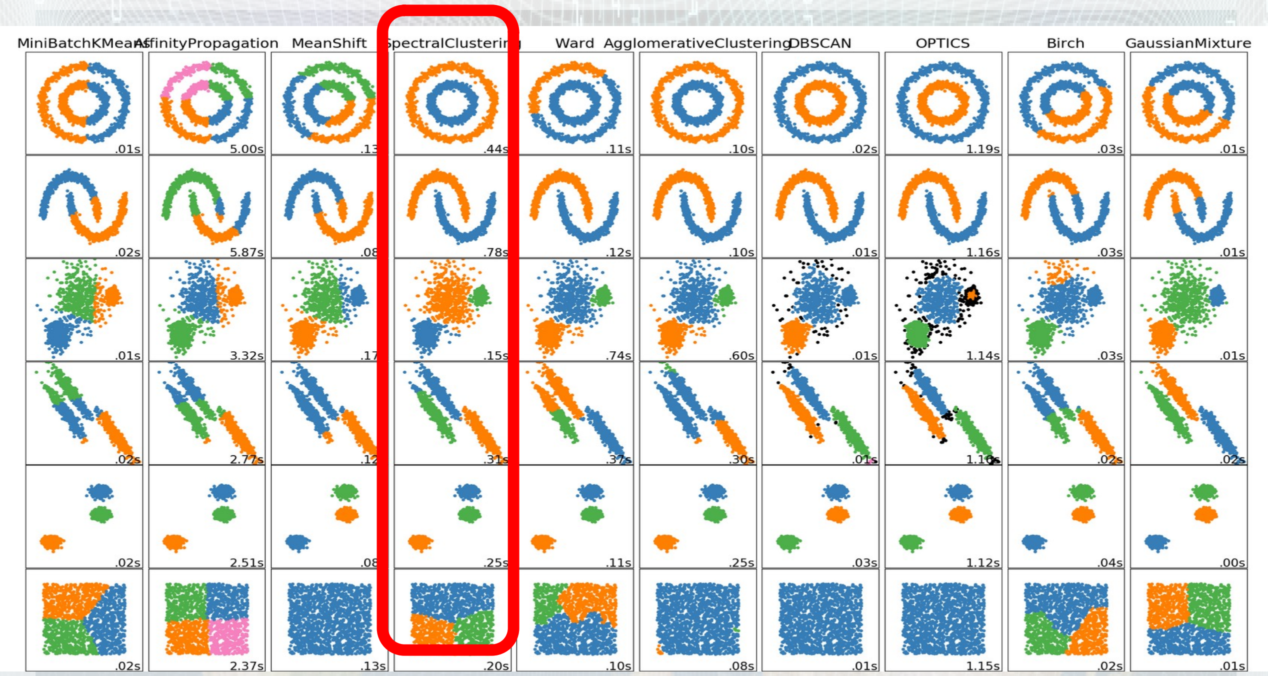


Figure 14: Comparing different clustering algorithms on toy data-sets

15 https://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html#

16 https://stackoverflow.com/questions/41429084/how-much-matrix-size-the-function-spectral-clustering-of-scikit-learn-can-handle

17 https://github.com/scikit-learn/scikit-learn/issues/8740

6. Model Development (machine learning)

Model development separated two main category. First for convert spectrometer's spectral signature to satellite's spectral signature. And second find optimum θ for satellite classification's method.

As Figure 7-(D) illustrated, most of the pixels are mixed some features. As mentioned spectral signature library is a unique fingerprint of objects. First of all, this paper assumes that the pixels are pure and no atmospheric noises interfere. So ML[18] can help to simulate and convert the spectral signature library to pixel's spectral. ML provides a deluge of remote sensing data automated methods of data analysis [6]. In other hand human would like the algorithm extract the task automatically by computer (machine) [6].

The ML divided (predictive or supervise), (descriptive or unsupervised) and (reinforcement learning) type to do mapping input x to output y [6]. ML may not be able to identify the process completely, but could construct a good and useful approximation.

Model is one of the most critical point in learning that defines or shows the template of relationship between the inputs and outputs [7]. Quantitative questions of the type how many or how much, can answer by supervised learning algorithm. Input data points are termed as independent or explanatory variable, while output is termed as a dependent variables. Input data samples (independent) along with output (dependent), trained by regression models [8]. regression summary has showed in Table 3

Table 3: regression summary

$h_\theta(x)=\theta_0+\theta_1 x$	Hypothesis
(θ_0,θ_1)	Parameters
$J=MSE=\frac{1}{n}\sum_{i=0}^{n}(y^i-t^i)^2=\frac{1}{n}\sum_{i=0}^{n}(x^i*w-t^i)^2=\frac{1}{2}\sum_{i=0}^{n}(h_\theta(x^{(i)})-y^{(i)})^2$	Cost functions
$\underset{\theta_0,\theta_1}{\text{minimize}}\ J(\theta_0,\theta_1)$	goal

6.1. convert spectrometer's spectral signature to satellite's spectral signature

As mentioned Figure 7D, pixels are not pure. Also Deep-learning methods need numerous and varied training data. spectrometer's spectral signature can help as a one pure pixel which is free of atmospheric noises. Spectrometer measures in hundred till thousands bands (according to type), but satellites (in this research (sentinel-2)) has twelve bands (band 10 omitted). Then needs to re-sample spectrometer's spectral signature to satellite's spectral signature by machine learning methods.This paper had used water spectral signature Figure 15. s seen this signature is so straight in 0.35 to 2.5 µm. Therefore, all methods were performed on 0 to 14 µm too.

18 Machine learning

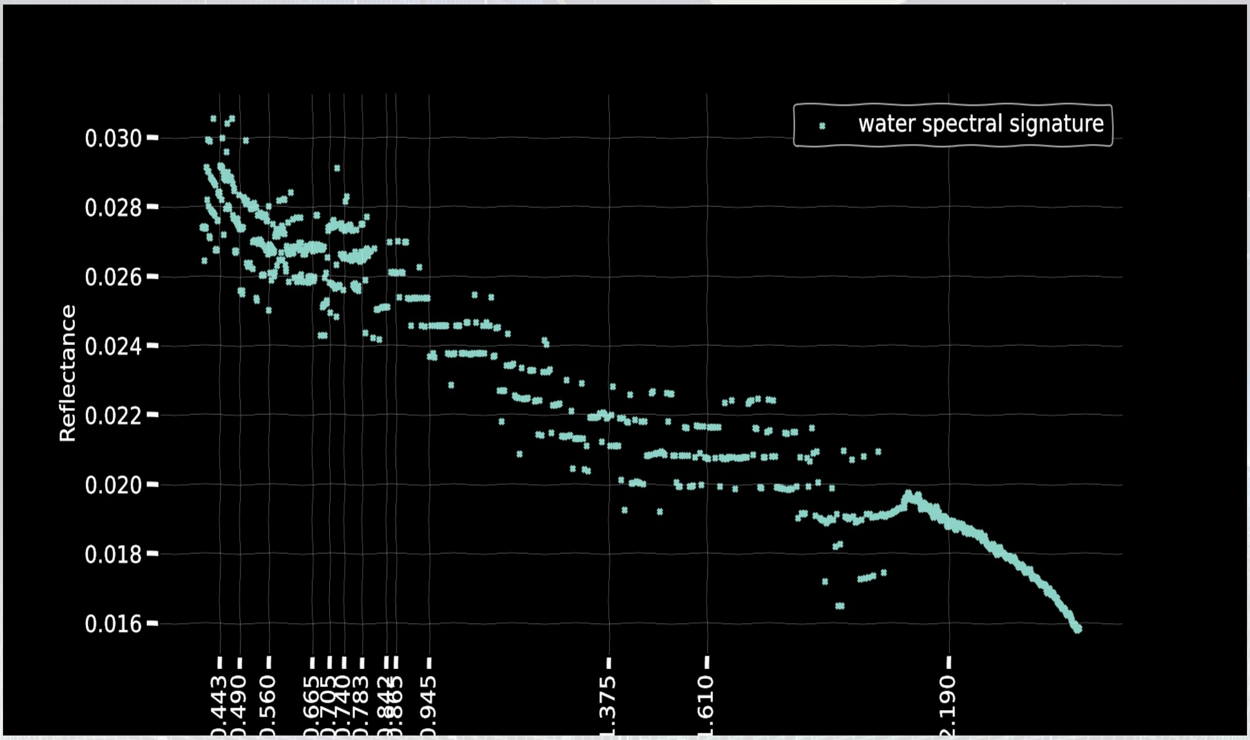


Figure 15: water spectral signature (0.35 to 2.5 µ meter)

6.1.1.1. Simple linear Regression

spectrometer's spectral signature to satellite's spectral signature [9]. Linear regression is known as one of the most widely use model for regression to represent the inner or scalar product between the input vector x and y [5]. a model works with single (independent and dependent) variable is simple linear regression [7]. Linear regression is equal to the dot product of input values x and the weights: $y=\vec{x}.\vec{w}$ [9]. or parameters (θ_0,θ_1) : $h_\theta(x)=\theta_0+\theta_1 x$.

6.1.1.2. Cost function or loss function

Finding appropriate parameters for best approximates the best target values over input data, needs to minimize a cost function. Cost function shows how far we are from getting correct result (t) and popular cost function is mean-square-error Illustration 3 [9]. Figure 16 shows the concept of cost function for: $h_\theta(x)=\theta_1 x$ (A), $h_\theta(x)=\theta_0+\theta_1 x$ (B) and (C) is contour of (B).

$$J=MSE=\frac{1}{n}\sum_{i=0}^{n}(y^i-t^i)^2=\frac{1}{n}\sum_{i=0}^{n}(x^i*w-t^i)^2=\frac{1}{2}\sum_{i=0}^{n}(h_\theta(x^{(i)})-y^{(i)})^2$$

Illustration 3: cost function

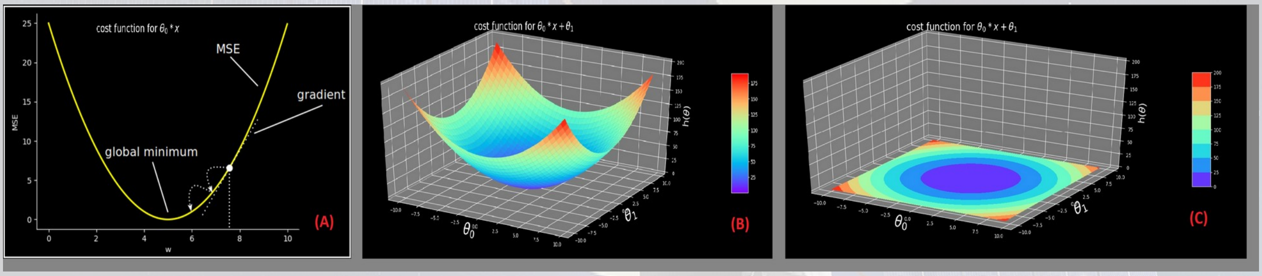


Figure 16: cost Function

6.1.1.3. Gradient-base optimization

Minimizing and maximizing some function are general optimization where x is a numerical vector or scalar ($\underset{\theta_0,\theta_1}{minimize}\ J(\theta_0,\theta_1)$). The derivative of function, detects how the function changes with small changes to x and can be used to reduce cost function[8]. to find most slope to convergence Figure 17-B, Should repeat Illustration 4 ntil convergence. The function should be convex to guaranteed, global and local minima are the same. Jacobian and Hessian are matrix to optimize functions whose input and output are vector. CG[19], BFGS[20], L-BFGS[21] are some methods of Gradient-base optimization.

$$\begin{pmatrix}\theta_0\\ \theta_1\end{pmatrix}_{old} = \begin{pmatrix}\theta_0\\ \theta_1\end{pmatrix}_{new} - \alpha * \begin{pmatrix}\frac{-\delta j}{\delta\theta_0}\\ \frac{-\delta j}{\delta\theta_1}\end{pmatrix}$$

Illustration 4: gradient descent

θ could be scalar or vector, also α is step size (learning rate). gradient descent procedure to minimize E stars from a random θ_{new}, and at each step, updates θ_{old}, in the opposite direction of gradient [10]

6.1.1.4. Learning rate

The computational cost of gradient descent for billion size training set, takes very long time and convergence process will very slowly. Full-batch gradient descent, stochastic gradient descent and mini-batch gradient descent Figure 17-A are different gradient descent [6]. Learning factor in gradient descent determines the magnitude of change to be made in parameter and generally is between 0-1. Figure 17-C shows learning rate effect on convergence of the iterative stochastic gradient descent.

19 conjugate gradient
20 Broyden–Fletcher–Goldfarb–Shanno algorithm
21 Limited BFGS

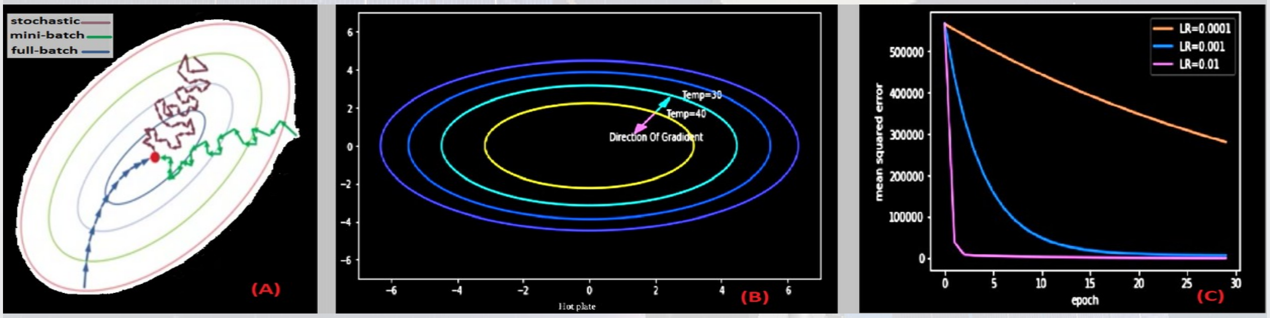


Figure 17: different type of gradient descent(A), direction for convergence(B), effect of learning rate for MSE(C)

6.1.1.5. Applied linear regression on water spectrometer's spectral signature

Linear regression applied to the water spectral signature and has showed Figure 18. optimum θ_0 and θ_1 are 0.213, -0.0021 for 0 to 14 µm and 0.222, -0039 for 0.35 to 2.5 µm.

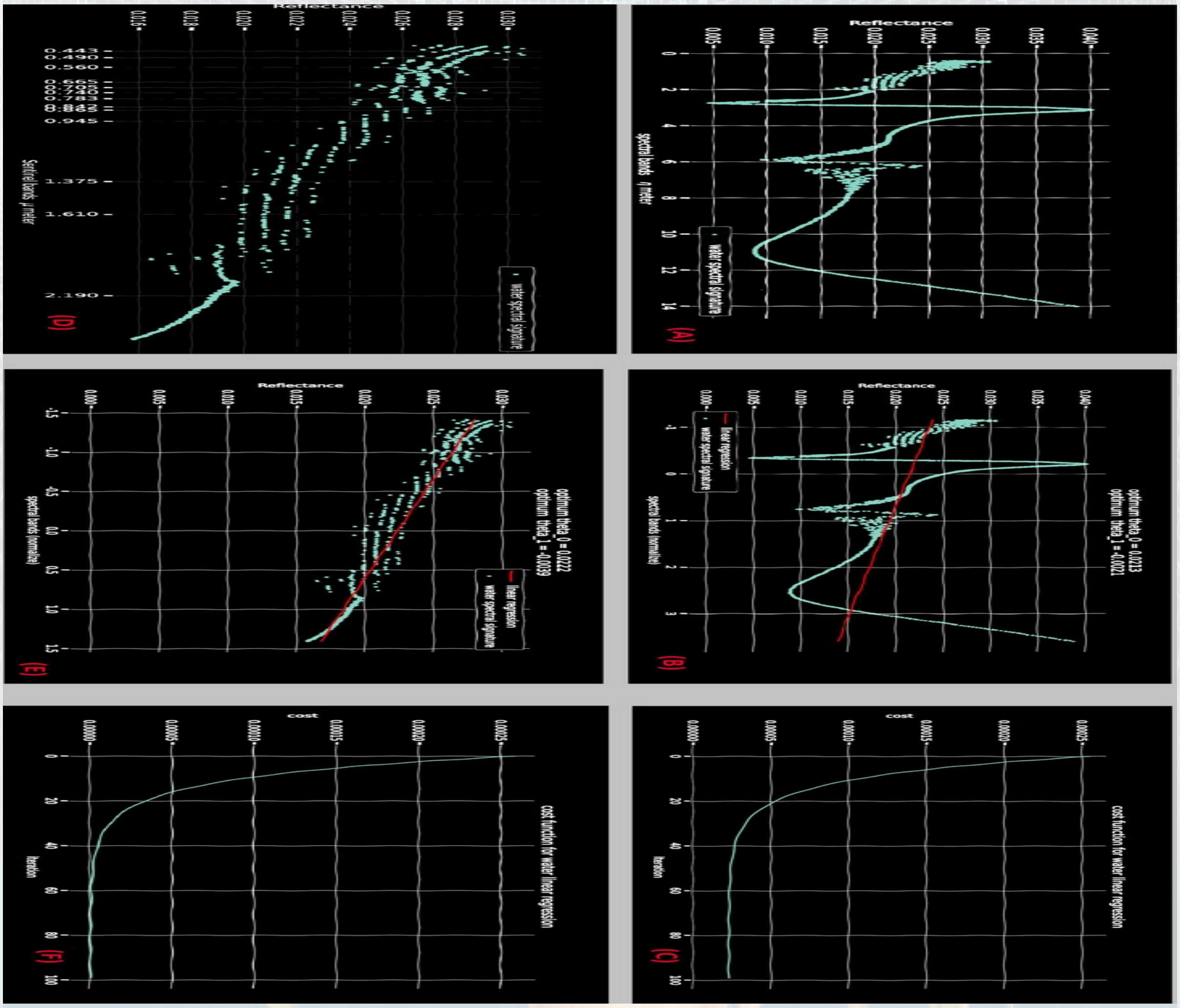


Figure 18: water spectral signature: first row (0-14 micro m),

second row (0.35-2.5 micro m), third column: cost function

6.1.2. Multivariate linear regression and Polynomial regression

A single dependent variable work with single dependence or independence variable. If each observation is a vector composed or multiple explanatory variable then use multiple regression or multivariate regression [8]. Assumed to be written as a linear function with weighted sum of several input variables [10]. Rest of the multivariate linear regression progress is same as simple linear regression, as mentioned Illustration 5.

$$h_\theta(x)=\theta_0+\theta_1 x_1+\theta_2 x_2+\theta_3 x_3+\ldots+\theta_n x_n$$

Illustration 5: multivariate regression

Polynomial regression is special case of multivariate regression that modeled to fit or map nonlinear relationships between dependent and independent variables [8]. to the n^{th} degree of independent variables of order k [10].

$$h_\theta(x)=\theta_0+\theta_1 x_1^k+\theta_2 x_2^k+\theta_3 x_3^k+\ldots+\theta_n x_n^k$$

Illustration 6: multivariate regression of order k

The most common way To avoid over-fit in high order of k of test data is regularization [6]. Weight decay, weight-sharing, early stopping and drop-out are the common regularization techniques [8]. Weight decay works by adding an additional penalty term (ƛ) to the value of the loss function [9]. The most popular of penalty terms are L^1 (lasso), L^2 (ridge) and Elastic net. The parameter θ and hyper-parameter ƛ should define by the operator. Occam's razor (or Ockham's razor) is a principle from philosophy. Suppose there exist two explanations for an occurrence. In this case the one that requires the smallest number of assumptions is usually correct[22].

$$J(\theta)=\frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^i)-y^i)^2+\lambda\sum_{i=1}^{n}\theta_j^2=\frac{1}{2}(x_\theta-y)^{\mathrm{T}}(x_\theta-y)+\frac{1}{2}\lambda\,\theta^\tau\theta$$

Illustration 7: Regularization

6.1.3. Locally weighted Regression ss (smoothing scattering)

Locally weighted regression Illustration 9 is a non-parametric method and use only training local data to performs a regression around a point of interest. A variant of LOESS and LOWESS [6] model is linear regression on points in the data set, weighted by a kernel centered Figure 19 at x [11]. where τ is smoothing parameters (width band) Illustration 8. When the kernel includes as many training points as can be accommodated, the estimator variance is minimize[23].

$$w^{(i)}=\exp\left(\frac{(x^{(i)}-x)^2}{2\tau^2}\right)$$

Illustration 8: smooth weight

22 https://simple.wikipedia.org/wiki/Occam%27s_razor
23 https://www.cs.cmu.edu/afs/cs/project/jair/pub/volume4/cohn96a-html/node7.html

$$j(\theta)=\sum_{i=1}^{m} w^{i}(y^{(i)}-h_{\theta}(x^{(i)}))^{2}$$

Illustration 9: locally linear regression

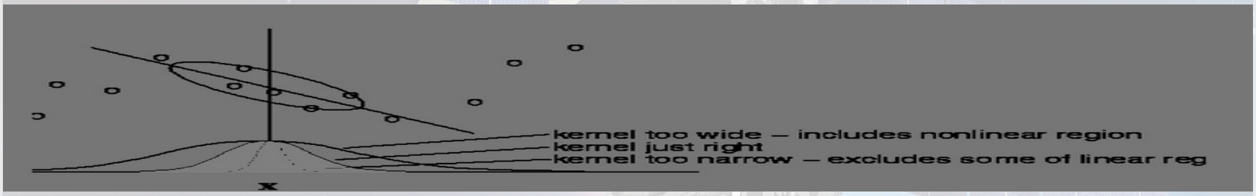


Figure 19: locally linear regression

6.1.4. applied Locally weighted Regression

Locally weighted applied with three different parameter f and has showed Figure 20. first column is for f = 2. / 3., second column is for f = 0.03 and finally 3th column is for f = 0.01. as seen this method doesn't have good property for spectral signature (0 to 14 µm) in f = 0.01 and faced singular matrix error.

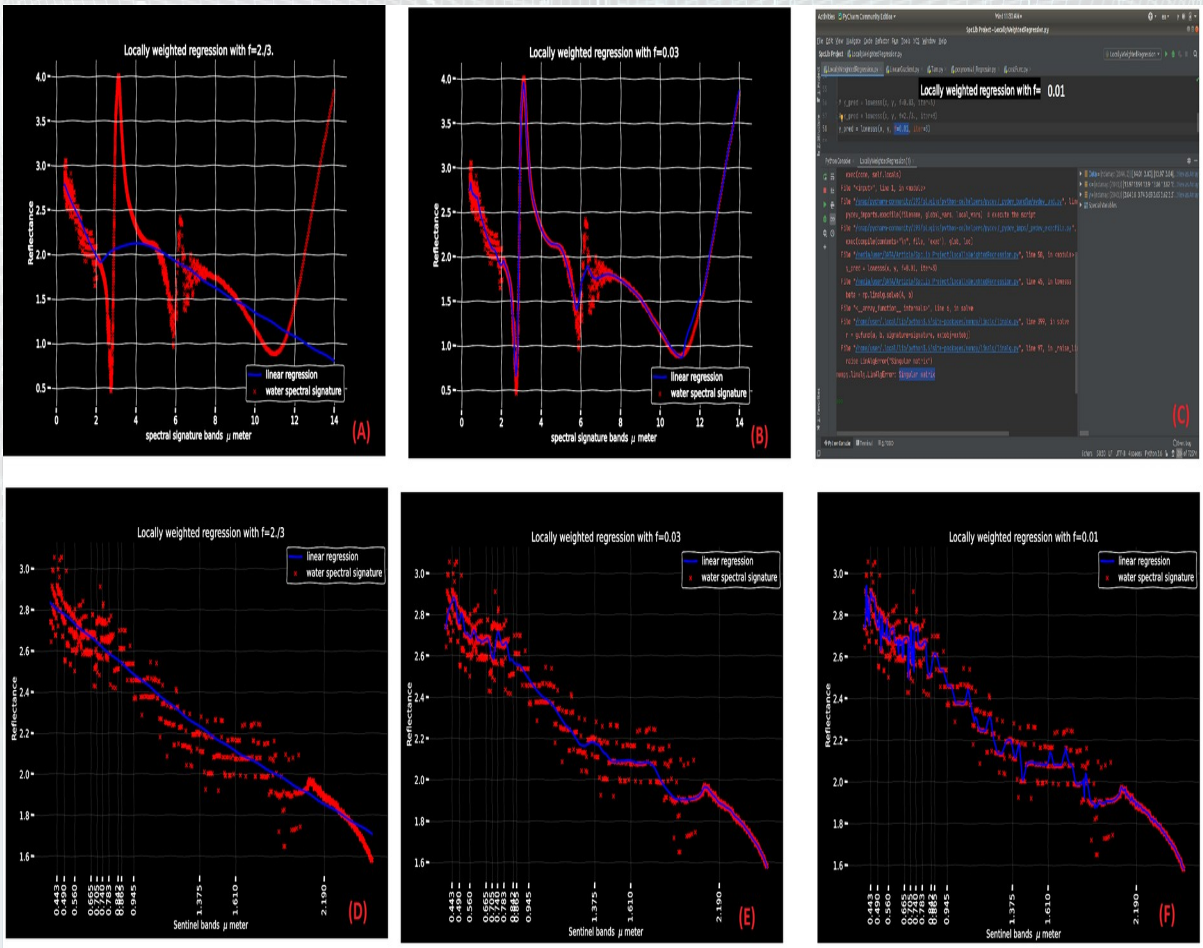


Figure 20: applied Locally weighted Regression

first row f=2./3. Second row: f = 0.03 third row: f=0.01

6.2. Find optimum θ for object detection

6.2.1. Normal Equation

The numeric output assumed to be written as a linear function in multivariate linear regression Illustration 10 [10]. θ calculated by the pseudo-inverse of the input (pixel's spectral signature) multiplied by the output (NDVI Re-Class). This method probably couldn't perform well for big data.

$$\theta = (X^T X)^{-1} X^T y$$

Illustration 10: Normal Equation

6.2.2. Applied Normal Equation

Normal Equation only has applied to case study small data and re-class in two ways. First way is rounded export value Figure 21 and second way is performed threshold Figure 22 according Table 4. First way's accuracy is 92% and second way's accuracy is 91%. the θ

Table 4: Normal Equation - Re-class export value

Threshold = (max – min) / 3		
Value <= Threshold	Water	1
Threshold < Value < 2 * Threshold	bare land	2
Value >= 2 * Threshold	agriculture	3

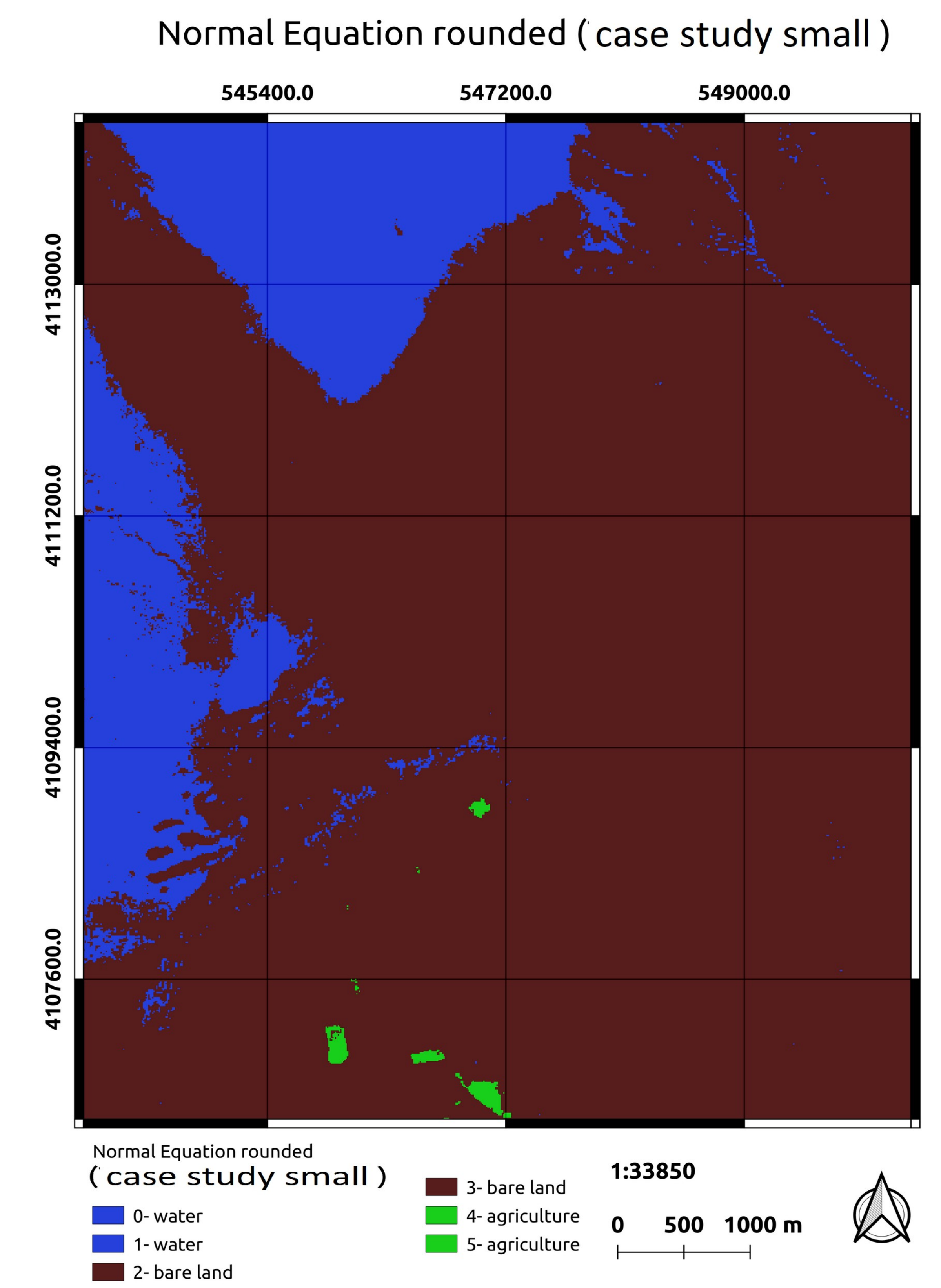


Figure 21: Normal Equation rounded value (case study small)

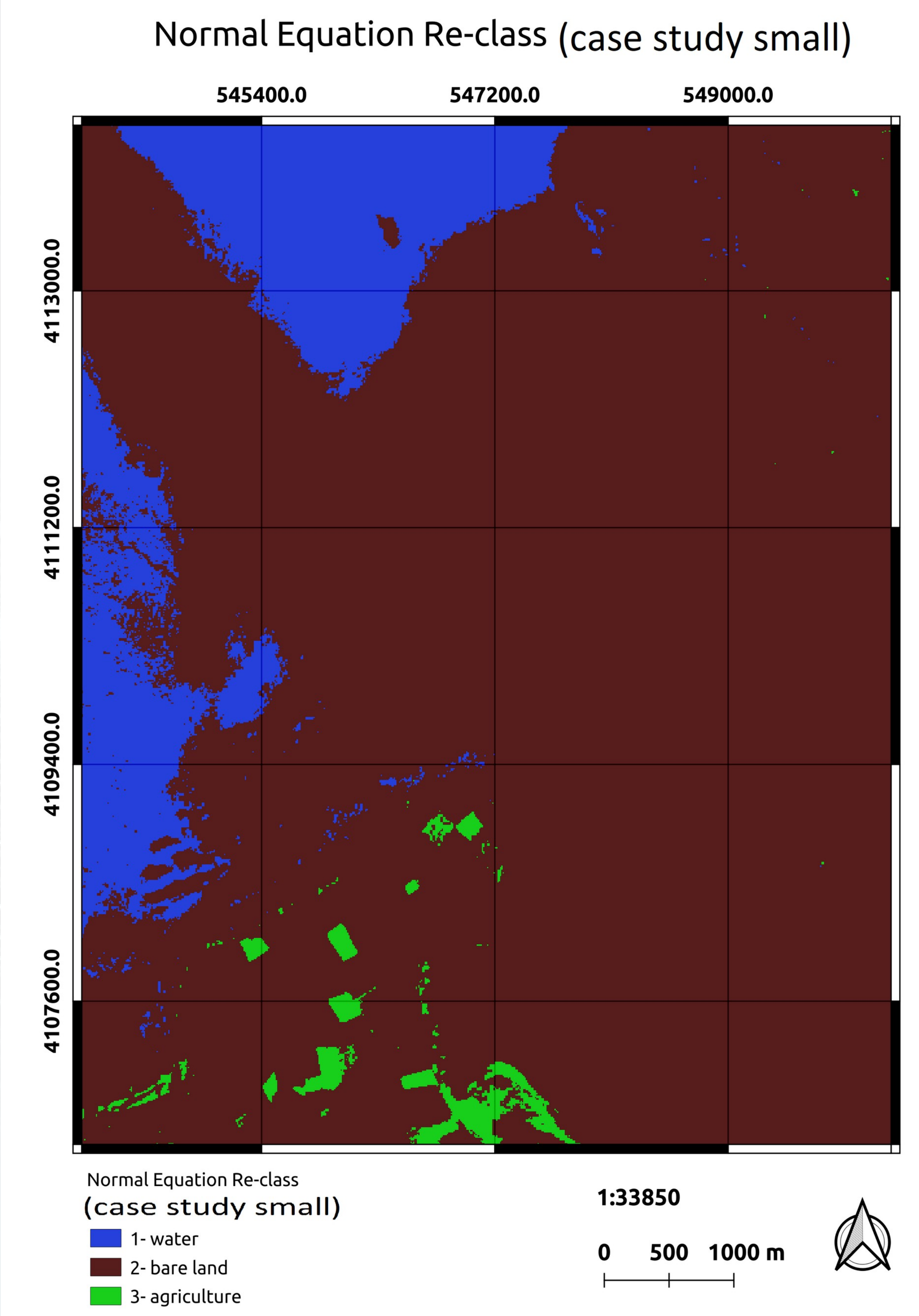


Figure 22: Normal Equation Re-class (case study small)

6.3. Neural Network

Over time, we’ve stopped trying to emulate neural network inspired by the brain. Instead we focused on finding the correct configurations for specific classification. Linear regression is a special case of neural network and is a single neuron with identify activation function.

Neural network has two main characteristics. 1. architecture include connections, feed-forward, recurrent, multi or single layer and 2. common ways to training such as gradient descent and back-propagation. [9]

6.3.1. Logistic Regression, soft-max or cross-entropy

Logistic regression Illustration 11 uses sigmoid activation function [10]. Soft-max function Illustration 13 generalize two classes classification problem for multiple classes [10] The soft-max function squashes the output of each unit to be between 0 and 1 [9]. Sigmoid performance has showed in Table 5. o find optimum θ, must minimize cost function Illustration 12. Linear regression is a special case of neural network and is a single neuron with identify activation function [9].

$$p(\theta|x) = sigmoid(\theta^T * x) = \frac{1}{1 + \exp(-\theta^T + \theta_0)}$$

Illustration 11: Logistic Regression activation function

$$J(\theta) = -y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))$$

Illustration 12: Logistic Regression cost function

$$p(Y = k \vee X = x^{(i)}) = \frac{e^{s_k}}{\sum_{j=1}^{c} e^{s_j}}$$

Illustration 13: soft-max activation function

$$-\log\left(\frac{e^{s} * y^{(i)}}{\sum_j e^{s_j}}\right)$$

Illustration 14: soft-max cost function

Table 5: sigmoid performance

Different Z value					
$\frac{1}{1+e^{-z}}$	Z = +∞	$\frac{1}{1+e^{-\infty}}$	$\frac{1}{1+0}$	$\frac{1}{1}$	1
$\frac{1}{1+e^{-z}}$	Z = - ∞	$\frac{1}{1+e^{+\infty}}$	$\frac{1}{1+\infty}$	$\frac{1}{\infty}$	0

6.3.1.1. applied Logistic regression

Case study small data split to train and test. Logistic regression applied to NDVI case study small data Figure 24 and has faced warning Error : n_iter_i = _check_optimize_result. Then the model has acquired applied on case study big Figure 23. Accuracy for case study small was 97.53 % in 14.46 second and for case study big was 97.92.

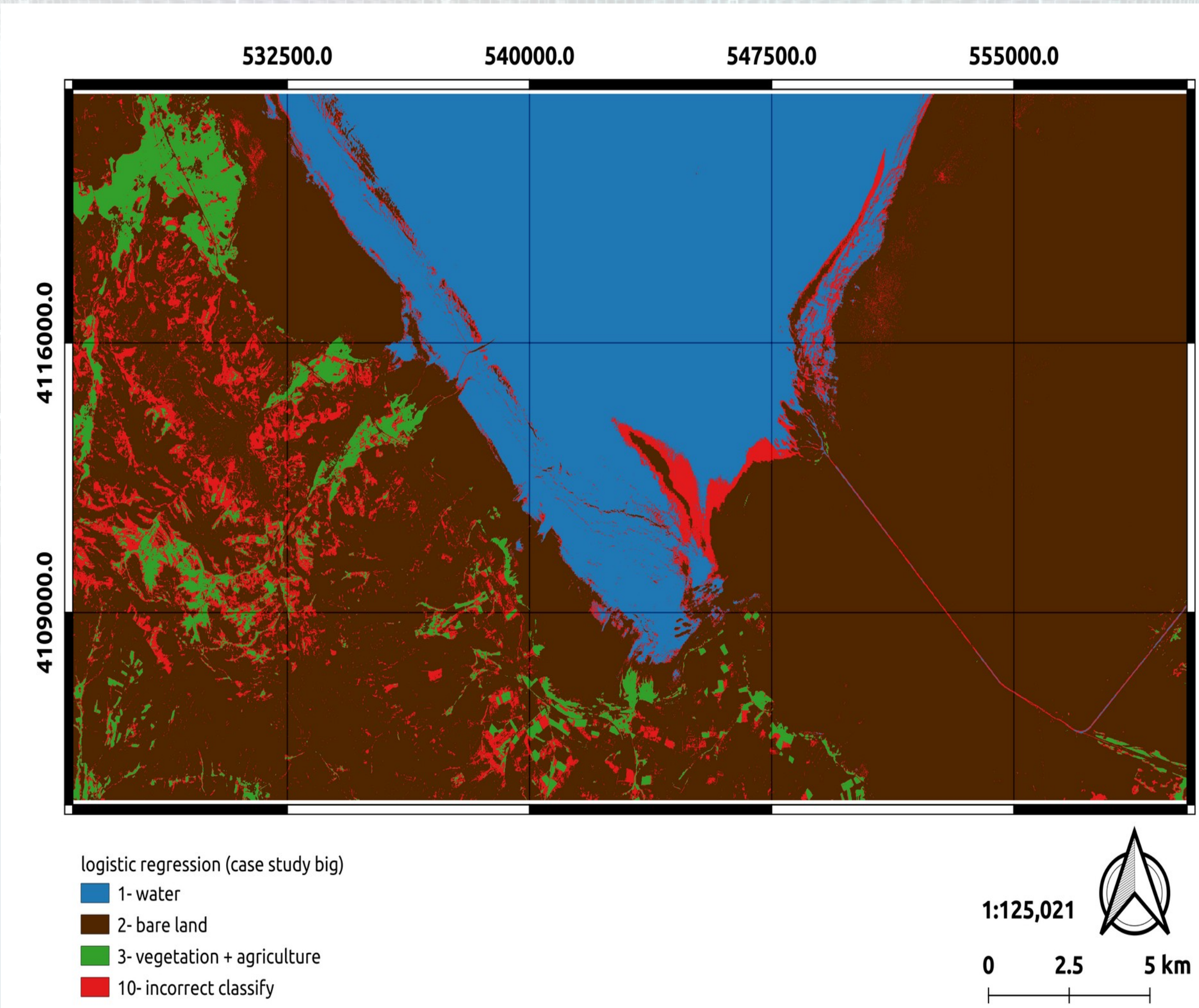


Figure 23: logistic regression (case study big)

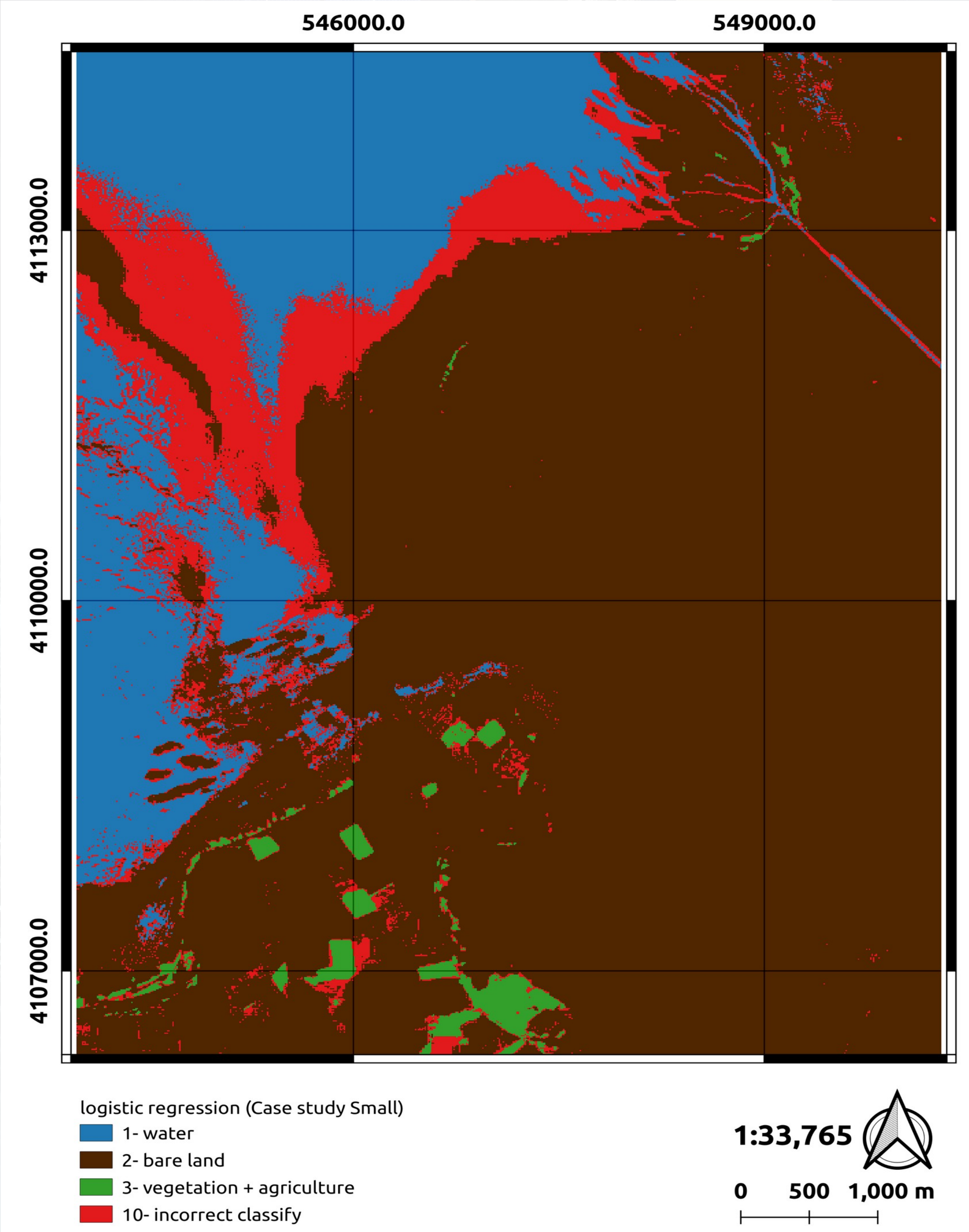


Figure 24: logistic regression (case study small)

6.3.2. NN – single layer

As schematically represented as Figure 25, linear neuron is the most basic of a deep neural network [8] and all parameter has discussed in part 6.1.1.1.

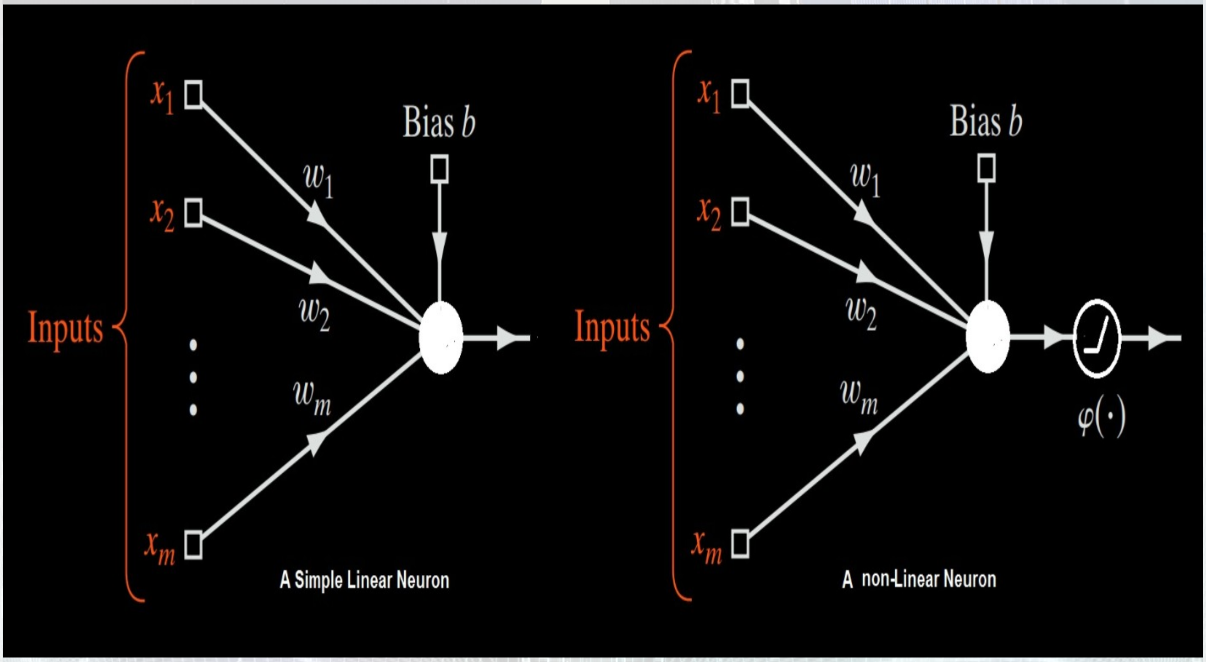
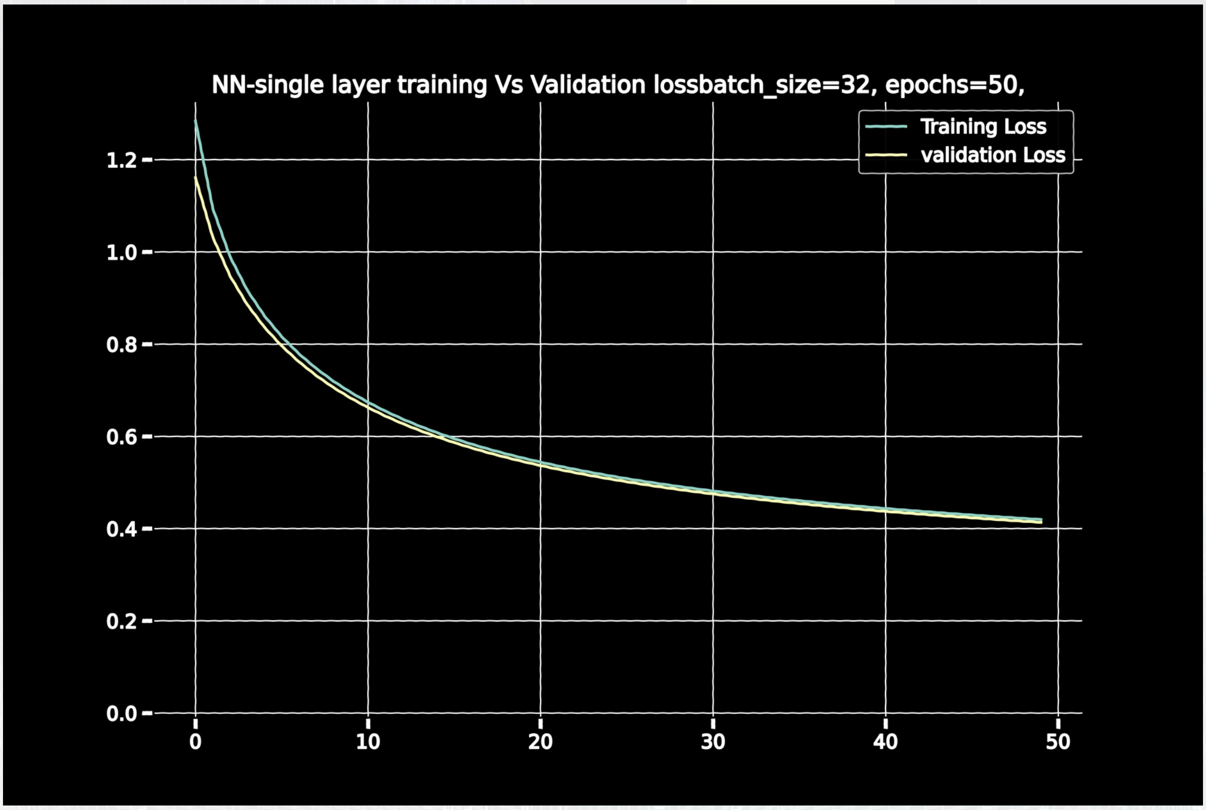
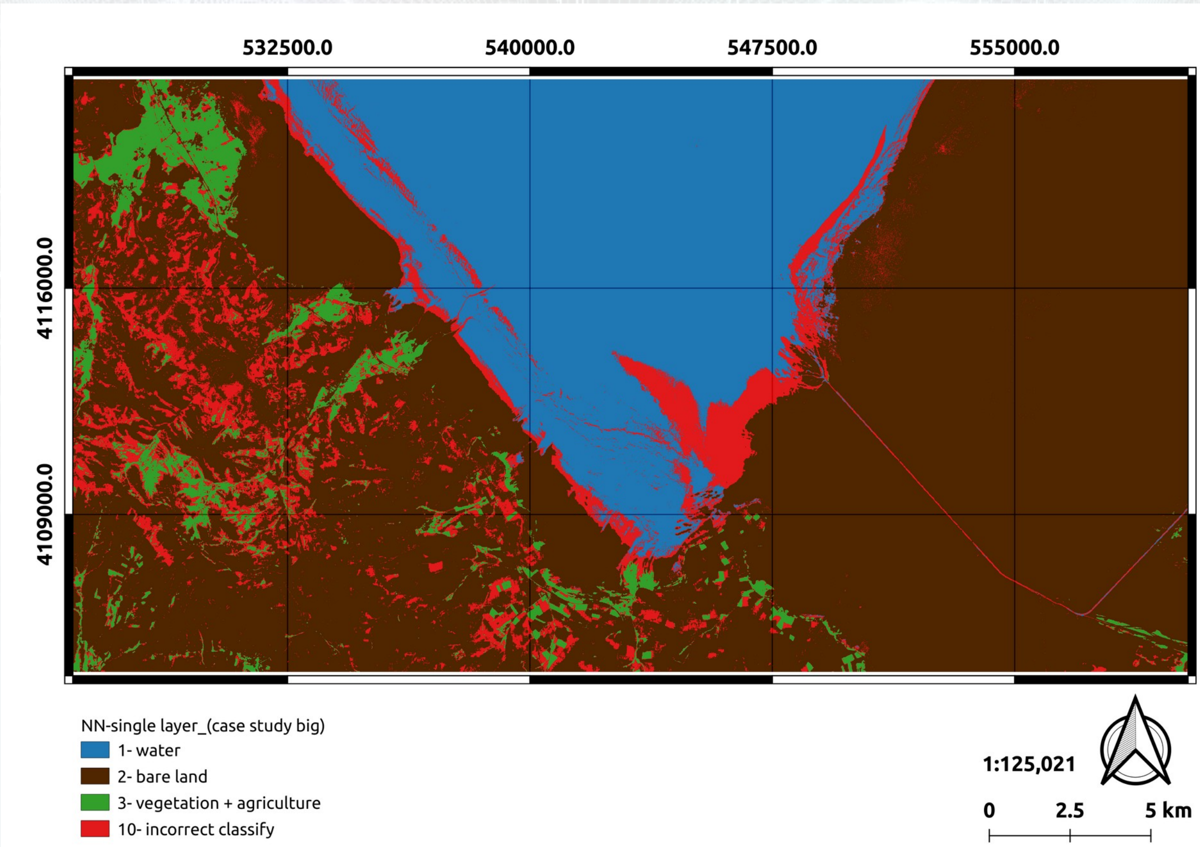
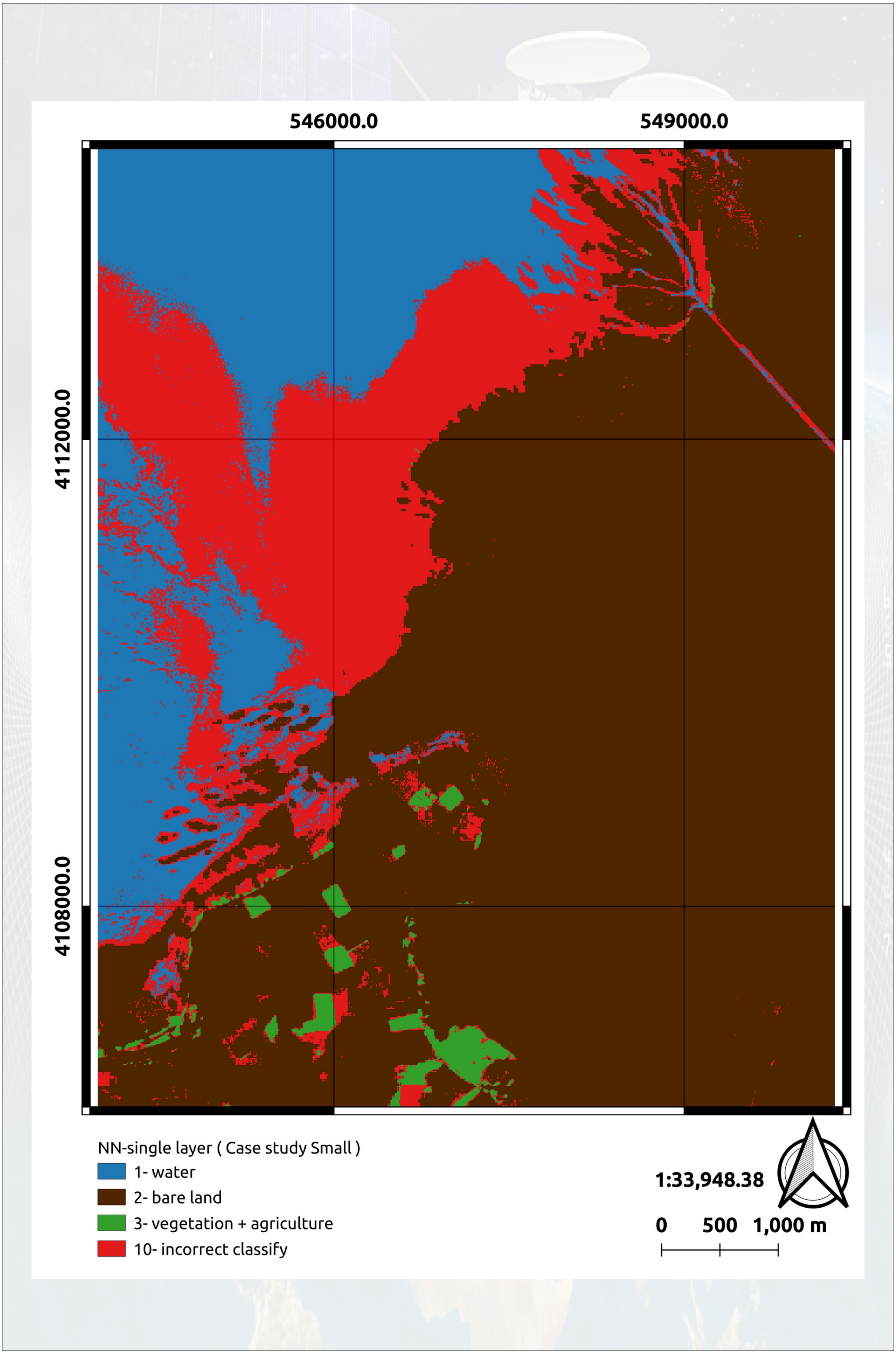


Figure 25: Schematic representation of a simple linear neuron and a simple non-linear neuron

```
Model: "sequential"
_________________________________________________________________
Layer (type)              Output Shape           Param #
=================================================================
dense (Dense)              (None, 4)             52
=================================================================
Total params: 52
Trainable params: 52
Non-trainable params: 0
_________________________________________________________________
None
Train on 15774 samples, validate on 1753 samples
```


NN-single layer training Vs Validation lossbatch_size=32, epochs=50,
Training Loss
validation Loss
1.2
1.0
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
50


532500.0
540000.0
547500.0
555000.0
4116000.0
4109000.0
NN-single layer_(case study big)
1- water
2- bare land
3- vegetation + agriculture
10- incorrect classify
1:125,021
0
2.5
5 km


546000.0
549000.0
4112000.0
4108000.0
NN-single layer (Case study Small)
1- water
2- bare land
3- vegetation + agriculture
10- incorrect classify
1:33,948.38
0
500
1,000 m

6.4. Multi-layer neural networks

1-layer neural nets can only classify linearly separable classes. The input layers of multi-layer neural networks Figure 26 has k input neurons and m hidden layers and each connection has its own weights [9].

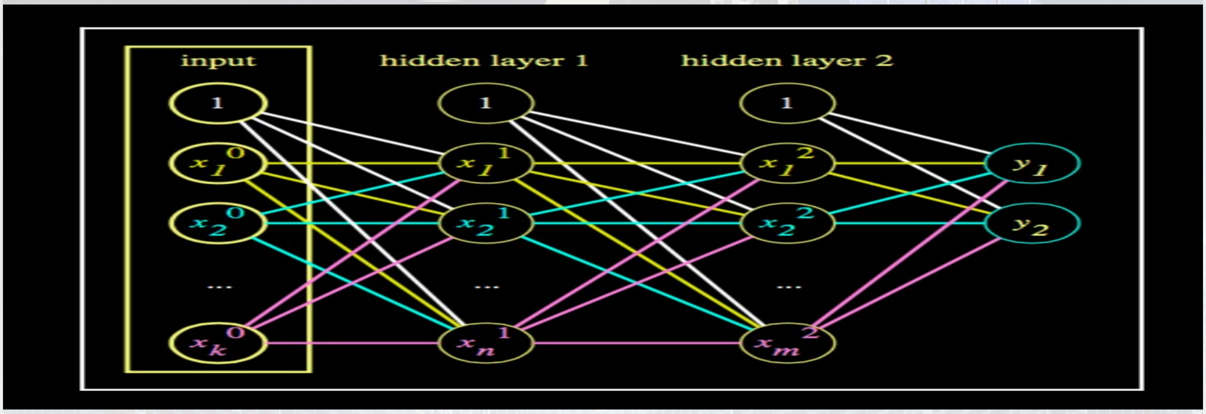
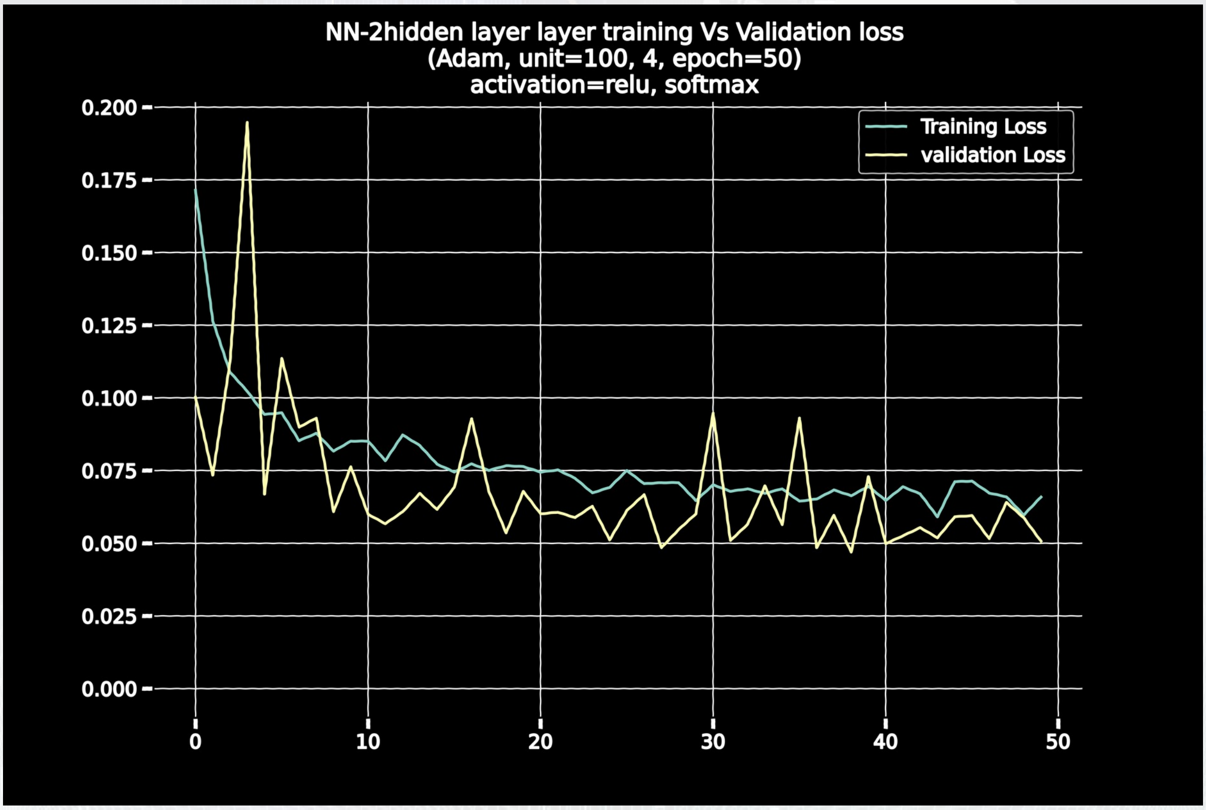


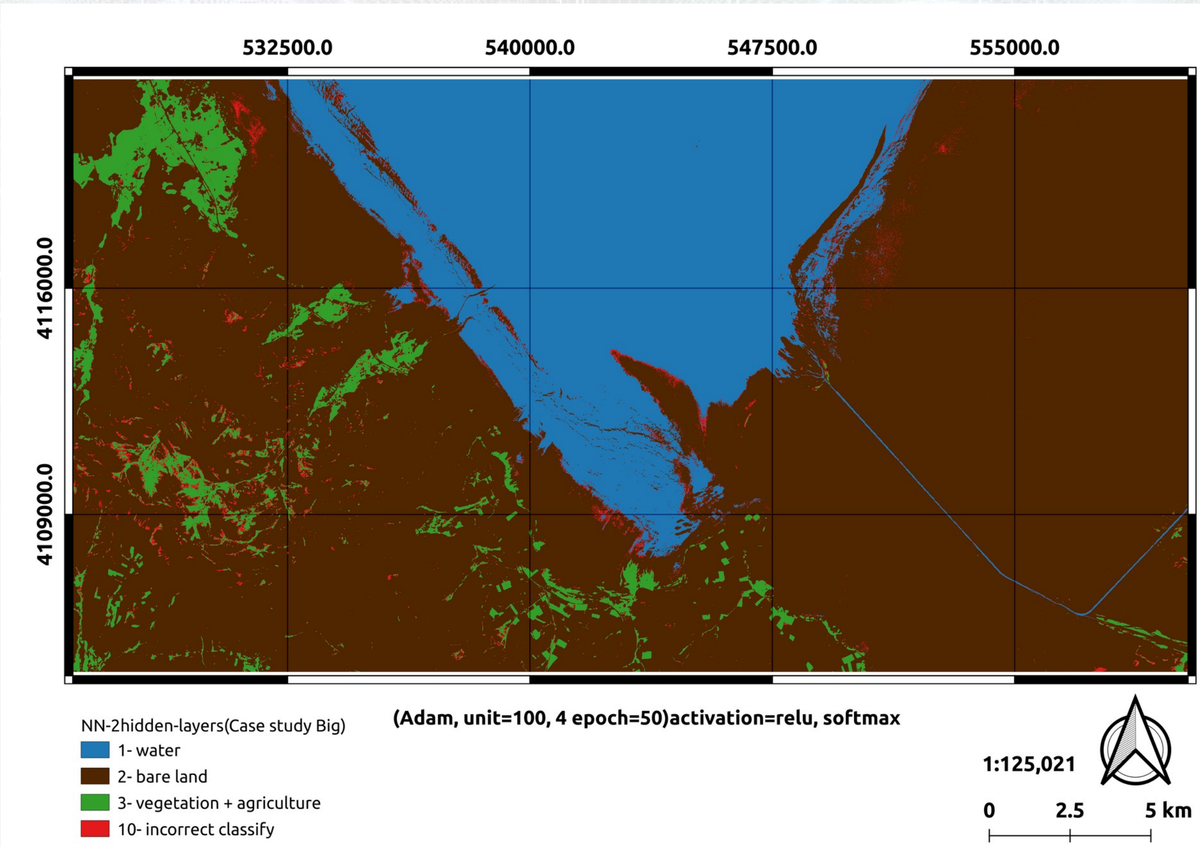
Figure 26: Multi-layer sequential network

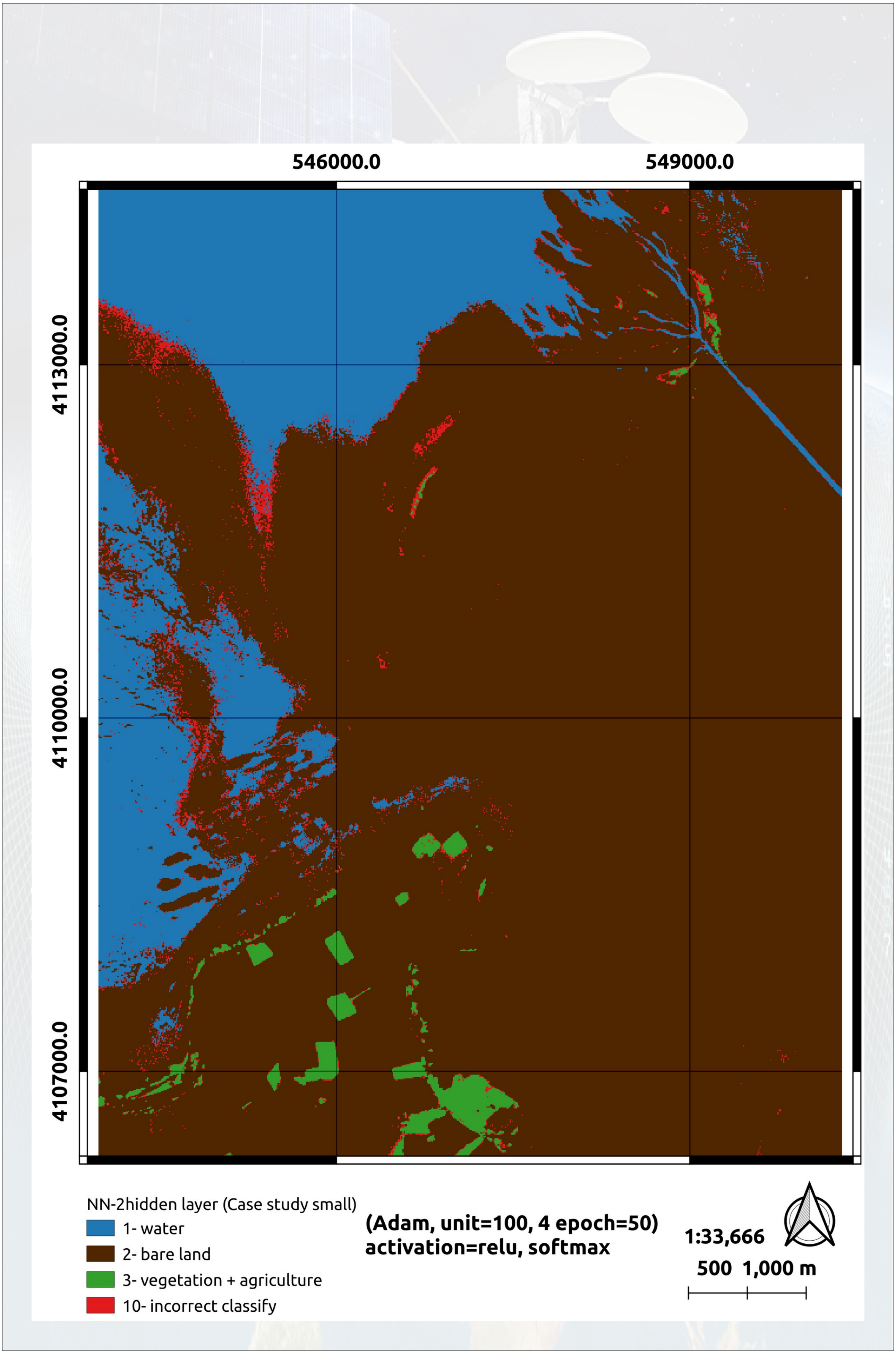
A single layered non linear explained by looking at XOR problem. In the XOR problem, we want a neural network model to learn XOR function. One way to solve XOR problem is to use a different representation of the input such that the linear model is able to find a solution. By adding non-linear hidden layer to the network, it could be achieved [8].

6.4.1. NN 2 hidden layer

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 100)               1300
_________________________________________________________________
dense_1 (Dense)              (None, 100)               10100
_________________________________________________________________
dense_2 (Dense)              (None, 4)                 404
=================================================================
Total params: 11,804     Trainable params: 11,804     Non-trainable params: 0
_________________________________________________________________
None
Train on 15774 samples, validate on 1753 samples
```


NN-2hidden layer layer training Vs Validation loss
(Adam, unit=100, 4, epoch=50)
activation=relu, softmax
Training Loss
validation Loss
0.200
0.175
0.150
0.125
0.100
0.075
0.050
0.025
0.000
0
10
20
30
40
50


532500.0
540000.0
547500.0
555000.0
4116000.0
4109000.0
NN-2hidden-layers(Case study Big)
1- water
2- bare land
3- vegetation + agriculture
10- incorrect classify
(Adam, unit=100, 4 epoch=50)activation=relu, softmax
1:125,021
0
2.5
5 km


546000.0
549000.0
4113000.0
4110000.0
4107000.0
NN-2hidden layer (Case study small)
1- water
2- bare land
3- vegetation + agriculture
10- incorrect classify
(Adam, unit=100, 4 epoch=50)
activation=relu, softmax
1:33,666
500 1,000 m

6.4.2. NN 2 hidden layer (Drop-out)

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 100)	1300
dropout (Dropout)	(None, 100)	0
dense_1 (Dense)	(None, 100)	10100
dropout_1 (Dropout)	(None, 100)	0
dense_2 (Dense)	(None, 4)	404

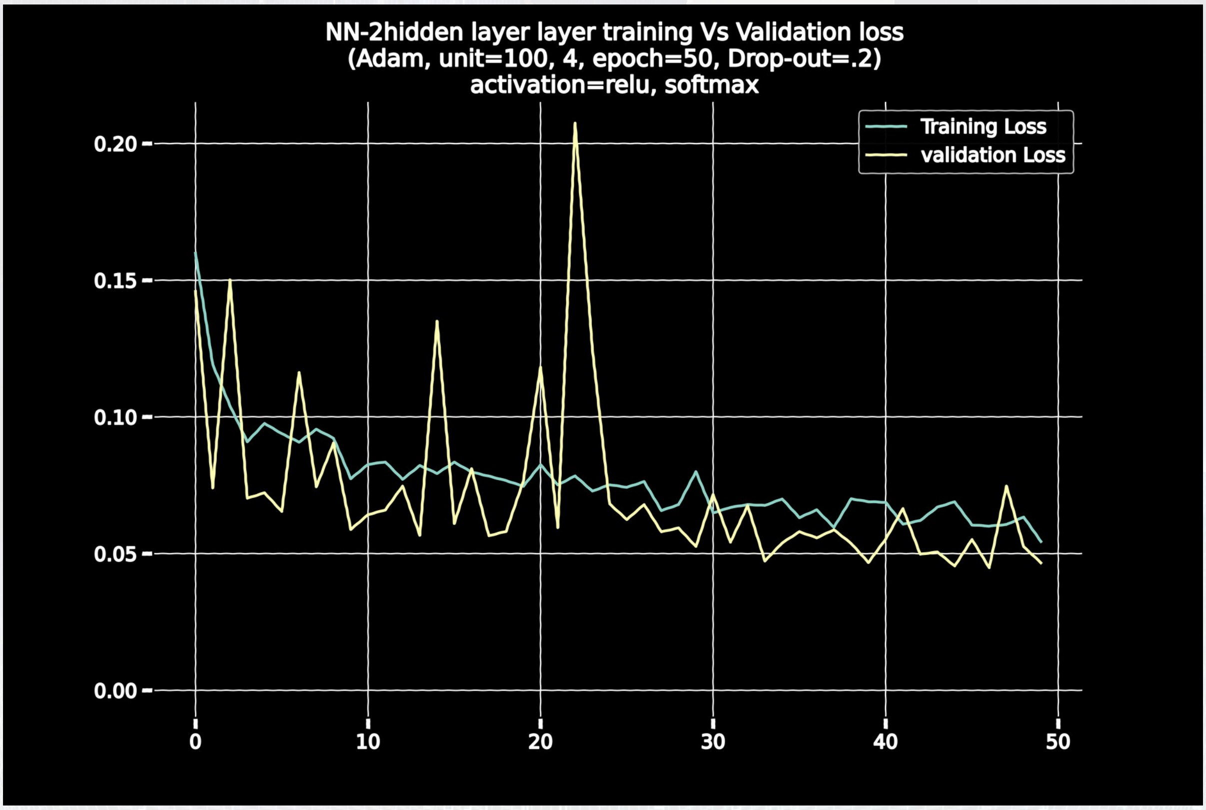
Total params: 11,804

Trainable params: 11,804

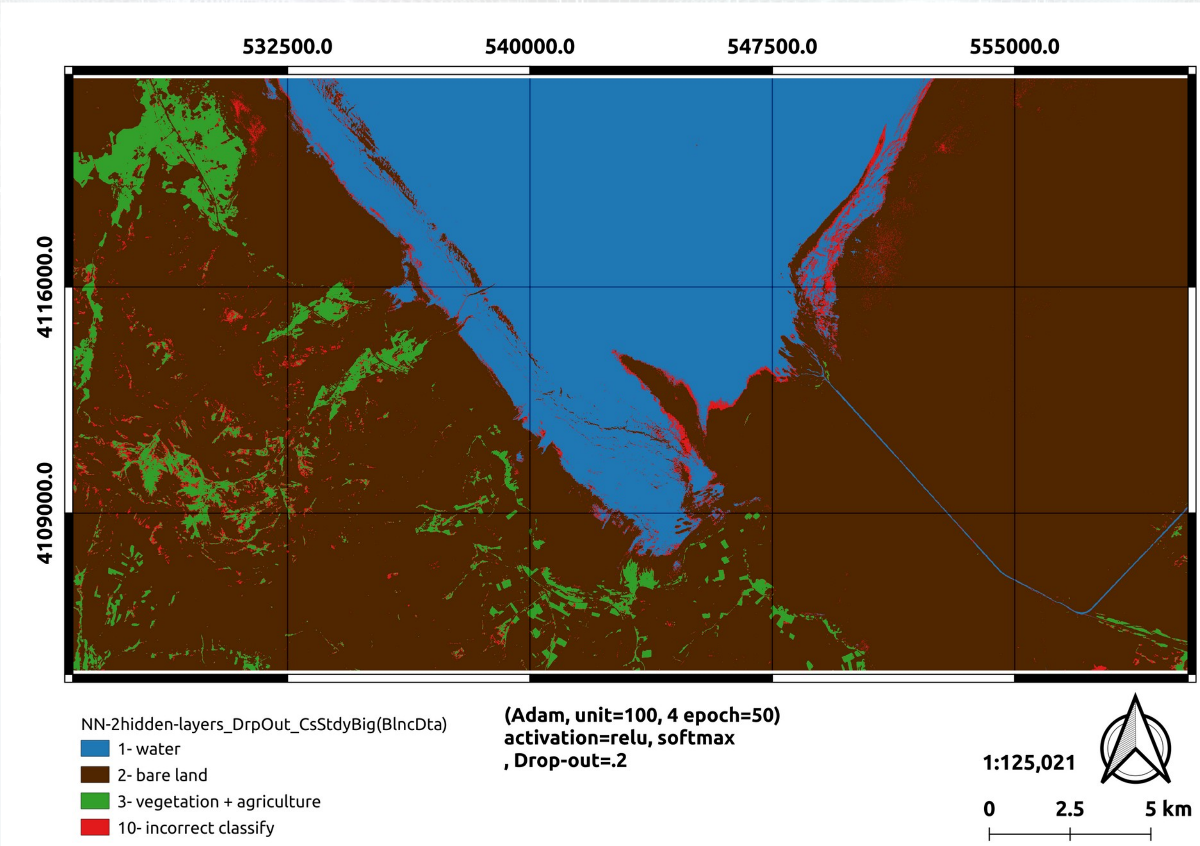
Non-trainable params: 0

None

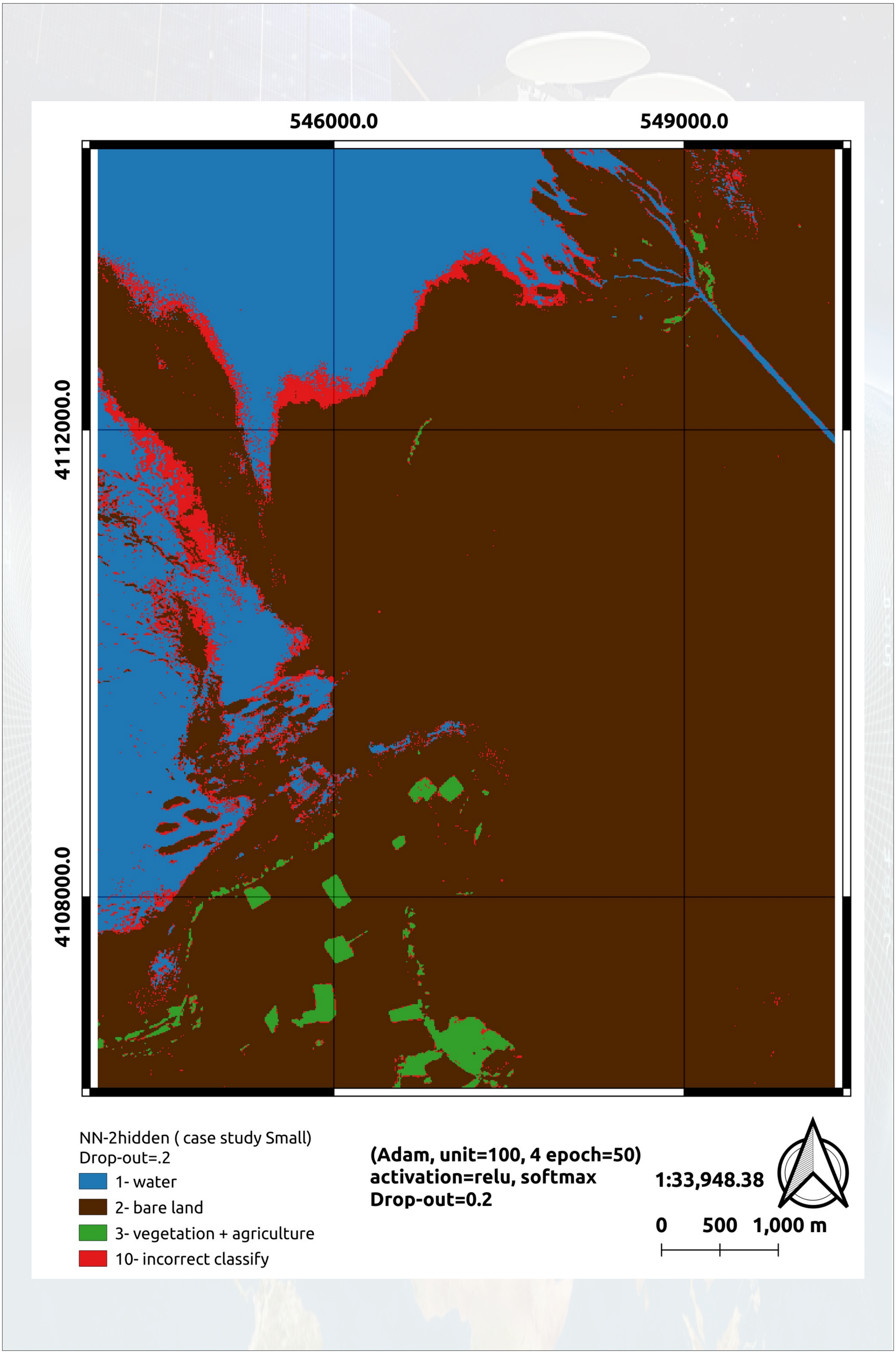
Train on 15774 samples, validate on 1753 samples



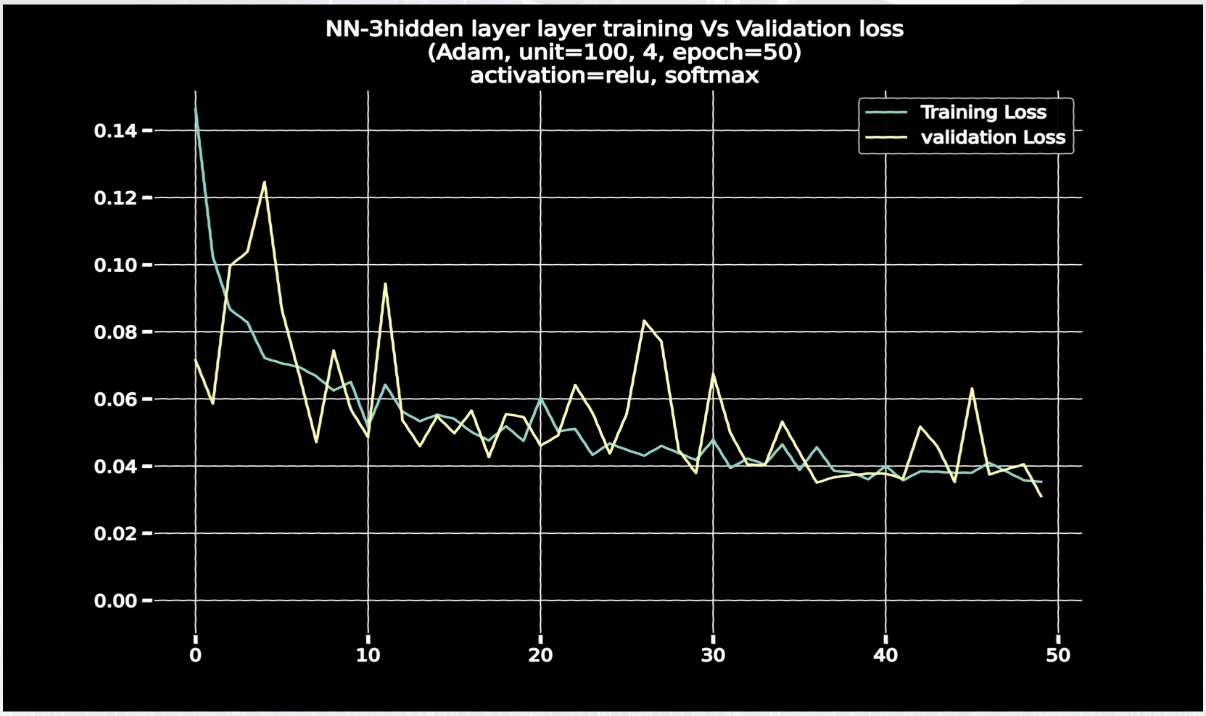
NN-2hidden layer layer training Vs Validation loss
(Adam, unit=100, 4, epoch=50, Drop-out=.2)
activation=relu, softmax
Training Loss
validation Loss
0.20
0.15
0.10
0.05
0.00
0
10
20
30
40
50

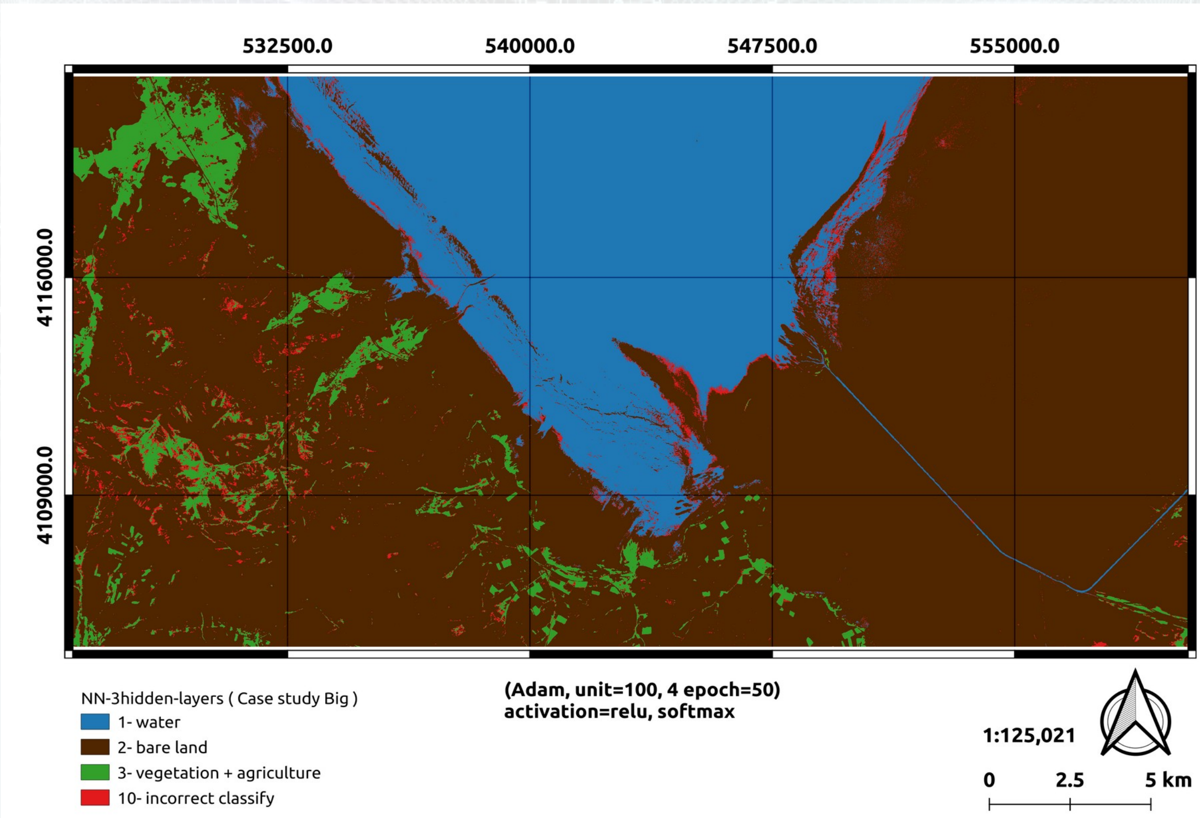


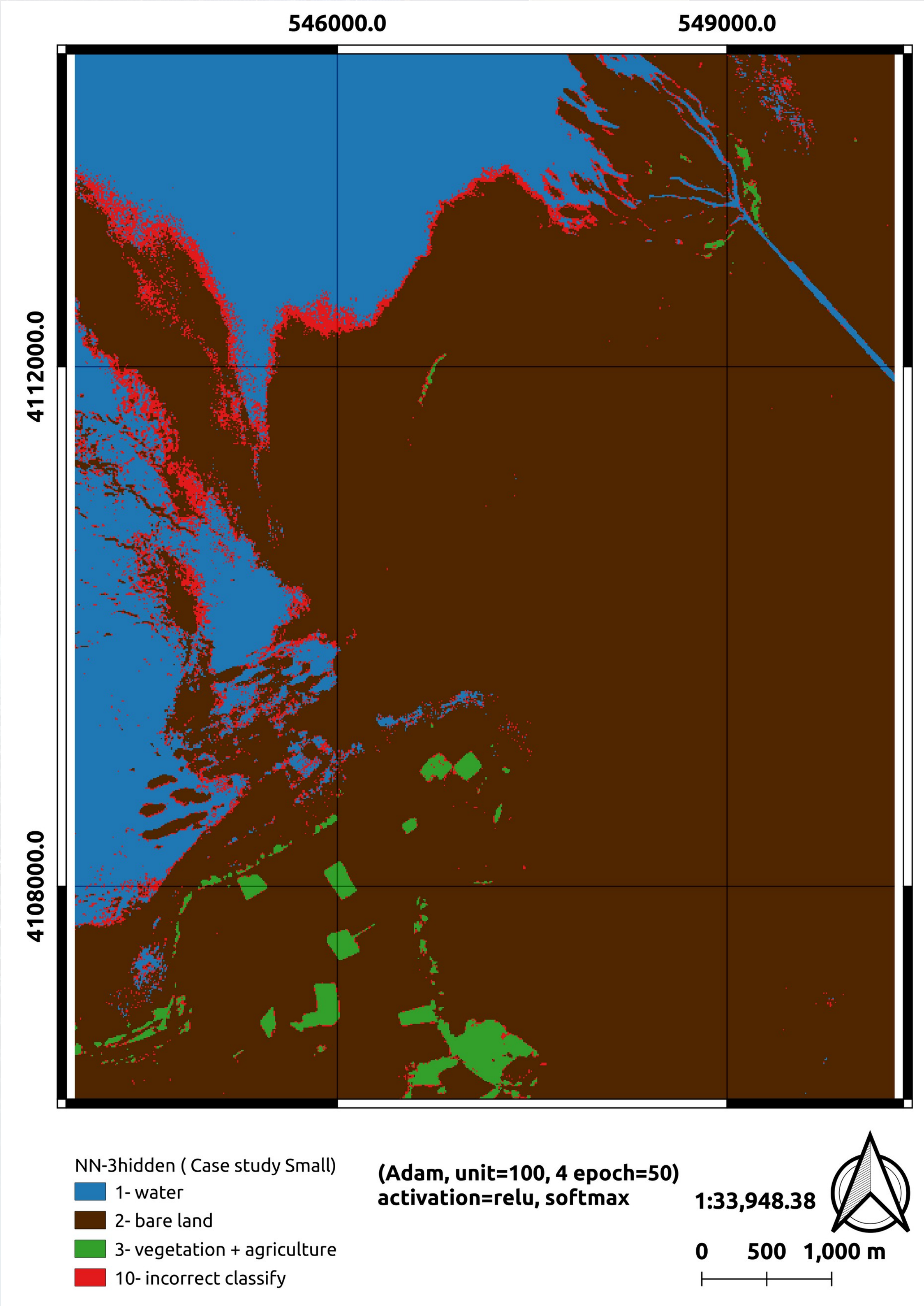
532500.0
540000.0
547500.0
555000.0
4116000.0
4109000.0
NN-2hidden-layers_DrpOut_CsStdyBig(BlncDta)
1- water
2- bare land
3- vegetation + agriculture
10- incorrect classify
(Adam, unit=100, 4 epoch=50)
activation=relu, softmax
, Drop-out=.2
1:125,021
0
2.5
5 km


546000.0
549000.0
4112000.0
4108000.0
NN-2hidden (case study Small)
Drop-out=.2
1- water
2- bare land
3- vegetation + agriculture
10- incorrect classify
(Adam, unit=100, 4 epoch=50)
activation=relu, softmax
Drop-out=0.2
1:33,948.38
0
500
1,000 m

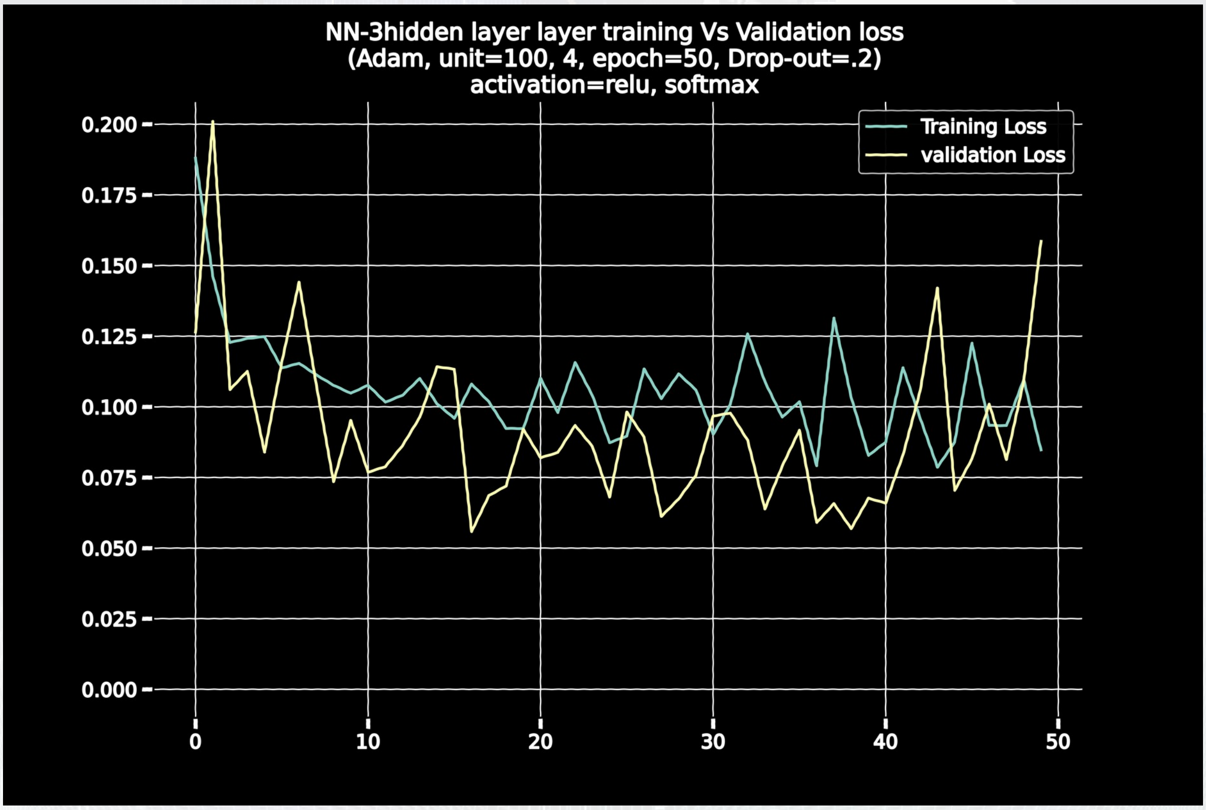
6.4.3. NN 3 hidden layer

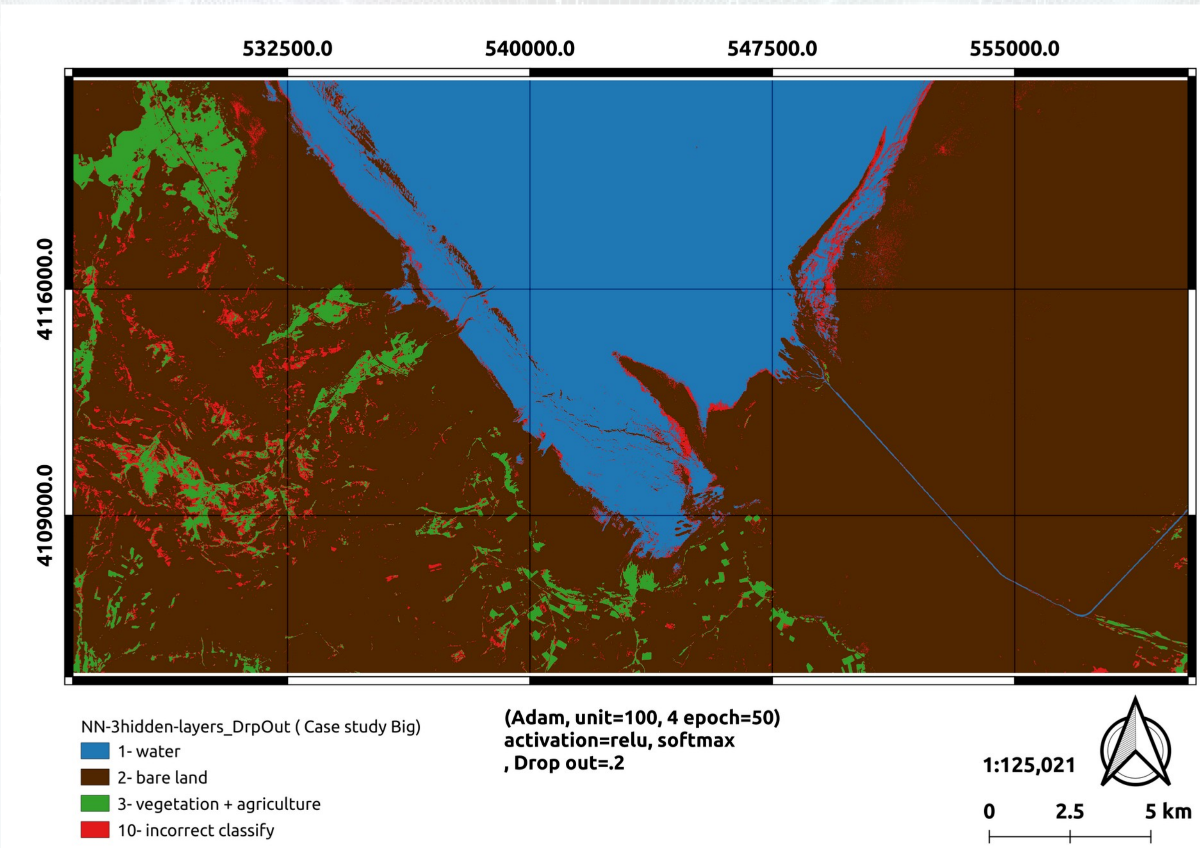


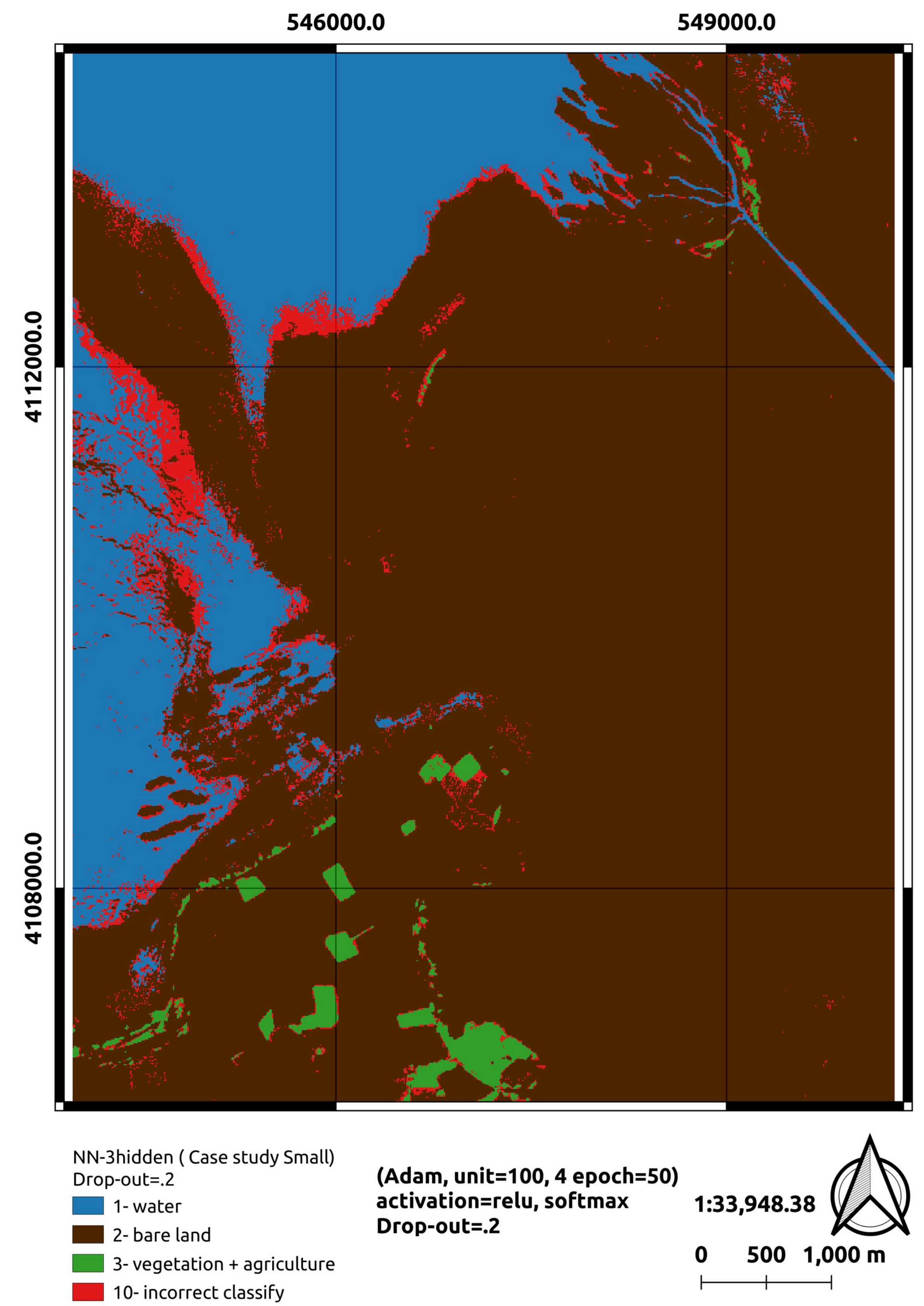



546000.0
549000.0
4112000.0
4108000.0
NN-3hidden (Case study Small)
1- water
2- bare land
3- vegetation + agriculture
10- incorrect classify
(Adam, unit=100, 4 epoch=50)
activation=relu, softmax
1:33,948.38
0 500 1,000 m

6.4.4. NN 3 hidden layer drop-out






546000.0
549000.0
4112000.0
4108000.0
NN-3hidden (Case study Small)
Drop-out=.2
1- water
2- bare land
3- vegetation + agriculture
10- incorrect classify
(Adam, unit=100, 4 epoch=50)
activation=relu, softmax
Drop-out=.2
1:33,948.38
0
500
1,000 m

6.5. CNN

To be continued …

Table 6: summary to find optimum θ for balance data

Balance Data {1.0: 7790, 2.0: 7790, 3.0: 7790} ---------- Case study small shape (861, 631) ----------- Case study Big shape (1831, 4564)	Min & max export values -------- Parameter or hyper-parameter	Unique counter		Accuracy (%)	Duration
		Model Export (predict (yTst))	Correct & Incorrect classify pixels (predict != yTst) *10.0 : incorrect		
Trained Normal Equation Balance data	-	{0.0: 5, 1.0: 7945, 2.0: 8583, 3.0: 6642, 4.0: 195}	-	87.75 (%)	0.06 sec
Apply Normal Equation <u>Case study small</u>	-0.23 to 4.42 ------------ Rounded value	{0.0: 94, 1.0: 136097, 2.0: 381640, 3.0: 25229, 4.0: 231}	{1.0: 107084, 2.0: 374512, 3.0: 6269, 10.0: 55426}	89.80 (%)	0.37 sec
Apply Normal Equation <u>Case study Big</u>	-0.62 to 4.58 ---------- Rounded value	{-1.0: 1, 0.0: 104019, 1.0: 1454015, 2.0: 4845359, 3.0: 1935119, 4.0: 18169, 5.0: 2}	{1.0: 1370923, 2.0: 4760012, 3.0: 409712, 10.0: 1816037}	78.27 (%)	2.20 sec
Apply Normal Equation	Re-Class	{1.0: 151023, 2.0: 392268}	{1.0: 108752, 2.0: 380235,	90.00 (%)	0.08 sec

Case study small			10.0: 54304}		
Apply Normal Equation Case study Big	Re-Class	{1.0: 2000470, 2.0: 6356214}	{1.0: 1517667, 2.0: 5888165, 10.0: 950852}	88.62 (%)	1.16 sec
Trained Logistic regression Balance data	extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG	{1.0: 2184, 2.0: 1708, 3.0: 1951}	{1.0: 1954, 2.0: 1661, 3.0: 1929, 10.0: 299}	94.8 (%)	1.05 sec
Apply Logistic regression Case study small	extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG	{1.0: 156922, 2.0: 374031, 3.0: 12338}	{1.0: 111188, 2.0: 372150, 3.0: 7714, 10.0: 52239}	90.38 (%)	1.02 sec
Apply Logistic regression Case study Big	extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG	{1.0: 1615024, 2.0: 5662406, 3.0: 1079254}	{1.0: 1507905, 2.0: 5640622, 3.0: 456111, 10.0: 752046}	91.00 (%)	5.83 sec
Trained NN Balance data	Single layer batch_size=32 epochs=25	{1: 2549, 2: 1583, 3: 1711}	{1: 1934, 2: 1442, 3: 1705, 10: 762}	87.28 (%)	48.80 sec
Trained NN Balance data	Single layer batch_size=32 epochs=50	{1: 2504, 2: 1567, 3: 1772}	{1: 1926, 2: 1468, 3: 1763, 10: 686}	88.26 (%)	53.77 sec
Apply NN Case study small	Single layer batch_size=32 epochs=50	{1: 206315, 2: 324973, 3: 12003}	{1: 111442, 2: 323258, 3: 7281, 10: 101310}	81.35 (%)	10.88 sec
Apply NN Case study Big	Single layer batch_size=32 epochs=50	{1: 1731125, 2: 5518907, 3: 1106652}	{1: 1510717, 2: 5495474, 3: 451218, 10: 899275}	89.24 (%)	2.86 Min
Trained NN Balance data	2 hidden Dence layer units=100, activation="relu" units=4, activation="softmax" Adam(learning_rate=0.02, decay=1e-6) loss="categorical_crossentropy", metrics=["accuracy"]	{1: 1926, 2: 1967, 3: 1950}	{1: 1902, 2: 1920, 3: 1946, 10: 75}	98.72 (%)	1.08 Min
Apply NN Case study small	“	{1: 108365, 2: 425323, 3: 9603}	{1: 107056, 2: 419383, 3: 7789, 10: 9063}	98.33 (%)	12.06 sec

Apply NN Case study Big	“	{1: 1505733, 2: 6247369, 3: 603582}	{1: 1496175, 2: 6214042, 3: 456431, 10: 190036}	97.73 (%)	2.85 Min
Trained NN Balance data	2 hidden Dence layer Drop-out=0.2 units=100, activation="relu" units=4, activation="softmax" Adam(learning_rate=0. 02, decay=1e-6) loss="categorical_cross entropy", metrics=["accuracy"]	{1: 1997, 2: 1898, 3: 1948}	{1: 1997, 2: 1898, 3: 1948}	98.15 (%)	1.65 Min
Apply NN Case study small	“	{1: 128282, 2: 406118, 3: 8891}	{1: 111535, 2: 404645, 3: 7777, 10: 19334}	96.44 (%)	19.03 sec
Apply NN Case study Big	“	{1: 1573735, 2: 6151794, 3: 631155}	{1: 1513357, 2: 6135533, 3: 456310, 10: 251484}	96.99 (%)	3.63 Min
Trained NN Balance data	3hidden Dence layer units=100, activation="relu" units=4, activation="softmax" Adam(learning_rate=0. 02, decay=1e-6) loss="categorical_cross entropy", metrics=["accuracy"]	{1: 1995, 2: 1898, 3: 1950}	{1: 1939, 2: 1889, 3: 1947, 10: 68}	98.84 (%)	1.18 Min
Apply NN Case study small	“	{1: 128120, 2: 406113, 3: 9058}	{1: 112500, 2: 405616, 3: 7788, 10: 17387}	96.80 (%)	12.39 sec
Apply NN Case study Big	“	{1: 1578029, 2: 6121657, 3: 656998}	{1: 1522599, 2: 6114838, 3: 456515, 10: 262732}	96.86 (%)	2.98 Min
Trained NN Balance data	3hidden Dence layer Drop-out=0.2 units=100, activation="relu" units=4, activation="softmax"	{1: 2118, 2: 1766, 3: 1959}	{1: 1939, 2: 1757, 3: 1947, 10: 200}	96.58 (%)	1.44 Min

	Adam(learning_rate=0.02, decay=1e-6) loss="categorical_crossentropy", metrics=["accuracy"]				
Apply NN Case study small	“	{1: 124826, 2: 407948, 3: 10517}	{1: 110653, 2: 405605, 3: 7789, 10: 19244}	96.46 (%)	16.52 sec
Apply NN Case study Big	“	{1: 1568107, 2: 6004446, 3: 784131}	{1: 1511255, 2: 5986276, 3: 456504, 10: 402649}	95.18 (%)	3.91 Min
Trained CNN Balance data	3hidden-layer Drop-out Conv2D	{1: 2003, 2: 1868, 3: 1972}	{1: 1870, 2: 1783, 3: 1940, 10: 250}	95.72 (%)	27.75 sec
Case study small	“			(%)	
Case study Big	“			(%)	

7. conclusion and suggestion

According labeled data, all results seem good. In other hand, as mentioned labeled data-set built by NDVI’s threshold. These thresholds are impalement and are not precise. Also data-set re-classed into three general categories (water, vegetation and bare land). These categories are not complete too. Because, as Figure 8 in case study small should have minimum 14 classes. For example, for a water body, the author classified all water as one class, whereas Urmia lake has a river. Then there should be minimum three classes of water. All of the above mentioned plus optimum hyper-parameter and number of layers, could affect on the final result.

The author has suggested additional spectral signature, some other features calculate. Such as LST[24], aspect, slope and RADAR data-set.

Another suggestion is, for labeling must be an ISO (standard) table. A prototype of table showed in Table 7.

24 Land surface temperature

Table 7: ISO (standard) labeling example for Deep learning methods

Features	labeling	Features	labeling	Features	labeling
cloud	0000				
Smoke pollution	1000	city	1100		
		wood	1010		
		oil	1001		
Water	20000	Flowing water	21000		
		Water interface	20100		
		Saline water	20010		
		Wast water	20001		
Vegetation	300	agriculture	310	farm	31010
				garden	31001
		Natural resources	301		
Bare land	400	soil	410		
		rock	401	Igneous	401100
				Metamorphic	401010
				Sedimentary	401001

8. acknowledge

….

9. References

[1] E. Chuvieco, *Fundamentals of satellite remote sensing: An environmental approach*. CRC press, 2016.

[2]............E. Chuvieco, *Earth observation of global change: The role of satellite remote sensing in monitoring the global environment*. Springer, 2008.

[3].........N.-B. Chang and K. Bai, *Multisensor data fusion and machine learning for environmental remote sensing*. CRC Press, 2018.

[4].................R. Pu, *Hyperspectral remote sensing: Fundamentals and practices*. CRC Press, 2017.

[5]..E. Sentinel, “User Handbook. 2015,” 2.

[6]............................K. P. Murphy, *Machine learning: a probabilistic perspective*. MIT press, 2012.

[7]..E. Alpaydin, *Machine learning: the new AI*. MIT press, 2016.

[8].............D. Sarkar, R. Bali, and T. Ghosh, *Hands-On Transfer Learning with Python: Implement advanced deep learning and neural network models using TensorFlow and Keras*. Packt Publishing Ltd, 2018.
[9].I. Vasilev, D. Slater, G. Spacagna, P. Roelants, and V. Zocca, *Python Deep Learning: Exploring deep learning techniques and neural network architectures with Pytorch, Keras, and TensorFlow*. Packt Publishing Ltd, 2019.
[10]..E. Alpaydin, *Introduction to machine learning*. MIT press, 2020.
[11].….S. Schaal and C. G. Atkeson, “Robot juggling: implementation of memory-based learning,” *IEEE Control Systems Magazine*, vol. 14, no. 1, pp. 57–71, 1994.