PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

ESCUELA DE POSGRADO

Clasificación de cultivos de quinua orgánica mediante el uso de imágenes aéreas multiespectrales y técnicas de aprendizaje automático

Classification of organic quinoa crops using multispectral aerial imagery and machine learning techniques

TRABAJO DE INVESTIGACIÓN PARA OPTAR EL GRADO ACADÉMICO DE MAGÍSTER EN FÍSICA APLICADA

AUTOR

DONATO ANDRÉS FLORES ESPINOZA

ASESOR

MIGUEL ANGEL CATAÑO SANCHEZ

Septiembre, 2021

RESUMEN

Según datos de la FAO, la planta de la quinua se originó en los alrededores del lago Titicaca que comparten Perú y Bolivia, destaca por la calidad nutricional que posee, así también los requerimientos del cultivo en cuanto a la poca exigencia en la cantidad de agua, lo cual lo hace un cultivo ideal para contrarrestar el cambio climático. La producción mundial de quinua se encuentra liderada primordialmente por los países de Perú y Bolivia, quienes concentran entre el 83% (año 2015) al 67% (año 2016). En ambos países dicha actividad involucra a no menos de 150,000 familias de pequeños productores. La producción de dicho cultivo es mayoritariamente orgánica con promedios bajos de producción. Como consecuencia del escaso uso de tecnologías productivas, el incremento de plagas, así como de la variabilidad cada vez más impredecible de las condiciones climáticas, los agricultores familiares de ambas regiones continúan presentando niveles de pobreza de 42.9% y extrema pobreza de 21.3% en promedio. (INEI-Perú, INE Bolivia). El presente trabajo de investigación contribuye al mapeo de cultivos el cual constituye una herramienta esencial para la gestión agrícola y la seguridad alimentaria. El objetivo del trabajo de investigación es la evaluación de diversos métodos de clasificación del cultivo de la quinua que permitirá realizar el mapeo automático. El estudio se centra en el uso de técnicas de aprendizaje automático para clasificar cultivos de quinua a partir de imágenes aéreas multiespectrales tomadas desde un sistema aéreo no tripulado. La reflectancia espectral de cinco bandas ópticas se utiliza para determinar modelos de clasificación que se evalúan en las diferentes etapas fenológicas de la quinua. Se exploraron estrategias de aprendizaje automático a las imágenes obtenidas, tales como árboles de decisión, análisis discriminatorio, máquinas de vectores de soporte, K vecino más cercano, conjunto de clasificadores, métodos de aprendizaje profundo de Segnet y Unet. Los conjuntos de datos de entrenamiento se obtuvieron de las ubicaciones de los campos de quinua en Cabana en la región Puno de Perú. Los resultados muestran que las técnicas de aprendizaje profundo superan a otras técnicas en la tarea de clasificación. Se muestran las pruebas realizadas sobre las diversas etapas fenológicas en donde las técnicas de aprendizaje profundo obtienen una precisión de entre 81% y 95%, mientras que las demás técnicas su precisión fueron entre 58% y 87%.

Classification of organic quinoa crops using multispectral aerial imagery and machine learning techniques

Andres Flores, *Senior Member, IEEE*

***Abstract*— Crop mapping is an essential tool for agricultural management and food security for which remote sensing data can be used. This study focuses on the use of machine learning techniques to classify quinoa crops from multispectral aerial images. Spectral reflectance of five optical bands are used for determining classification models which are evaluated for different phenological stages of quinoa. Decision Trees, Discriminant Analysis, Support Vector Machines, K nearest Neighbor, Ensemble Classifiers, deep learning methods Segnet and Unet were explored. Training datasets were obtained from quinoa crop fields locations at Cabana in the Puno region of Peru. An unmanned aircraft system (UAS) was used to acquire the multispectral images from an altitude of 50 meters. Results show that deep learning techniques outperform the other techniques in the classification task.**

I. Introduction

The Puno plateau region of Peru and Bolivia is used for cultivating many types of vegetable crops: alfalfa, oats, potato, pasture, barley, broad beans, cañihua and quinoa. An important agricultural economic activity is organic quinoa which is exported to important world markets. Due to its high nutritional value it is considered as a food that has become very important for food security. It also stands out for the requirements of the crop in terms of the low demand in the amount of water, which makes it an ideal crop to counteract climate change [1] [2].

Since the second quarter of 2018, the Pontifical Catholic University of Peru together with the Association of Agricultural Producers of Vizallani1 have been executing a precision agriculture project that will help to optimize the processes and procedures for organic certification carried out each year by local farming organizations. The project also includes the registration of georeferenced maps of the productive units and multispectral images captured by an unmanned aircraft vehicle that will jointly improve the competitiveness of this activity for the benefit of smallholder farmers.

The Association of Agricultural Producers of Vizallani1 is located in southern Peru in the Puno plateau as shown in Fig. 1. Smallholder farmers near the Cabana district (15°38'58.1"S 70°19'18.9"W, elevation 3,901 m.a.s.l.) are organized in fields of land of various sizes of no more than one hectare, which makes it difficult to assess using satellite imagery. Determination of remote sensing data is best obtained using unmanned aircraft systems (UAS), flying below 50 meters above the fields. These systems can deliver very high resolution images with a spatial resolution of five centimeters.

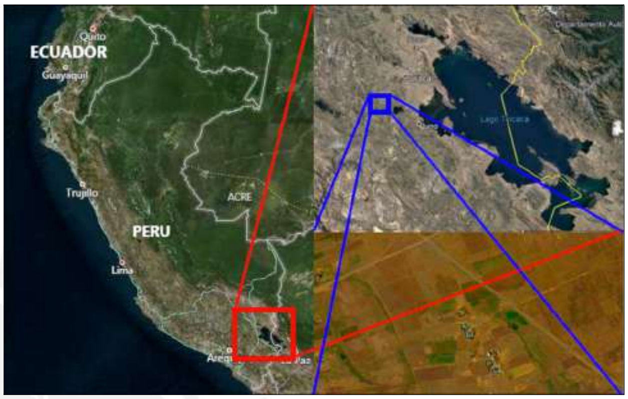


Fig. 1. Location of the Association of Agricultural Producers of Vizallani1

As part of this project, technological developments specific to the cultivation of quinoa is being carried out, taking into account the difficult geographical conditions due to altitude (3,901 m.a.s.l) and limited knowledge of the population to information and communication technologies. Among the technological developments that the mentioned project raises are the use of multispectral images acquired from an unmanned aircraft vehicle and consequent processing in order to estimate parameters that allow the adequate use of inputs such as water, fertilizers and pesticides. In this research the need to classify the cultivation terrain in an automated way arises and will become the starting point for the analysis of the crop in subsequent stages. Therefore, the generation of georeferenced maps that allow the farmer to estimate the amount of quinoa cultivation and therefore its yield production is of vital importance for food security. To generate maps, proper classification of land cover is required, for which machine learning strategies need to be evaluated to the multispectral images acquired using an unmanned aircraft vehicle.

Remote sensing science is playing a significant role in precision agriculture being crop classification an important first step for monitoring, assessment and crop management. Satellite technology is a convenient means for acquiring imaging data and suitable for great extensions of land. Several studies [3]-[8] have shown the effectiveness of machine learning (ML) techniques on several types of crops mainly on corn and soy bean using satellite images in global, continental, and regional scales. Smallholder farming systems farming generally consists of many heterogeneous fields with different types of crops in small areas. It is common for crops to rotate on a particular field, being a common practice especially for organic farming. Due to these conditions a precise and automated method is required to correctly map these fields using high resolution imaging for which unmanned aircraft systems (UAS) are more suitable than

*Research supported by Innovate Peru, Ministry of Production.

A. Flores is with the Pontificia Universidad Católica del Perú, Lima, PERU (phone: 51-999-650-502; e-mail: dflores@ pucp.edu.pe).

satellites because they can better adapt to different time and spatial requirements [9], [10]. There is an increased interest in using UAS for crop classification due to its flexibility for smallholder farming [11]-[14]. Regarding machine learning techniques, several methods have been applied for this task among them decision trees [3], discriminant analysis [4], random forest classification [5], [6], [7], [14], maximum likelihood method [5], support vector machines [5], [9], [14], [17] and deep learning [12]-[16]. Current research can be divided into several fronts, on the one hand about the type of data to be processed, be multispectral images that consider the spectrum of light reflected by the leaves of crops. The other aspect is temporal considering phenological characteristics, that is, the growth development of the plant. What is evident from the present state of the art is that it becomes necessary to consider all these aspects, which implies involving a large amount of data [5]. In this work we consider both types of approach towards an effective and automated method for crop classification. Several machine learning techniques are considered using high spatial resolution time series data.

Light is composed of particles of energy called photons, or quanta of light. The energy of a photon is inversely proportional to its wavelength, the longer the wavelength, the lower the energy. Photons of violet light, for example, have almost twice the energy of photons of red light, the longest visible wavelength. Photosynthesis is the process in which plant cells incorporate CO_2 from the air into its organic molecules. For this to happen light energy is converted to chemical energy of the organic molecules. These molecules are important for photosynthetic organisms: they are the building blocks of cells and a source of chemical energy that fuels the metabolic reactions that sustain life. It is known that radiation within the visible light portion of the spectrum excites certain types of biological molecules, moving electrons into higher energy levels. Radiation with wavelengths longer than those of visible light does not have enough energy to excite these biological molecules. For light to be used by living organisms it must be absorbed, substances that are able to do this are called pigments. Most pigments absorb certain wavelengths and transmit or reflect other wavelengths. In the case of plant organisms, chlorophyll is the main pigment of photosynthesis, it absorbs light primarily in the blue (400 to 500nm) and red (600 to 700 nm) regions of the visible spectrum. Green light (500 to 600nm) is not appreciably absorbed by chlorophyll. Plants usually appear green because some of the green light that strikes them is scattered or reflected. There are many types of chlorophyll pigments in nature, three of them are the most important for photosynthesis in plants, chlorophyll a, chlorophyll b and carotenoids. The absorption spectrum of a pigment is a plot of its absorption of light at different wavelengths. The upper part of Fig. 2 shows the absorption spectra for chlorophyll a, chlorophyll b and carotenoids as a function of wavelength. Shown in the lower part of Fig. 2 is the action spectrum of photosynthesis which is the relative effectiveness of different wavelengths of light. The action spectrum for photosynthesis shows which wavelengths are used by plants to create energy, while the absorption spectrum shows which wavelengths are most absorbed by a specific molecule [19], [20].

In order to acquire images with an optical camera, reflectance must be considered. As is known light energy from the sun will encounter a plant leaf and will fraction in absorbed, transmitted and reflected energy. There is a wavelength dependency that for a certain type of pigment the proportion of reflected, absorbed and transmitted energy will vary at different wavelengths [22].

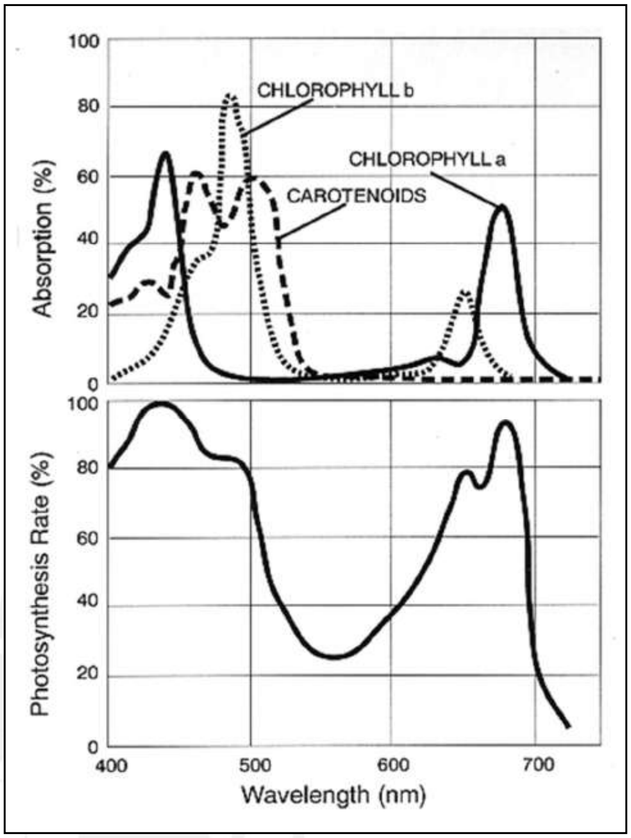


Fig. 2. Absorption spectrum and Action Spectrum. Source: [21]

It is evident from Fig. 2 that plants will have no reflectance in regions blue and red, but relative high reflectance in the green region. Reflected energy can be plotted as a function of wavelength depicting a “spectral signature”. Fig. 3 shows spectral signatures of various types of surface on a broader range of wavelengths. Reflectance of vegetation reveals what was shown as absorption in the visible part of the spectrum. Leaf pigments are transparent to near-infrared wavelengths (700 to 1300 nm), and leaf absorption is small. Most of the energy is transmitted and reflected, dependent on leaf structural characteristics, which results in a high near-infrared (NIR) plateau. The sharp rise in reflectance between the red and NIR regions is known as the red edge (680 to 750 nm) and has been found to be dependent on chlorophyll concentration and can be used in plant stress detection. The middle infrared (MIR) region (1300 to 2500 nm) is dominated by soil and leaf water absorption, particularly at 1400 and 1900 nm with reflectance increasing when leaf liquid water content decreases [23], [24]. Information on how sun based radiation interacts with vegetation is important to decipher and process remote sensing data for agriculture. A plant leaf regularly has a low reflectance in the visible spectrum due to absorption by chlorophylls, a generally high reflectance in the near infrared as a result of internal leaf scattering and no absorption, and a moderately low reflectance in the infrared beyond 1.3µm due to absorption by water. Plant canopy reflectance is similar, but is modified by non-uniformity of incident solar radiation, plant structures, leaf area, shadows, and reflectivity of the background. Airborne sensors get an integrated view of all of these effects, and each type of crop or vegetation tends to have a characteristic signature that allows for discrimination and classification. Because of the sensitivity of chlorophyll to physiological disturbances when disease and physiological stress affect the reflectance properties of individual leaves, the most pronounced changes often occur in the visible spectral region rather than in the infrared.

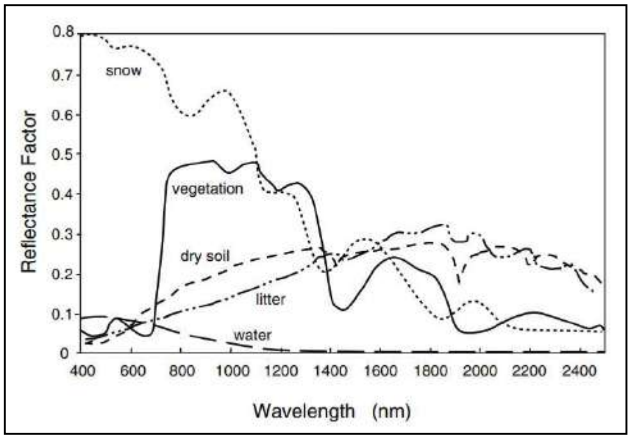


Fig. 3. Reflectance spectral signatures. Source: [23]

However, the primary basis for aerial remote sensors to detect stress conditions in a crop or other vegetation is often not a change in the reflectance characteristics of individual leaves, but a reduction in the total leaf area exposed to the sensors. Such reduction can result from a direct loss of leaves, a change in their orientation, or overall plant growth suppression. In such situations, the overall infrared reflectance tends to be reduced significantly more than the visible reflectance due to fewer different leaf layers and an increase in background exposure [25].

All these aspects regarding spectral information of leaf and canopy characteristics constitute valuable information that can be acquired using multispectral sensors considering additional aspects like texture and non-vegetation surroundings. All of which constitute data that describe characteristics of a particular plant or crop that is subject to be classified.

It is the aim of this work to describe and evaluate several machine learning methods to classify quinoa crops with the use of aerial multispectral images acquired from an unmanned aircraft system. Time series acquisition is considered to deal with phenological aspects of the crop. The generation of an automated and efficient method for the determination of quinoa crop pixels in aerial images is accomplished.

II. Materials

A. Study area

The region of study focuses on agricultural fields of quinoa. Data was collected by means of a DJI Phantom 4 unmanned aircraft system (UAS) with a Micasense Rededge multispectral camera, which acquired images up to 50 meters above the fields. Fig. 4 shows the UAS system employed during takeoff at one of the sites.

The data was obtained from fields belonging to farmer Daniel Cárdenas who is part of the before mentioned association. The study area consists of two Quinoa fields of Salcedo INIA variety and its surroundings. The two fields shown in Fig. 5 are “Hatumpata Yurac Cruz 4” (HYC4) with an area of 700 m^2 (0.07 Ha) and “Hatumpata Yurac Cruz 5” (HYC5) with an area of 800 m^2 (0.08 Ha). Surrounding ground cover consisted of other types of vegetation (alfalfa, oats and grass), bare soil and house structures occupying an area of 6400 m^2 (0.64Ha).

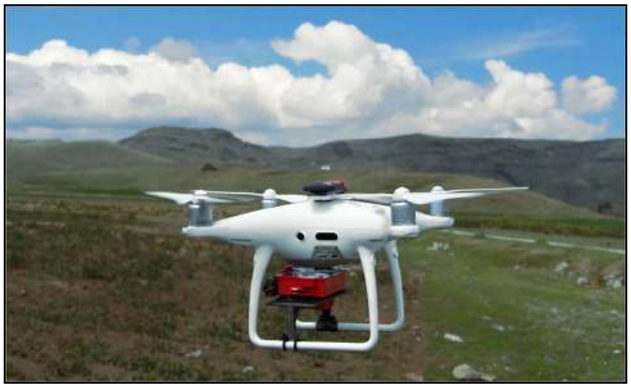


Fig. 4. DJI Phantom 4 with onboard Micasense Rededge Camera.

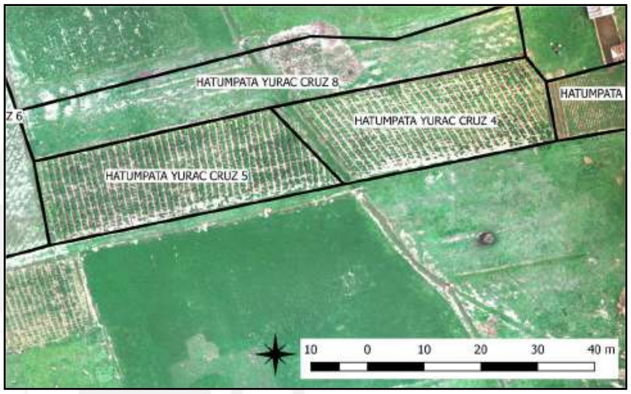


Fig. 5. Aerial view of region of study showing Quinoa fields HYC4 and HYC5.

B. Image data

As is known, the principle behind digital optical cameras is that they are composed of a large number of electronic sensors, which convert reflected light energy to an electrical signal. This signal is then digitized and a numerical value that is proportional to the intensity of the light energy is obtained [18].

A Rededge Multispectral camera from Micasense capable of acquiring images in five narrow spectral bands was used. Spectral bands used were blue (Center Wavelength: 475nm, Full Width at Half Maximum (FWHM) Bandwidth: 20nm), green (Center Wavelength: 560nm, FWHM Bandwidth: 20nm), red (Center Wavelength: 668nm, FWHM Bandwidth: 10nm), near-infrared (NIR, Center Wavelength: 840nm, FWHM Bandwidth: 40nm) and red edge (Center Wavelength: 717nm, FWHM Bandwidth: 10nm), with a spatial resolution depends on flight altitude being between of 1.2 to 2.6 cm per pixel, 1 capture per second, 12-bit RAW and 1280 x 960 pixels. Spectral response of each band is shown in Fig. 6 with a comparison to typical plant reflectance. For each band an image is obtained constituting a five dimensional vector data for each pixel. Table I shows the image data details acquired.

To ensure spatial variability approximately 160 images were acquired over an area of 1.60 hectares, at an altitude of 50 meters during a mission flight. This mission flight was repeated during six occasions (dates: 17/12/2018, 4/1/2019, 25/1/2019, 2/2/2019, 16/2/2019 and 28/2/2019) in order to have temporal variability during different stages of the crop.

TABLE I
IMAGE DATA ACQUIRED

Orthomosaic Image	Number of images	Date	Quinoa phenology stage	Class	Number of pixels	Coverage Area (m^2)	Flight Altitude (m)	Image resolution (cm/pixel)
1	115	27-12-2018	Four True Leaves	Quinoa	1762789	2870	17.1	1.21
				Other Vegetation	2464839			
				Non Vegetation	46717562			
2	162	04-01-2019	Six tree leaves	Quinoa	1219951	11000	28.8	1.94
				Other Vegetation	7390730			
				Non Vegetation	44565629			
3	95	25-01-2019	Branching	Quinoa	467815	16100	48.7	3.25
				Other Vegetation	9849099			
				Non Vegetation	15615408			
4	165	02-02-2019	Ear formation	Quinoa	521403	12300	39	2.62
				Other Vegetation	9895510			
				Non Vegetation	17162312			
5	143	16-02-2019	Florescence	Quinoa	555173	13200	38.2	2.56
				Other Vegetation	12332686			
				Non Vegetation	21881649			
6	87	28-02-2019	Maturity	Quinoa	246830	9410	37.9	2.54
				Other Vegetation	7628707			
				Non Vegetation	17720278			

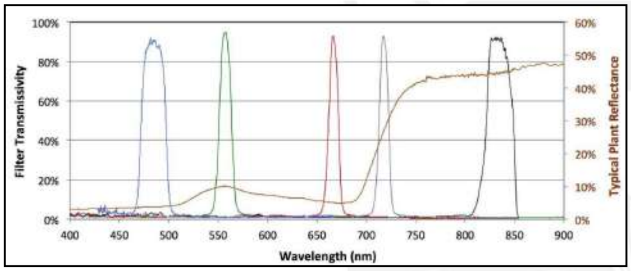


Fig. 6. Micasense Rededge Multispectral Camera Spectral bands compared to Typical Plant Reflectance. Source: [26]

Obtained data included quinoa, alfalfa, oats, and grass, soil and house structures. Fig. 7 shows an RGB version of the orthomosaic image over the fields being quinoa crops the ones located at the center acquired on January 25, 2019. Because of the variation of area coverage for each orthomosaic image as shown in Table I all time data is not necessarily the same in spatial terms. But in all cases the Quinoa fields were considered, so the variation only affects the other vegetation and non-vegetation classes for which there is plenty of data.

C. *Preprocessing of images*

Images need to be assembled in order to have an overall representation of the land cover that the UAS has flown over. Photoscan software from Agisoft was used to obtain an orthomosaic image of the fields in GEOTIFF format. Six orthomosaic images were obtained constituting the time series data for machine learning procedure.

Automated vegetation detection and masking were carried out on each image using the Normalized Differential Vegetation Index (NDVI). The method is based on the difference of reflectance in the near-infrared and red bands as in:

$$NDVI = \frac{NIR-RED}{NIR+RED} \quad (1)$$

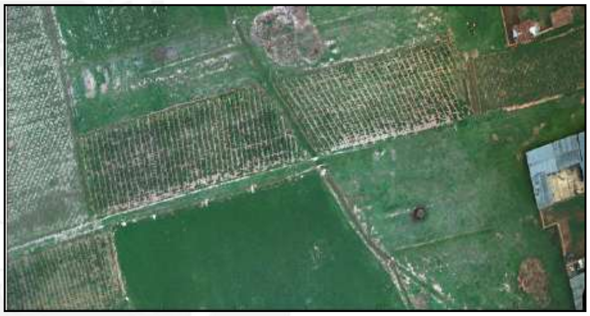


Fig. 7. RGB image acquired on January 25, 2019.

RED and NIR are the reflectance for red and near infrared spectral bands, respectively. High NDVI values will result from the combination of a high reflectance in the near infrared and lower reflectance in the red band. This combination is typical of the spectral “signature” of vegetation. Non-vegetated areas, including soil, water, snow, ice, and construction materials, will have much lower NDVI values. Typical values of NDVI for vegetation are in the range of 0.3 and 0.8 [27]. A simple threshold logic technique was used to select pixels as vegetation.

$$V = \begin{cases} Im, & NDVI \geq 0.3 \\ 0, & NDVI < 0.3 \end{cases} \quad (2)$$

Where V represents the segmented image and Im is the original multispectral image. In this way vegetation is segmented from other types of land cover as shown in Fig. 8, where black pixels refer to non-vegetation. This segmentation mask V will be used for the needed classification of vegetation. A pixel based strategy was conceived in which a certain region was selected. Quinoa and non-quinoa vegetation pixels were manually selected from image V and labeled considering farmer knowledge for each image. In this way ground truth data was obtained for each image. Fig. 9 shows the result of this mixed NDVI and manual process in which 3 pixel classes are segmented: quinoa crop, other types of vegetation and non-vegetation pixels shown in yellow, light green and blue respectively. This process was repeated for the six orthomosaic image time series obtaining the necessary labeled data ground truth for subsequent processing. Table I shows the image data obtained.

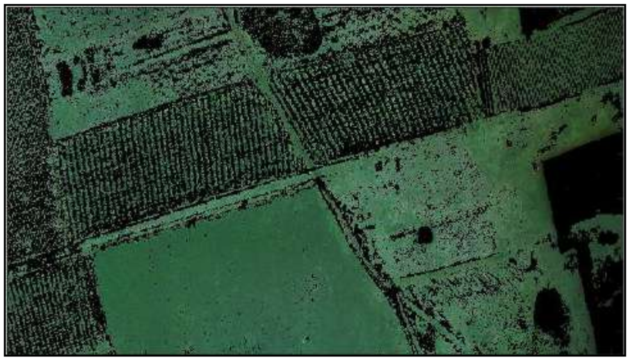


Fig. 8. RGB image of the vegetation segmented image

D. *Dataset preparation*

To perform machine learning, a dataset needs to be constructed from the othomosaic multispectral images. Pixel data is represented by a five dimensional vector considering reflectance in five spectral bands. Data was divided in training and testing sets as shown in Table I. Two regions were considered for the training dataset the HYC4 region and its surroundings, the HYC5 region and its surroundings were selected for the testing dataset. Figures 11 and 12 show the image time series dataset for training and testing respectively.

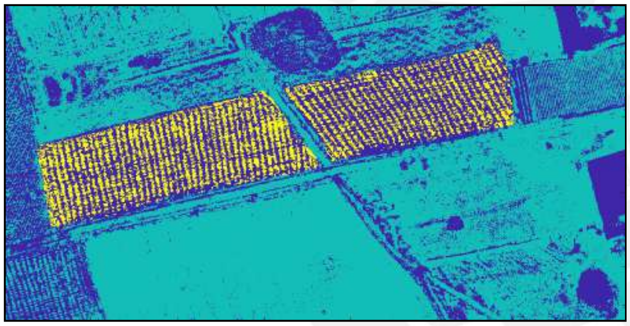


Fig. 9. Ground truth image of January 25, 2019, quinoa crop, other types of vegetation and non-vegetation pixels are shown in yellow, light green and blue pixels respectively.

III. Methods

A supervised learning approach was used to classify land cover. Three classes were considered for the machine learning stage: quinoa crop, non-quinoa crop and non-vegetation,. In this work, six approaches of machine learning algorithms were used: Decision Trees, Discriminant Analysis, Support Vector Machines, K-nearest Neighbor, Ensemble Classifiers and Deep Learning. These techniques are often cited in state of the art research on land cover classification and comparisons of results between some of them has been the main focus. In the following sections a brief explanation of these classifiers are described for several types of land cover emphasizing on crop applications.

A. *Decision Trees*

A decision tree (DT) is a flowchart tree like structure in which a node is where a decision must be made, each branch represents a probable result of the decision and each leaf represents a class label. The method of labeling is considered to be a series of simple decisions based on sequential test results, rather than a single, complicated decision. Sets of decision sequences form the DT branches, and tests are applied to the nodes. The leaves represent the labels. DT construction requires the recursive partitioning of a collection of training data, which is divided into progressively homogeneous subsets based on tests applied to one or more of the characteristic values. Such tests are represented by nodes as shown in Fig. 10. The decision tree is determined using the CART algorithm [28]. Studies have been done to evaluate Decision Trees for land cover classification using satellite multispectral images and is regarded as a computationally efficient method [29], [30]. This method can be regarded as simple but has gained acceptance due to the fact that it has derived to newer techniques such as the Random Forest classifier.

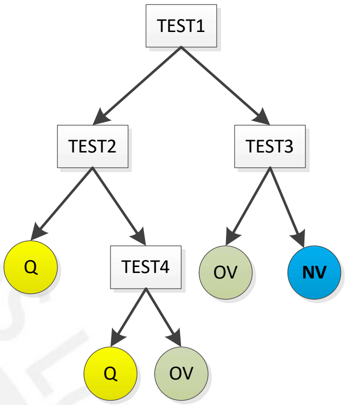


Fig. 10. A decision tree classifier example. Each box is a node at which tests are applied to split the data into smaller groups. Labels Q, NQ and NV are leafs which refer to classes quinoa crop, other types of vegetation and non-vegetation.

B. *Discriminant Analysis*

Discriminant Analysis is a method used to find a linear combination of features that characterizes or separates two or more classes of objects or events. It assumes that different classes generate data based on different Gaussian distributions. It learns the most discriminative axes between the classes, and these axes can then be used to define a hyperplane onto which to project the data [31], [32]. This method has been used for land use classification using multispectral images [33], [34]. It is also used in data classification where reduced dimensionality is needed which is the case for hyperspectral images [35].

C. *Support Vector Machines*

Support Vector Machine (SVM) is a kernel based classification algorithm that considers data as an N dimensional vector (where N is the number of features, five in our case). The goal is to determine hyperplane boundaries that enclose groups of data that, after training, eventually belong to a certain class. It does this by maximizing margins between decision boundaries, which constitutes a nonlinear optimization problem. Support vectors constitute data that define the hyperplane boundaries that separate distinct classes [36], [37], [41, [42], [43]. Fig. 13 shows a simple example for a linear binary data in which data is classified and a hyperplane boundary is represented by a line. Data is represented in by two features. In our case in which data is represented by 5 features the hyperplane required is a 4 dimensional boundary. Crop type classification using this method for multispectral satellite images has been achieved [5]. Images from unmanned aircraft vehicles (UAV) provide higher resolution, which has the advantage of providing texture information in addition to the spectral data are convenient for the SVM classification of crops [14].

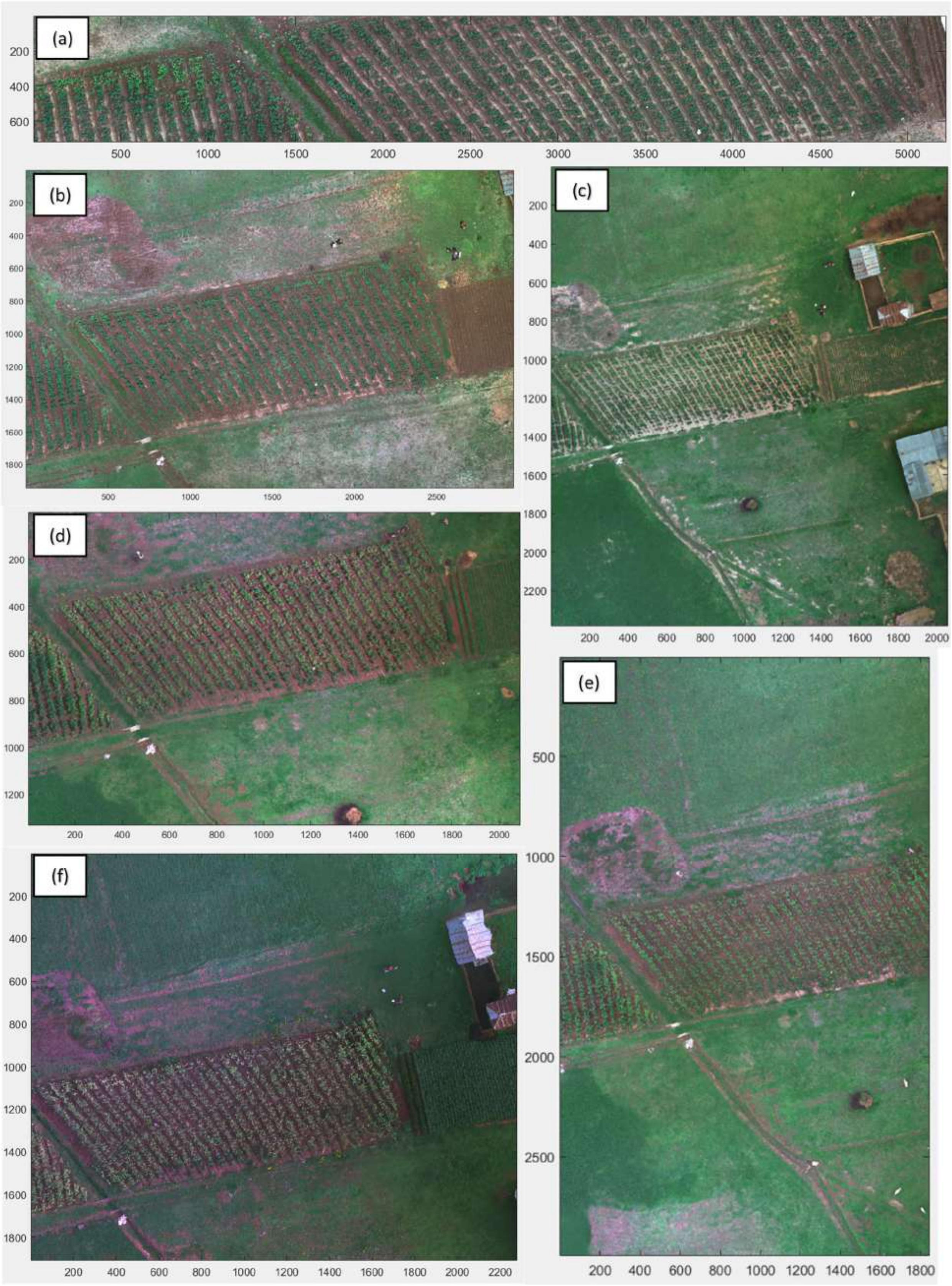


Fig. 11. RGB version time series images of HYC4 region and surroundings used as training dataset, dates: a) 27-12-2018, b) 04-01-2019, c) 25-01-2019, d) 25-01-2019, e) 16-02-2019 and f) 28-02-2019

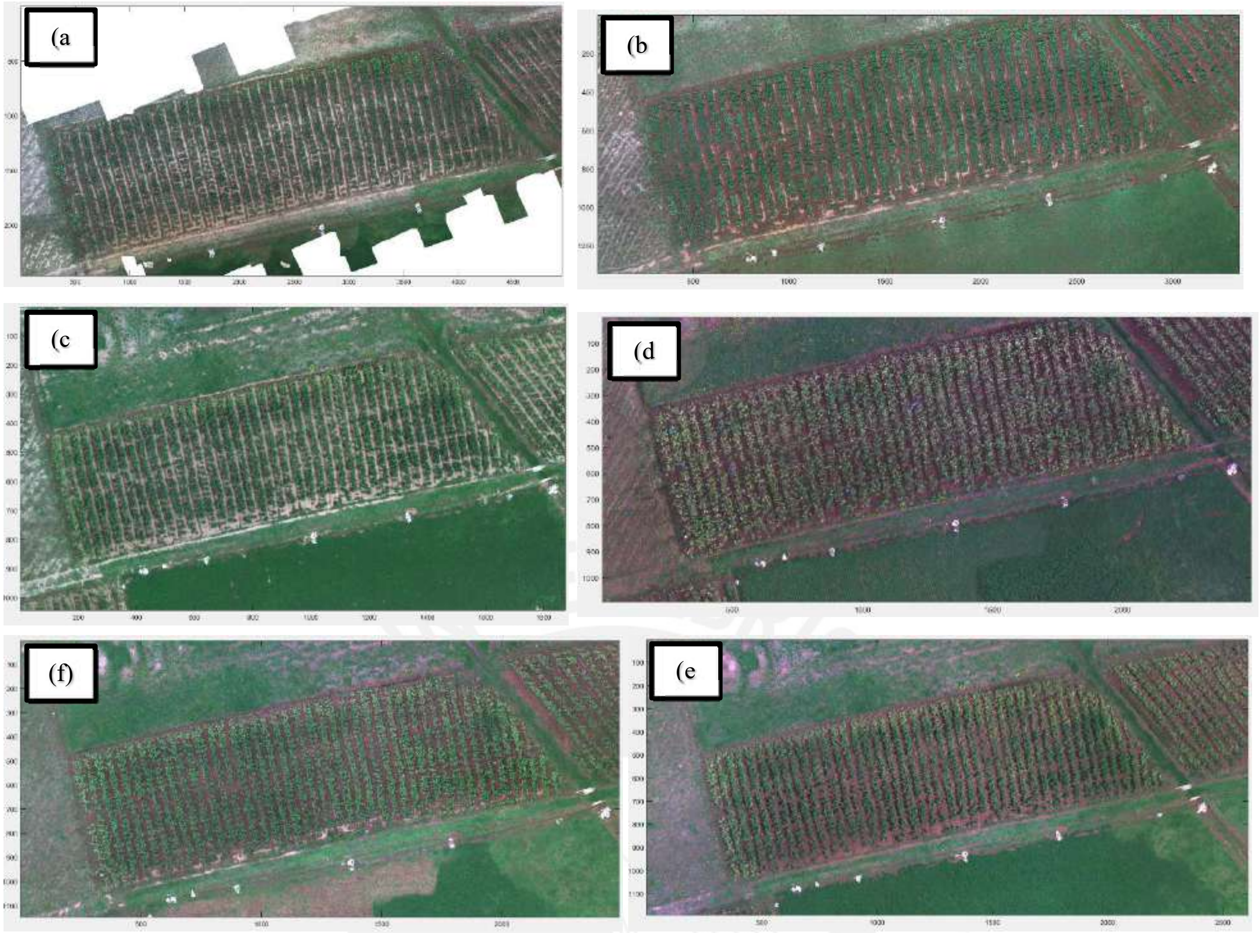


Fig. 12. RGB version time series images of HYC5 region and surroundings used as testing dataset, dates: a) 27-12-2018, b) 04-01-2019, c) 25-01-2019, d) 25-01-2019, e) 16-02-2019 and f) 28-02-2019

Compared to other non-Deep Learning techniques the SVM algorithm has proved to have better performance for land cover classification [38], [39], [40].

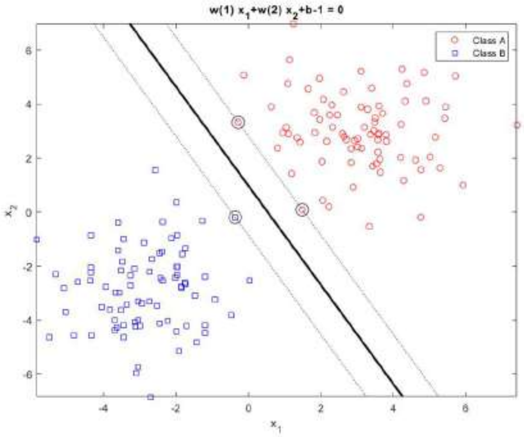


Fig. 13. A Linear Support Vector Machine classification example for 2 features x_1 and x_2 in which 2 distinct data are classified. The bold line represents the hyperplane boundary and the dotted ones are the margins. Support vectors are displayed as the 3 circled data [38].

D. *K Nearest Neighbor*

The goal is to memorize the training set, and then determine the label of each new instance based on the labels of the training set's closest neighbors. The reasoning for such approach is based on the premise that the features used to define the domain points are applicable to their labels such that close-by points are likely to be labeled the same way.

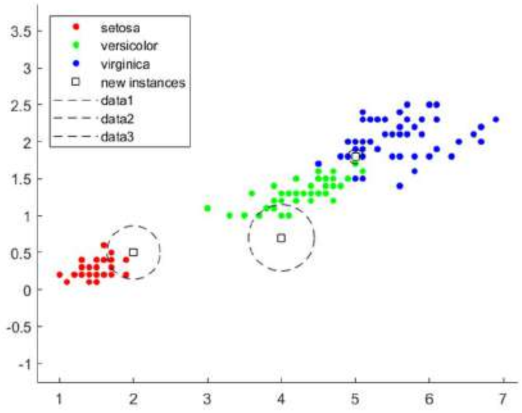


Fig. 14. A KNN classification example for 2 features petal width vs petal length in which 3 distinct Iris species are classified. Three new instances are classified consdering K=5 and are shown as circled data [44].

In addition, in certain cases, locating a nearest neighbor can be achieved extremely fast, even though the training set is vast. An object is classified based on the class attributes of its K nearest neighbors. Hence, in this classifier, K is the main tuning parameter, which largely determines the performance of the KNN classifier. Fig. 14 shows the use of KNN for the known Fisher's Iris dataset [32] in which Iris flower species are classified from two features. This technique has been used for classification of crops [39], and land cover with satellite images [40]. These studies compare classification results with other types of strategies.

Random Forest Classifier

Ensemble classifiers refers to several methods that consists of a combination of classifiers in which a better performance can be obtained in comparison to applying individual models. They develop several predictive models from resampled versions of training data and combine predictions by simple averaging or voting. The random forest algorithm is a bagging (bootstrap aggregating) type of ensemble classifier. It makes use of multiple decision tree predictors using different random samples from the training dataset. A voting scheme is used to decide the outcome from the predictors. Fig. 15 shows an example using three decision trees. This classifier is often cited for comparing results for crop and land cover classification [5], [6], [7], [41], [42], [43], [45].

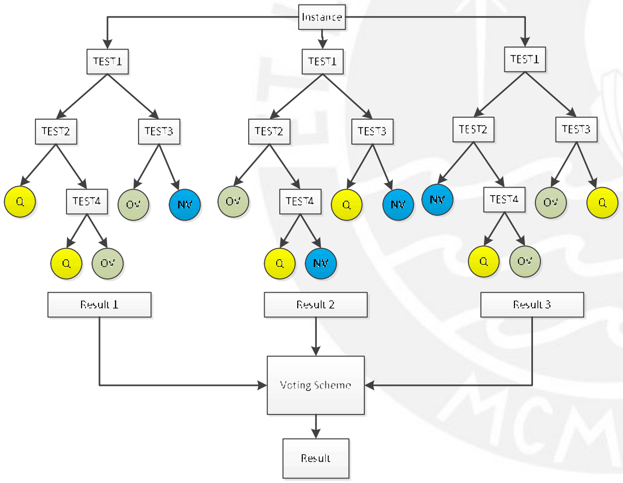


Fig. 15. A Random Forrest classification example in which an instance is classified using 3 decision trees. The results are subject to a voting scheme to obtain the final result.

E. *Adaboost Classifier*

Another type of ensemble approach is boosting where many weak classifiers can be combined into a strong classifier. The basic concept of most boosting methods is to train sequential predictors, each attempting to correct their predecessor. The main boosting algorithm is AdaBoost (adaptive boosting). The basic principle behind Adaboost is to set the weights of the classifiers in each iteration and train the data sample to ensure accurate predictions of unusual observations. Any machine learning algorithm, if accepting weights on the training set, can be used as a base classifier. It typically uses a decision tree as the base classifier. The output is a "solid" classifier centered on a weighted sum of all the weak hypotheses [41], [42], [43]. This classifier has been used for land cover classification with satellite multispectral and hyperspectral images [45], [46].

These classifiers are referred to as non-deep learning methods.

F. *Deep Learning*

Artificial neural networks (ANNs), inspired by human learning process, employ a general structure of interconnected units to learn feature representation exclusively from data and minimize task specific and explicit rule based programming. This approach is to allow computers to learn from experience and to understand the world in terms of a concept hierarchy, with each concept being described in terms of its connection to simpler concepts. This approach eliminates the need for human operators to explicitly define all of the information that the computer requires by collecting information from experience. The idea hierarchy helps the computer to learn complex concepts by building them out of simpler ones. Deep learning models which are ANNs with more than two hidden layers are able to learn feature representations from data instead of manual feature extraction based on human experience [38].

Convolutional neural networks (CNN) are a particular type of deep learning architecture, which is currently a very successful method used in image classification and many other types of imaging applications. It is based on mimicry of the hierarchy architecture of the visual cortex present in mammals. It is composed of a set of layers, input, output layers and multiple hidden layers in between as shown in Fig. 16. The main mathematical operation on which it is based is convolution, it also consists of activation functions known as RELU layers, as well as fully connected layers and normalization layers.

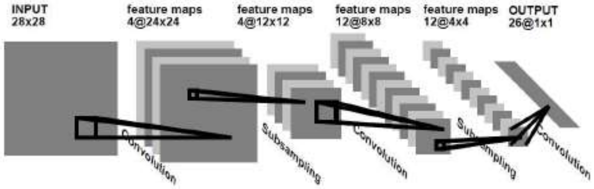


Fig. 16. Convolutional Neural Network for image processing, e.g. handwriting recognition. Source: [48].

Convolution is the process which operates over a small window of an image at a time. At each window it applies a kernel, a matrix multiplication and sum operation. Individual kernels are often referred to as filters. The result of applying the kernel to the entire image is a new image in which features are mapped. The role of the convolutional layer is to detect local conjunctions of characteristics from the previous layer; the function of the pooling layer is to combine similar characteristics semantically into one. This process is repeated in subsequent layers until an output layer describes at which class does the image relate to [48], [49], [50].

The particular problem in relation to image classification which we specifically apply for crop classification is called semantic segmentation. Segmentation is important for tasks relating to image analysis. Semantic segmentation refers to how each pixel of an image is associated with a class label (e.g. object, human, lane, sky, ocean, or car). Current applications for semantic segmentation are autonomous driving, medical imaging, industrial inspection and remote sensing. Being crop classification in this work being studied for segmenting quinoa crops from other type of land cover from a multispectral image we can consider it as a semantic segmentation problem.

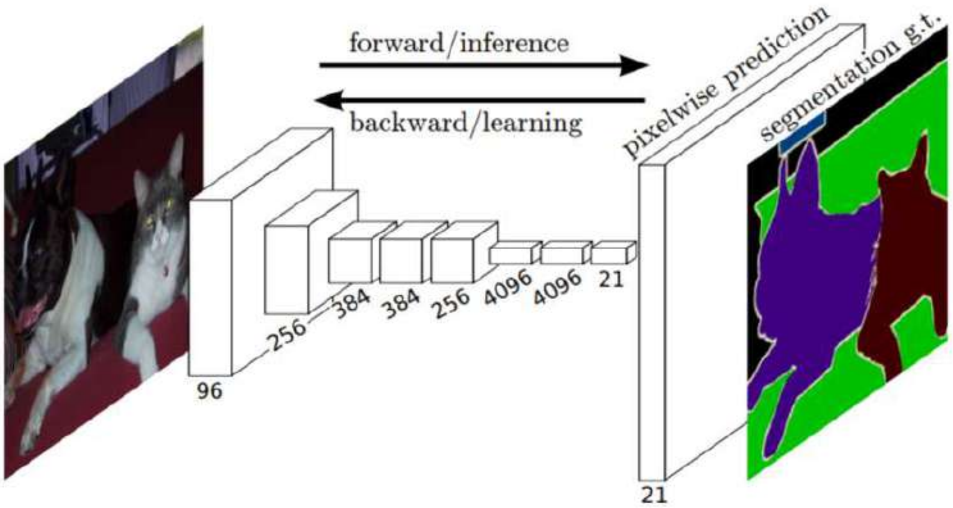


Fig. 17. Fully Convolutional Neural Network for semantic segmentation. Source: [41]

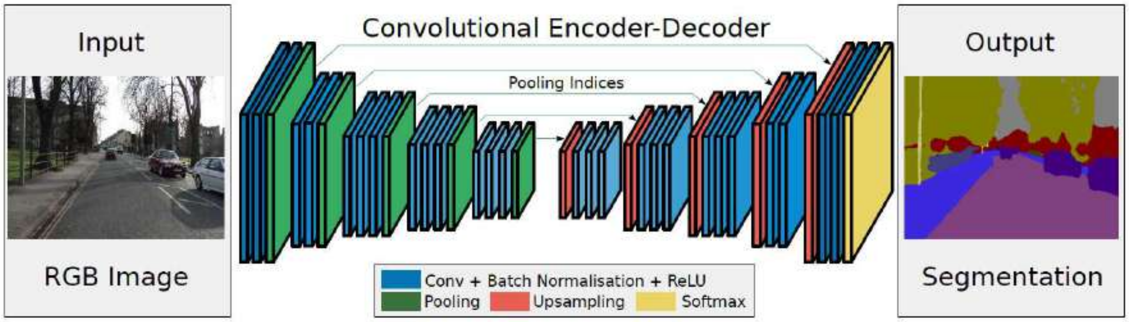


Fig. 18. SegNet Architecture. Source: [43].

Several types of CNN architectures exist that have been developed to be applied for semantic segmentation. In this study SegNet and U-Net have been evaluated for crop classification for which quinoa pixels is segmented from other type of land cover.

Deep learning algorithms have made major contributions to visual object recognition and classification in which a CNN is used consisting of convolution layers, ReLU activation functions, max-pooling layers and fully connected layers. Fully connected layers can cause loss of spatial information. The output layer is generally a Softmax function applied to the previous one-dimensional fully connected layer, that contains probability values for each class that determines the prediction output. The predicted class will be the one with the highest probability. In the case of semantic segmentation, an end to end approach is required for pixel-wise prediction. This means that the output layer is an image (2D) in which each pixel class is predicted. Fully Convolutional Networks (FCN) can accomplish dense predictions without any fully connected layers as shown in Fig. 17. To preserve spatial information, upsample and deconvolution layers are required. FCNs use an encoder-decoder architecture in which the encoder gradually reduces the spatial dimension with pooling layers and the decoder gradually recovers the object details and spatial dimension with upsampling operators [51]. Two FCN architectures used for semantic segmentation are SegNet and U-Net.

The SegNet architecture is a fully convolutional neural network for semantic pixel-wise segmentation having an encoder-decoder scheme. The encoder is similar to the 13 convolutional layers present in the VGG16 network [52]. The key component of SegNet is a decoder network consisting of a hierarchy of one decoder corresponding to each encoder. Of these, the appropriate decoders use the max-pooling indices received from the corresponding encoder to perform non-linear upsampling of their input feature maps as shown in Fig. 18. The key impetus behind SegNet was the need to create an efficient architecture for understanding road and indoor scenes [52]. Despite this, it has been applied to other types of semantic segmentation including remote sensing for weed classification [53], crop row detection [54] and rice and corn classification [55].

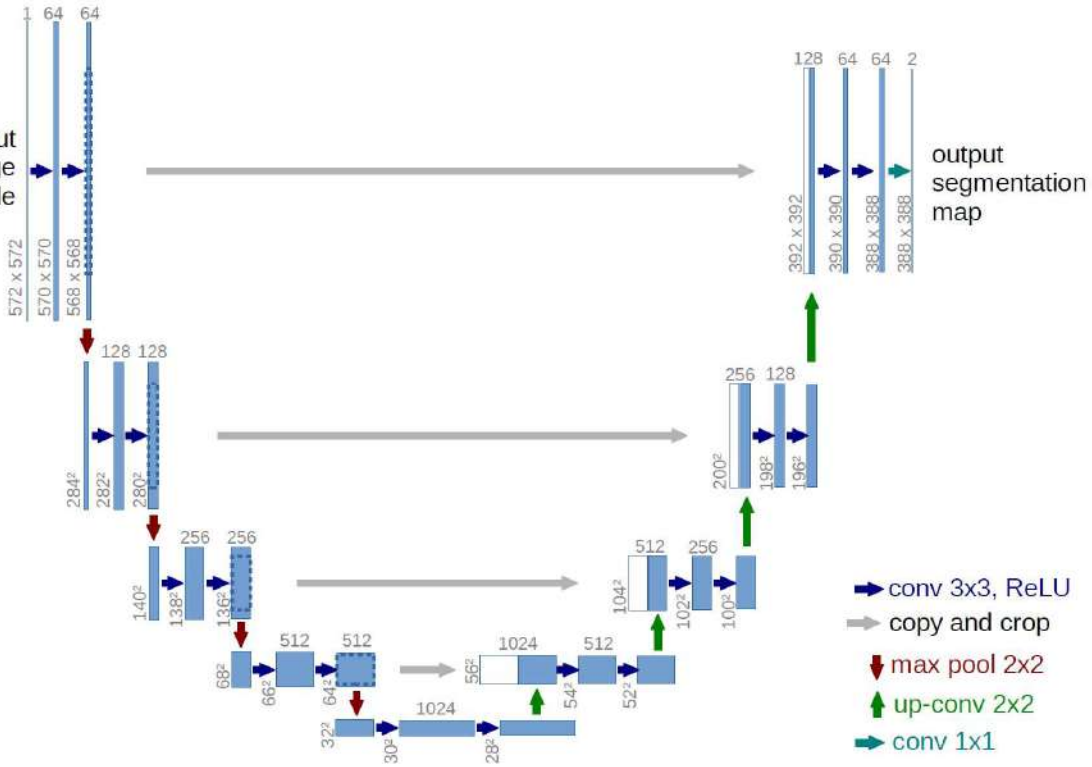


Fig. 19. U-Net Architecture. Source: [47].

The U-Net architecture is another FCN network for semantic segmentation. It involves an encoding path (contracting) to capture context coupled with a decoding path (expanding) that gives it a "U" form as shown in Fig. 19. The contracting path consists of convolution layers, ReLU activation functions and max-pooling layers which generate feature maps but gradually eliminates spatial information. The expanding path creates a high resolution segmentation map, it consists of a sequence of up-convolutions and concatenation with the corresponding high resolution features from the contracting path. The output segmentation map is generated and will have as many channels as classes needed for prediction [56]. U-Net is being used for many kinds of applications including biomedical image segmentation [56], cellular segmentation [57], road extraction [58], crowd counting [55] and saliency detection [60]. With respect to agriculture applications of the U-Net architecture, it is being used for rice mapping [61], [62], wheat yellow rust monitoring [63], plant disease recognition [64], and cropland parcel extraction [65].

IV. Experimental results

Experiments were performed using the data selected from Table I as was described as regions HYC4 and HYC5 to form a training and testing datasets respectively. The training dataset shown in Table II along with the label information were trained in order to get classification models of the methods described previously. Matlab version 2019b software was used to train models of the machine learning methods described. It is important to distinguish between non deep machine learning and deep learning techniques. The methods need to be distinguished because of the different algorithms involved and how the different Matlab tools are used to obtain models by means of training. A computer with a NVIDIA GeForce GTX 1070, 8GB RAM graphical processing unit (GPU), Intel Core i7-8700 @3.2 GHz processor, and 32 GB of RAM was used.

A. *Non Deep Learning Methods*

The Matlab Classification Learner App (CLA) [66] can train and classify data using various algorithms and compare the results in the same environment. To use it, datasets and ground truth labeled information must be adequately prepared. It was used to train models for the following classifiers: decision trees, discriminant analysis, support vector machines, K nearest neighbors, and ensemble classification. For the CLA process a validation scheme to avoid overfitting was used which included the use of part of the training dataset as validation data, in order to obtain a model for each classifier and its corresponding accuracy. The CLA provides an efficient way to decide which model has the best predictions in order to decide which non deep learning method to use for a specific dataset. Model accuracies are shown in Table III and as it shows best accuracies for the Support Vector Machine (95.4%) and Random Forest (94.7%) classifiers. These models were selected for testing.

TABLE II
Training Dataset HYC4
Number of pixels

Data	Date	Non Vegetation	Other Vegetation	Quinoa
1	27-12-2018	2929509	159406	664265
2	04-01-2019	3923729	1251091	593028
3	25-01-2019	1365821	3330347	203280
4	02-02-2019	1339181	1202338	236851
5	16-02-2019	1659807	3594423	268550
6	28-02-2019	1628694	2588546	135650
	Total	12846741	12126151	2101624

TABLE III
NON DEEP LEARNING MODEL ACCURACY

Classifier	Accuracy
Decision tree	91.2%
Discriminant Analysis	91.8%
Support Vector Machines	**95.4%**
K-Nearest Neighbor	94.5%
Random Forest	**94.7%**
Adaboost	91.2%

B. *Deep Learning Methods*

Deep learning architectures SegNet and U-Net were trained using the same dataset as the previous classifiers in Matlab. The training dataset was divided in smaller images commonly known as patches. The size of the square patches PxP pixels was chosen in a manner to describe small groups of quinoa plants, which depends on the resolution of the image data. Two group patch sizes were considered: 8x8, 12x12 and 16x16 for SegNet and 8x8 and 16x16 for U-Net. Because of the multispectral images used in this study, the patch image has dimensions of PxPx5, considering the 5 channels of different wavelengths. Data augmentation was used in order to increase the number of patches needed for training. Reflection and rotation of patches were used for this task obtaining a total of 2 million patches for training. Specific Matlab functions were used for extracting patches from the dataset, and augmenting patches in a random manner. For this Matlab's imagedatastore object is particularly convenient since it manages all the needed data in a simple and efficient way, providing the necessary organization for the subsequent stage of training the networks.

Matlab commands segnetLayers and unetLayers were used to generate both networks. The SegNet network generated is composed of 31 layers as shown in Fig. 20. The input layer is PxPx5 followed by encoder and decoder paths. Each encoder and decoder path has convolution layers and ReLU activation layers as well as maxpooling and max unpooling layers. Details of all layers including hyperparameters can be seen on Appendix 1. For this network values of P used were 8, 12 and 16.

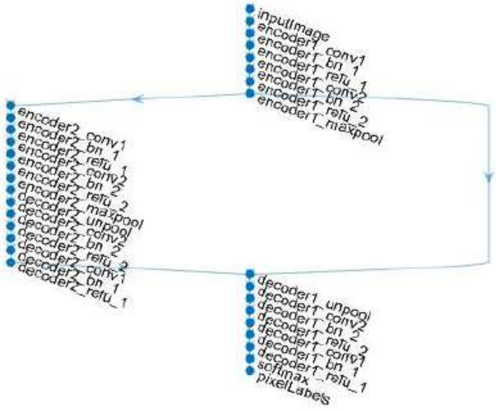


Fig. 20. SegNet architecture generated in Matlab Each blue dot represents a specific layer. The input layer (inputImage) is on top and otuput layer (pixelLabels) is in the bottom.

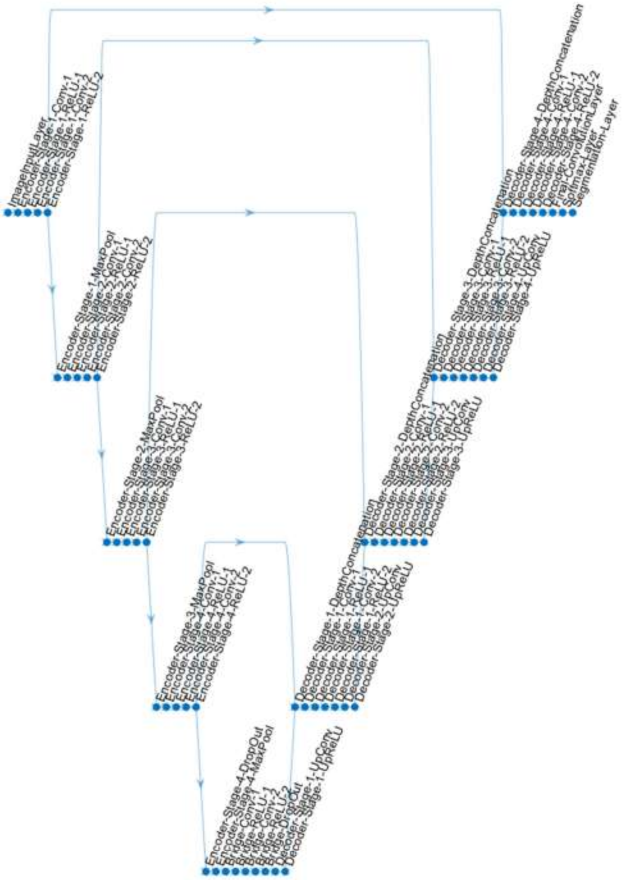


Fig. 21. U-Net architecture generated in Matlab Each blue dot represents a specific layer. The input layer (InputImageLayer) is on the left side and the otuput layer (Segmentation-Layer) is on the right side.

The U-Net architecture was generated considering encoder depths of 3 and 4. Each depth meaning that the network will have 46 and 51 layers respectively. Similar to the SegNet network encoder and decoder paths use convolution, ReLU, maxpooling and max unpooling layers as shown in Fig. 21 for encoder depth of 4. Appendix 1 shows layer details for a U-Net architecture for P equal to 16, and depth of 4.

C. *Training Deep Learning Networks*

Matlab was used to train the six convolutional neural networks generated. Validation was obtained using 20% of the training dataset. Among training options 8 epochs were considered in which 12500 iterations were used to update the internal model parameters. Matlab code used to train the classifier models are shown in Appendix B1 and B2. Training progress of the six models are shown in Appendix C. After training, a network model is obtained for each of the networks and validation is performed in order to avoid overfitting during training. Three trained network models were obtained for each SegNet and U-Net architectures. The six network models obtained with validation accuracies are shown in Table IV. It is worthwhile noting that the best accuracies occur for image sizes of 8 by 8 pixels, this could be because of the fact that it best describes individual plants. Image patches of this size seem to take advantage of its resolution.

D. *Classification Results and Comparison*

For numerical assessment of the trained models the Testing Dataset extracted from the HYC5 region was used.

TABLE IV
DEEP LEARNING MODEL VALIDATION ACCURACY

Classifier model	Validation Accuracy
SegNet8	98.39%
SegNet12	95.46%
SegNet16	94.53%
U_Net8	98.78%
U_Nett16_3	96.86%
U_Net16_4	97.58%

Pixel numbers of the six multispectral images are shown in Table V. Each image of the testing dataset was classified using the deep and non-deep learning models using Matlab. Qualitative comparison of the classification output images with ground truth image are shown in Fig. 22, where a group of selected close-up results can be seen. The full qualitative results of all the classified images are shown in Appendix D. As can be clearly noticed, deep learning techniques have superior performance than non-deep learning ones. Many other type of vegetation pixels are considered as quinoa pixels. Among the deep learning techniques U_Net16_4 can be viewed as the best classifier.

E. Numerical Evaluation Results

For semantic segmentation evaluation and subsequent numerical comparison there are several types of metrics that can be considered. The following metrics were used in this work: global accuracy (GA), mean accuracy (MA), mean intersection over union (mIoU), and weighted intersection over union (wIoU). Global accuracy is the ratio of correctly classified pixels, regardless of class, to the total number of pixels as show in equation 3. Mean accuracy is the ratio of correctly classified pixels in each class to total pixels, averaged over all classes (equation 4). These types of metrics are not necessarily well suited for situations where class pixels are extremely imbalanced.

$$GA = \frac{\sum_i n_{ii}}{\sum_i t_i} \quad (3)$$

$$MA = \frac{1}{n_{cl}} \sum_i \frac{n_{ii}}{t_i} \quad (4)$$

Where:

n_{ii} = number of pixels of class i predicted to belong to class i.

t_i = total number of pixels of class i.

n_{cl} = total number of classes.

For situations where there are different number of pixels for each class involved a much suited metric is based on the intersection over union (IoU) index. Also called the Jaccard Index (JI) it is defined as the area of intersection between the predicted segmentation map A and the ground truth map B, divided by the area union between the two maps. The formula is shown in equation 5.

$$IoU = J(A,B) = \frac{|A \cap B|}{|A \cup B|} \quad (5)$$

For computing the IoU in terms of pixels for a particular class, the IoU index can also be calculated as:

$$IoU = \frac{TP}{TP+FP+FN} \quad (6)$$

TABLE V
TESTING DATASET HYC5 – GROUND TRUTH
NUMBER OF PIXELS

Data	Date	Non Vegetation	Other Vegetation	Quinoa
HYC51	27-12-2018	9678093	1278911	1190432
HYC52	04-01-2019	2590172	1081423	830120
HYC53	25-01-2019	750116	879587	323836
HYC54	02-02-2019	1555599	1173204	388731
HYC55	16-02-2019	1549353	922420	385599
HYC56	28-02-2019	1748899	841832	131625
	Total	17872232	6177377	3250343

Where:

TP = number of true positive pixels of a class.

FP = number of false positive pixels.

FN = number of false negative pixels.

Mean IoU is the average intersection over union (IoU) of all classes and it is computed by equation 7.

$$mIoU = \frac{1}{n_{cl}} \sum_i IoU_i \quad (7)$$

Weighed IoU is the average IoU of all classes, weighted by the number of pixels in the class. This metric can be considered as the best in cases where there is a wide variety of pixel class numbers. Table V shows this particular feature in the testing dataset for which it is considered as the preferred metric.

$$wIoU = \frac{\sum_i t_i . IoU_i}{\sum_i t_i} \quad (8)$$

Appendix B3 shows Matlab code for numerical evaluation and results are shown in Table V1 of all classifiers for the testing dataset. It shows that the Unet classifier achieves a better wIoU than Segnet and the other non-deep learning models. It also shows better results in the first three phenological stages which can be attributed to high reflectance of vegetation due to the fact that in these stages the plants are more vigorous. Among the three Unet models, best results are obtained for the Unet16 architecture which has more layers than Unet8, this higher complexity can be attributed to a better generalization for the classification task.

V. CONCLUSION

In this study machine learning models were applied to classify quinoa crops from multispectral aerial images. Crops from a typical quinoa growing region of Puno in Peru were considered as part of the experimental setup for different phenological stages. Deep learning models Unet and Segnet outperformed Decision tree, Discriminant Analysis, Support Vector Machines, K-Nearest Neighbor, Random Forest and Adaboost classifiers allowing discrimination of quinoa crops from other types of land cover. An accuracy between 80 to 95% was obtained for the Unet architecture model. Higher accuracies were obtained for the initial phenological stages of the crop. This can be explained because higher volume of vegetation biomass is abundant during these early stages compared to the final ones. Higher vegetation biomass has a higher reflectance in the green, rededge and near infrared optical bandwidths which will contribute to a better classification of the quinoa crop. A classification method for mapping quinoa crops has been obtained which can be used to as a previous stage for predicting yields.

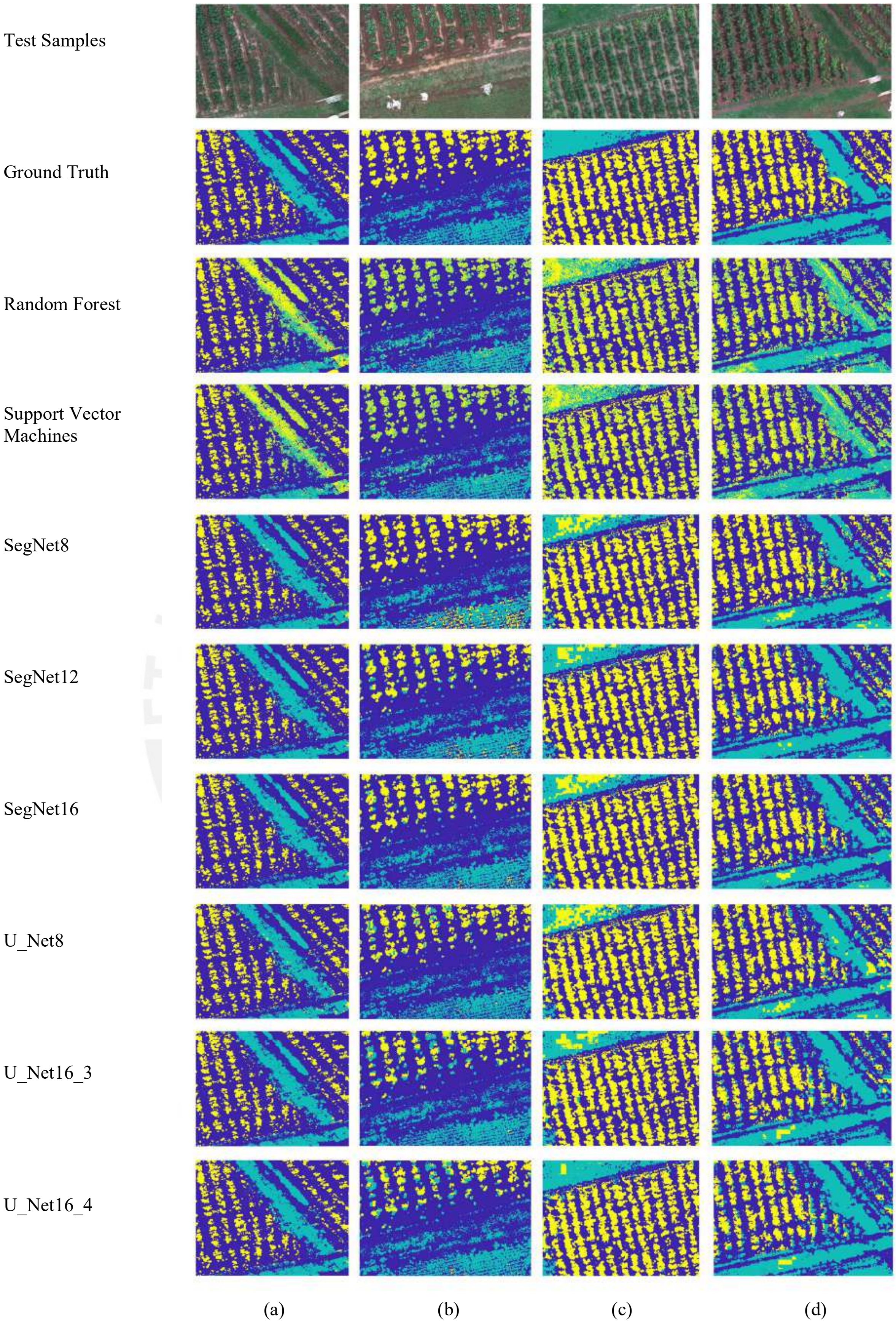


Fig. 22. Close-up results on HYC5 sample images.

TABLE V1
CLASSIFICATION ACCURACY COMPARISON

	Classifier	HYC51				HYC52				HYC53			
		GA %	MA %	mIoU %	wIoU %	GA %	MA %	mIoU %	wIoU %	GA %	MA %	mIoU %	wIoU %
1	SVM	66.12	64.58	39.68	58.68	92.61	88.32	80.22	87.27	68.48	72.68	54.60	58.65
2	RF	92.15	75.42	61.64	87.86	92.04	87.47	79.04	86.36	69.46	73.58	55.44	59.55
3	SegNet8	95.83	88.10	79.56	92.59	95.06	93.54	88.25	90.67	83.23	84.09	69.39	73.92
4	SegNet12	96.33	89.58	81.44	93.48	95.89	94.18	89.72	92.23	88.90	88.50	77.35	81.49
5	SegNet16	96.89	91.02	84.27	94.32	95.88	94.25	89.73	92.19	90.24	90.41	79.68	83.34
6	U_Net8	95.43	86.27	76.70	92.07	94.93	92.31	87.40	90.42	81.12	83.43	67.08	71.35
7	U_Nett16_3	97.53	93.36	87.34	95.44	95.45	92.90	87.94	91.48	94.38	92.64	86.88	89.61
8	U_Net16_4	97.25	92.61	85.93	94.99	94.94	91.88	86.50	90.56	94.80	92.49	87.63	90.25

	Classifier	HYC54				HYC55				HYC56			
		GA %	MA %	mIoU %	wIoU %	GA %	MA %	mIoU %	wIoU %	GA %	MA %	mIoU %	wIoU %
1	SVM	77.64	73.93	57.96	70.24	82.47	79.49	62.95	75.03	67.27	60.97	37.56	63.64
2	RF	78.93	75.00	59.41	71.62	82.57	79.77	63.11	75.10	68.46	61.22	38.59	64.91
3	SegNet8	90.81	87.23	76.99	84.33	86.51	83.52	69.44	78.68	66.63	57.28	35.63	61.55
4	SegNet12	91.10	83.38	75.55	84.52	88.77	85.17	73.01	81.88	69.61	50.77	38.80	66.04
5	SegNet16	90.83	85.40	76.19	84.37	89.07	86.07	73.73	82.19	69.04	47.75	38.77	65.26
6	U_Net8	88.29	84.38	72.17	81.15	85.43	81.83	67.29	77.48	67.18	57.62	36.09	62.89
7	U_Nett16_3	91.05	83.01	75.08	84.62	88.64	84.67	72.51	81.90	78.03	61.30	47.57	73.55
8	U_Net16_4	92.19	82.83	76.50	85.89	90.03	84.94	74.56	83.54	86.35	63.71	55.73	81.00

VI. BUSINESS PLAN

World production of quinoa is led primarily by Peru and Bolivia, with between 83% (year 2015) to 67% (year 2016) of the total volume produced. In both countries, this activity involves more than 150,000 families of small producers (81,643 families in Peru - INEI IV Agricultural Census 2012-, and 70,000 families in Bolivia -INE I Agricultural Census 2013). The production of this crop is mainly organic with low production averages that vary from 0.3 Ton/Ha (Bolivia) to 0.8 Ton/Ha (Peru). As a consequence of the scarce use of modern technologies, the increase in pests that are harmful to the crop, as well as the increasingly unpredictable variability of climatic conditions, situations that contribute to family farmers in both regions continuing to present poverty levels of 42.9% and extreme poverty of 21.3% on average.

This work is part of a project that seeks to generate a management system to monitor the status of crops for quinoa producers who develop their work in the context of family farming to achieve certification as an organic quinoa product and export to foreign markets.

Drone technology, remote sensing, and computing resources will be used to have accurate information in the form of georeferenced maps that will be available in the Internet so producers can monitor their processes and guarantee traceability. This will be achieved with the development of management modules for users, producers, consultants, technicians and administrative staff of the association and buyers. A commercial product application is expected. A market study was carried out with the main objective to detail the current market for management and monitoring systems of organic quinoa production, as well as the impact that the development of the project could have on family farming in Puno, Peru.

For this purpose, the study has been carried out in the area of the Puno region; In this region, there are 19 organizations that involve small family producers of organic quinoa. Of these, there are 5 organizations that are currently operating as organic operators, that is, they are working every year in the organic certification process of their producers. For this reason, the present study has been carried out with all of these organizations, actors and stakeholders being the directors and team of professionals of the 5 existing organic producer organizations. Among the main results found, we can mention the following:

Regarding the availability and use of systems that support the organic certification process and/or the management of the production chain; it was found that more than 50% of producers and professionals indicated that the most used system was Excel (60%), followed by desktop programs (27%), and the web (3%). These applications are used to manage the organic certification process (42%), followed by the production management (25%), warehouse processes (17%) and marketing (17%). It is precisely due to the need to have management systems or computer programs that help in the production chain, that both producers and leaders as well as professionals and technicians interviewed showed interest in a new program that would help them manage in with organic certification (87%), crop production systems (73%), marketing (63%), and warehouse management (47%) [67]. For these reasons is that the work done in this research will contribute to the automation of data relevant to crop management.

VII. Acknowledgment

This research is supported by the Innovate Peru, Ministry of Production under Grant Number 106-INNOVATEPERU-IDIBIO-2018 through the Unmanned Aircraft Systems Laboratory at PUCP, Lima Peru. The author would like to also acknowledge the Association of Agricultural Producers of Vizallani1 for accessing their crop land, specially to farmer Daniel Cárdenas whose land was used in this study.

References

[1] K. Murphy and G. J. Matanguihan, Eds., *Quinoa: improvement and sustainable production*. Hoboken, New Jersey: John Wiley & Sons, Inc, 2015.

[2] N. S. Bedoya-Perales, G. Pumi, E. Talamini, and A. D. Padula, "The quinoa boom in Peru: Will land competition threaten sustainability in one of the cradles of agriculture?" *Land Use Policy*, vol. 79, pp. 475–480, Dec. 2018.

[3] P. C. Doraiswamy, A. J. Stern, and B. Akhmedov, "Crop classification in the U.S. Corn Belt using MODIS imagery," in *2007 IEEE International Geoscience and Remote Sensing Symposium*, 2007, pp. 809–812, doi: 10.1109/IGARSS.2007.4422920.

[4] J. You, Z. Pei, and D. Wang, "Crop Mapping of Complex Agricultural Landscapes Based on Discriminant Space," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 7, no. 11, pp. 4356–4367, Nov. 2014, doi: 10.1109/JSTARS.2014.2320156.

[5] S. Feng, J. Zhao, T. Liu, H. Zhang, Z. Zhang, and X. Guo, "Crop Type Identification and Mapping Using Machine Learning Algorithms and Sentinel-2 Time Series Data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 12, no. 9, pp. 3295–3306, Sep. 2019, doi: 10.1109/JSTARS.2019.2922469.

[6] A. Khaliq, L. Peroni, and M. Chiaberge, "Land cover and crop classification using multitemporal sentinel-2 images based on crops phenological cycle," in *2018 IEEE Workshop on Environmental, Energy, and Structural Monitoring Systems (EESMS)*, 2018, pp. 1–5, doi: 10.1109/EESMS.2018.8405830.

[7] H. Zhang *et al.*, "Image Classification Using RapidEye Data: Integration of Spectral and Textual Features in a Random Forest Classifier," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 12, pp. 5334–5349, Dec. 2017, doi: 10.1109/JSTARS.2017.2774807.

[8] Y. Cai *et al.*, "A high-performance and in-season classification system of field-level crop types using time-series Landsat data and a machine learning approach," *Remote Sensing of Environment*, vol. 210, pp. 35–47, Jun. 2018, doi: 10.1016/j.rse.2018.02.045.

[9] O. Hall, S. Dahlin, H. Marstorp, M. F. Archila Bustos, I. Öborn, and M. Jirström, "Classification of Maize in Complex Smallholder Farming Systems Using UAV Imagery," *Drones*, vol. 2, no. 3, p. 22, Sep. 2018, doi: 10.3390/drones2030022.

[10] Z. Du, J. Yang, C. Ou, and T. Zhang, "Smallholder Crop Area Mapped with a Semantic Segmentation Deep Learning Method," *Remote Sensing*, vol. 11, no. 7, p. 888, Jan. 2019, doi: 10.3390/rs11070888.

[11] A. G. T. Schut, P. C. S. Traore, X. Blaes, and R. A. de By, "Assessing yield and fertilizer response in heterogeneous smallholder fields with UAVs and satellites," *Field Crops Research*, vol. 221, pp. 98–107, May 2018, doi: 10.1016/j.fcr.2018.02.018.

[12] A. S. Terliksiz and D. T. Altýlar, "Use of Deep Neural Networks for Crop Yield Prediction: A Case Study of Soybean Yield in Lauderdale County, Alabama, USA," in *2019 8th International Conference on Agro-Geoinformatics (Agro-Geoinformatics)*, 2019, pp. 1–4, doi: 10.1109/Agro-Geoinformatics.2019.8820257.

[13] B. Xie, H. K. Zhang, and J. Xue, "Deep Convolutional Neural Network for Mapping Smallholder Agriculture Using High Spatial Resolution Satellite Image," *Sensors*, vol. 19, no. 10, p. 2398, Jan. 2019, doi: 10.3390/s19102398.

[14] G.-H. Kwak and N.-W. Park, "Impact of Texture Information on Crop Classification with Machine Learning and UAV Images," *Applied Sciences*, vol. 9, no. 4, p. 643, Jan. 2019, doi: 10.3390/app9040643.

[15] Y. Kim, G.-H. Kwak, K.-D. Lee, S.-I. Na, C.-W. Park, and N.-W. Park, "Performance Evaluation of Machine Learning and Deep Learning Algorithms in Crop Classification: Impact of Hyper-parameters and Training Sample Size," *Korean Journal of Remote Sensing*, vol. 34, no. 5, pp. 811–827, 2018, doi: 10.7780/kjrs.2018.34.5.9.

[16] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, "Deep Learning Classification of Land Cover and Crop Types Using Remote Sensing Data," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 5, pp. 778–782, May 2017, doi: 10.1109/LGRS.2017.2681128.

[17] R. Khatami, G. Mountrakis, and S. V. Stehman, "A meta-analysis of remote sensing research on supervised pixel-based land-cover image classification processes: General guidelines for practitioners and future research," *Remote Sensing of Environment*, vol. 177, pp. 89–100, May 2016, doi: 10.1016/j.rse.2016.02.028.

[18] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3 edition. Upper Saddle River, N.J: Pearson, 2007.

[19] R. F. Evert and S. E. Eichhorn, *Raven Biology of Plants*, Eighth edition. W. H. Freeman, 2012.

[20] E. Solomon, C. Martin, D. W. Martin, and L. R. Berg, *Biology*, 11 edition. Australia ; Boston, MA: Cengage Learning, 2018.

[21] J. Whitmarsh and Govindjee, "The Photosynthetic Process." [Online]. Available: http://www.life.illinois.edu/govindjee/paper/gov.html. [Accessed: 12-Feb-2020].

[22] T. Lillesand, R. W. Kiefer, y J. Chipman, "Remote Sensing and Image Interpretation", 7th edition. Hoboken, NJ: Wiley, 2015.

[23] J. Artiola, I. L. Pepper, and M. L. Brusseau, *Environmental Monitoring and Characterization*, 1 edition. Amsterdam ; Boston: Academic Press, 2004.

[24] D. N. H. Horler, M. Dockray, and J. Barber, "The red edge of plant leaf reflectance," *International Journal of Remote Sensing*, vol. 4, no. 2, pp. 273–288, Jan. 1983, doi: 10.1080/01431168308948546.

[25] E. B. Knipling, "Physical and physiological basis for the reflectance of visible and near-infrared radiation from vegetation," *Remote Sensing of Environment*, vol. 1, no. 3, pp. 155–159, Jun. 1970, doi: 10.1016/S0034-4257(70)80021-9

[26] "RedEdge User Manual (PDF Download)," *MicaSense Knowledge Base*. [Online]. Available: http://support.micasense.com/hc/en-us/articles/215261448-RedEdge-User-Manual-PDF-Download-. [Accessed: 17-Feb-2020].

[27] N. Pettorelli, "The Normalized Difference Vegetation Index". Oxford University Press,Oxford, 2013.

[28] Breiman, L., J. H. Friedman, R. A. Olshen, and C. J. Stone. "Classification and Regression Trees". Chapman & Hall, Boca Raton, 1993.

[29] M. Pal and P. M. Mather, "An assessment of the effectiveness of decision tree methods for land cover classification," *Remote Sensing of Environment*, vol. 86, no. 4, pp. 554–565, Aug. 2003, doi: 10.1016/S0034-4257(03)00132-9.

[30] Land cover change assessment using decision trees, support vector machines and maximum likelihood classification algorithms, *International Journal of Applied Earth Observation and Geoinformation*, vol. 12, pp. S27-S31, feb. 2010, doi: 10.1016/j.jag.2009.11.002

[31] "Discriminant Analysis Classification - MATLAB & Simulink." [Online]. Available: https://www.mathworks.com/help/stats/discriminant-analysis.html. [Accessed: 09-Aug-2019].

[32] Fisher, R. A. "The Use of Multiple Measurements in Taxonomic Problems". Annals of Eugenics, Vol. 7, pp. 179–188, 1936. Available at http://digital.library.adelaide.edu.au/dspace/handle/2440/15227.

[33] T. He y K. Zhao, "Multispectral Remote Sensing Land Use Classification Based on RBF Neural Network with Parameters Optimized by Genetic Algorithm", en *2018 International Conference on Sensor Networks and Signal Processing (SNSP)*, oct. 2018, pp. 118-123. doi: 10.1109/SNSP.2018.00031.

[34] Y. Gu, Q. Wang, P. Liu, y D. Zuo, "Linear discriminant multiple kernel learning for multispectral image classification", en *2014 IEEE International Conference on Image Processing (ICIP)*, oct. 2014, pp. 5052-5056. doi: 10.1109/ICIP.2014.7026023.

[35] C.-H. Li, H.-S. Chu, B.-C. Kuo, y C.-T. Lin, "Hyperspectral image classification using spectral and spatial information based linear discriminant analysis", en *2011 IEEE International Geoscience and Remote Sensing Symposium*, jul. 2011, pp. 1716-1719. doi: 10.1109/IGARSS.2011.6049566.

[36] M. A. Hearst, S. T. Dumais, E. Osuna, J. Platt and B. Scholkopf, "Support vector machines," in *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18-28, July-Aug. 1998.

[37] A. Ben-Hur and J. Weston, "A User's Guide to Support Vector Machines," in *Data Mining Techniques for the Life Sciences*, O. Carugo and F. Eisenhaber, Eds. Totowa, NJ: Humana Press, 2010, pp. 223–239.

[38] Marco Cococcioni (2021). SVM Basic Support Vector Machines using quadprog (https://www.mathworks.com/matlabcentral/fileexchange/68218-svm-basic-support-vector-machines-using-quadprog), MATLAB Central File Exchange. Retrieved September 12, 2021.

[39] Y. Qian, W. Zhou, J. Yan, W. Li, and L. Han, "Comparing Machine Learning Classifiers for Object-Based Land Cover Classification Using Very High Resolution Imagery," *Remote Sensing*, vol. 7, no. 1, pp. 153–168, Jan. 2015, doi: 10.3390/rs70100153.

[40] S. Bharathi et al., "Automatic land use/land cover classification using texture and data mining classifier," *2013 IEEE International Conference of IEEE Region 10* (TENCON 2013), 2013, pp. 1-4, doi: 10.1109/TENCON.2013.6718977.

[41] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*, 1 edition. New York, NY, USA: Cambridge University Press, 2014.

[42] A. Géron, Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems, 1 edition. O'Reilly Media, 2017.

[43] S. Marsland, *Machine Learning: An Algorithmic Perspective, Second Edition*. CRC Press, 2015.

[44] David Ferreira. "k-Nearest Neighbors (kNN)", 2019. Retrieved January 23, 2019. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/67018-k-nearest-neighbors-knn), MathWorks.

[45] J. C.-W. Chan y D. Paelinckx, "Evaluation of Random Forest and Adaboost tree-based ensemble classification and spectral band selection for ecotope mapping using airborne hyperspectral imagery", *Remote Sensing of Environment*, vol. 112, n.º 6, pp. 2999-3011, jun. 2008, doi: 10.1016/j.rse.2008.02.011.

[46] L. N. Eeti y K. M. Buddhiraju, "Comparison of AdaBoost.M2 and perspective based model ensemble in multispectral image classification", *2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, Jul. 2017, pp. 3684-3687. doi: 10.1109/IGARSS.2017.8127798.

[47] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, Massachusetts: The MIT Press, 2016.

[48] Y. Lecun and Y. Bengio, "Convolutional networks for images, speech, and time-series", *The handbook of brain theory and neural networks*, The MIT Press 1995.

[49] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: 10.1038/nature14539.

[50] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, Apr. 2017, doi: 10.1109/TPAMI.2016.2572683.

[51] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv:1409.1556 [cs]*, Apr. 2015, Accessed: May 02, 2020. [Online]. Available: http://arxiv.org/abs/1409.1556.

[52] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," *arXiv:1511.00561 [cs]*, Oct. 2016, Accessed: May 02, 2020. [Online]. Available: http://arxiv.org/abs/1511.00561.

[53] I. Sa *et al.*, "weedNet: Dense Semantic Weed Classification Using Multispectral Images and MAV for Smart Farming," *IEEE Robotics and Automation Letters*, vol. 3, no. 1, pp. 588–595, Jan. 2018, doi: 10.1109/LRA.2017.2774979.

[54] M. D. Bah, A. Hafiane, and R. Canals, "CRowNet: Deep Network for Crop Row Detection in UAV Images," *IEEE Access*, vol. 8, pp. 5189–5200, 2020, doi: 10.1109/ACCESS.2019.2960873.

[55] M. D. Yang, H. H. Tseng, Y. C. Hsu, and W. C. Tseng, "Real-time Crop Classification Using Edge Computing and Deep Learning," in *2020 IEEE 17th Annual Consumer Communications Networking Conference (CCNC)*, Jan. 2020, pp. 1–4, doi: 10.1109/CCNC46108.2020.9045498.

[56] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Cham, 2015, pp. 234–241, doi: 10.1007/978-3-319-24574-4_28.

[57] G. Yang *et al.*, "DAGAN: Deep De-Aliasing Generative Adversarial Networks for Fast Compressed Sensing MRI Reconstruction," *IEEE Transactions on Medical Imaging*, vol. 37, no. 6, pp. 1310–1321, Jun. 2018, doi: 10.1109/TMI.2017.2785879.

[58] Z. Zhang, Q. Liu, and Y. Wang, "Road Extraction by Deep Residual U-Net," *IEEE Geoscience and Remote Sensing Letters*, vol. 15, no. 5, pp. 749–753, May 2018, doi: 10.1109/LGRS.2018.2802944.

[59] Z. Shen, Y. Xu, B. Ni, M. Wang, J. Hu, and X. Yang, "Crowd Counting via Adversarial Cross-Scale Consistency Pursuit," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 5245–5254, doi: 10.1109/CVPR.2018.00550.

[60] N. Liu, J. Han, and M.-H. Yang, "PiCANet: Learning Pixel-Wise Contextual Attention for Saliency Detection," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 3089–3098, doi: 10.1109/CVPR.2018.00326.

[61] S. Wei, H. Zhang, C. Wang, L. Xu, F. Wu, and B. Zhang, "Large-Scale Rice Mapping of Thailand using Sentinel-1 Multi-Temporal SAR Data," in *2019 SAR in Big Data Era (BIGSARDATA)*, Aug. 2019, pp. 1–4, doi: 10.1109/BIGSARDATA.2019.8858438.

[62] M. Gargiulo, D. A. G. Dell'Aglio, A. Iodice, D. Riccio, and G. Ruello, "Semantic Segmentation using Deep Learning: A case of study in Albufera Park, Valencia," in *2019 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor)*, Oct. 2019, pp. 134–138, doi: 10.1109/MetroAgriFor.2019.8909243.

[63] J. Su *et al.*, "Aerial Visual Perception in Smart Farming: Field Study of Wheat Yellow Rust Monitoring," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2020, doi: 10.1109/TII.2020.2979237.

[64] S. Huang, W. Liu, F. Qi, and K. Yang, "Development and Validation of a Deep Learning Algorithm for the Recognition of Plant Disease," in *2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, Aug. 2019, pp. 1951–1957, doi: 10.1109/HPCC/SmartCity/DSS.2019.00269.

[65] L. Xia, J. Luo, Y. Sun, and H. Yang, "Deep Extraction of Cropland Parcels from Very High-Resolution Remotely Sensed Imagery," in *2018 7th International Conference on Agro-geoinformatics (Agro-geoinformatics)*, Aug. 2018, pp. 1–5, doi: 10.1109/Agro-Geoinformatics.2018.8476002.

[66] "Train models to classify data using supervised machine learning - MATLAB." https://www.mathworks.com/help/stats/classificationlearner-app.html (accessed May 04, 2020).

[67] P. Huarahuara Mamani, "Resultados del Estudio de Mercado Realizado para el "Sistema de Gestión y Monitoreo de la Producción de Quinua Orgánica"", PUCP, Lima Peru, 2019. [Online]. Available: https://drive.google.com/file/d/1QOPJtf0GW1a74u1vbQAZ-p-W4VWjop1L/view?usp=sharing

APPENDIX A

The following tables show details of the SegNet and U-Net architectures.

TABLE A.1
SEGNET ARCHITECTURE FOR P=8, P=12, P=16

N	Layer	Type	Hyperparameters
1	'inputImage'	Image Input	PxPx5 images with 'zerocenter' normalization
2	'encoder1_conv1'	Convolution	64 3x3x5 convolutions with stride [1 1] and padding [1 1 1 1]
3	'encoder1_bn_1'	Batch Normalization	Batch normalization
4	'encoder1_relu_1'	ReLU	ReLU
5	'encoder1_conv2'	Convolution	64 3x3x64 convolutions with stride [1 1] and padding [1 1 1 1]
6	'encoder1_bn_2'	Batch Normalization	Batch normalization
7	'encoder1_relu_2'	ReLU	ReLU
8	'encoder1_maxpool'	Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0 0 0]
9	'encoder2_conv1'	Convolution	64 3x3x64 convolutions with stride [1 1] and padding [1 1 1 1]
10	'encoder2_bn_1'	Batch Normalization	Batch normalization
11	'encoder2_relu_1'	ReLU	ReLU
12	'encoder2_conv2'	Convolution	64 3x3x64 convolutions with stride [1 1] and padding [1 1 1 1]
13	'encoder2_bn_2'	Batch Normalization	Batch normalization
14	'encoder2_relu_2'	ReLU	ReLU
15	'encoder2_maxpool'	Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0 0 0]
16	'decoder2_unpool'	Max Unpooling	Max Unpooling
17	'decoder2_conv2'	Convolution	64 3x3x64 convolutions with stride [1 1] and padding [1 1 1 1]
18	'decoder2_bn_2'	Batch Normalization	Batch normalization
19	'decoder2_relu_2'	ReLU	ReLU
20	'decoder2_conv1'	Convolution	64 3x3x64 convolutions with stride [1 1] and padding [1 1 1 1]
21	'decoder2_bn_1'	Batch Normalization	Batch normalization
22	'decoder2_relu_1'	ReLU	ReLU
23	'decoder1_unpool'	Max Unpooling	Max Unpooling
24	'decoder1_conv2'	Convolution	64 3x3x64 convolutions with stride [1 1] and padding [1 1 1 1]
25	'decoder1_bn_2'	Batch Normalization	Batch normalization
26	'decoder1_relu_2'	ReLU	ReLU
27	'decoder1_conv1'	Convolution	3 3x3x64 convolutions with stride [1 1] and padding [1 1 1 1]
28	'decoder1_bn_1'	Batch Normalization	Batch normalization
29	'decoder1_relu_1'	ReLU	ReLU
30	'softmax'	Softmax	softmax
31	'pixelLabels'	Pixel Classification Layer	Cross-entropy loss

TABLE A.2

U-Net Architecture for P=8, P=16, encoder Depth = 3

N	Layer	Type	Hyperparameters
1	'ImageInputLayer'	Image Input	PxPx5 images with 'zerocenter' normalization
2	'Encoder-Stage-1-Conv-1'	Convolution	64 3x3 convolutions with stride [1 1] and padding 'same'
3	'Encoder-Stage-1-ReLU-1'	ReLU	ReLU
4	'Encoder-Stage-1-Conv-2'	Convolution	64 3x3 convolutions with stride [1 1] and padding 'same'
5	'Encoder-Stage-1-ReLU-2'	ReLU	ReLU
6	'Encoder-Stage-1-MaxPool'	Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0 0 0]
7	'Encoder-Stage-2-Conv-1'	Convolution	128 3x3 convolutions with stride [1 1] and padding 'same'
8	'Encoder-Stage-2-ReLU-1'	ReLU	ReLU
9	'Encoder-Stage-2-Conv-2'	Convolution	128 3x3 convolutions with stride [1 1] and padding 'same'
10	'Encoder-Stage-2-ReLU-2'	ReLU	ReLU
11	'Encoder-Stage-2-MaxPool'	Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0 0 0]
12	'Encoder-Stage-3-Conv-1'	Convolution	256 3x3 convolutions with stride [1 1] and padding 'same'
13	'Encoder-Stage-3-ReLU-1'	ReLU	ReLU
14	'Encoder-Stage-3-Conv-2'	Convolution	256 3x3 convolutions with stride [1 1] and padding 'same'
15	'Encoder-Stage-3-ReLU-2'	ReLU	ReLU
16	'Encoder-Stage-3-DropOut'	Dropout	50% dropout
17	'Encoder-Stage-3-MaxPool'	Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0 0 0]
18	'Bridge-Conv-1'	Convolution	512 3x3 convolutions with stride [1 1] and padding 'same'
19	'Bridge-ReLU-1'	ReLU	ReLU
20	'Bridge-Conv-2'	Convolution	512 3x3 convolutions with stride [1 1] and padding 'same'
21	'Bridge-ReLU-2'	ReLU	ReLU
22	'Bridge-DropOut'	Dropout	50% dropout
23	'Decoder-Stage-1-UpConv'	Transposed Convolution	256 2x2 transposed convolutions with stride [2 2] and cropping [0 0 0 0]
24	'Decoder-Stage-1-UpReLU'	ReLU	ReLU
25	'Decoder-Stage-1-DepthConcatenation'	Depth concatenation	Depth concatenation of 2 inputs
26	'Decoder-Stage-1-Conv-1'	Convolution	256 3x3 convolutions with stride [1 1] and padding 'same'
27	'Decoder-Stage-1-ReLU-1'	ReLU	ReLU
28	'Decoder-Stage-1-Conv-2'	Convolution	256 3x3 convolutions with stride [1 1] and padding 'same'
29	'Decoder-Stage-1-ReLU-2'	ReLU	ReLU
30	'Decoder-Stage-2-UpConv'	Transposed Convolution	128 2x2 transposed convolutions with stride [2 2] and cropping [0 0 0 0]
31	'Decoder-Stage-2-UpReLU'	ReLU	ReLU
32	'Decoder-Stage-2-DepthConcatenation'	Depth concatenation	Depth concatenation of 2 inputs
33	'Decoder-Stage-2-Conv-1'	Convolution	128 3x3 convolutions with stride [1 1] and padding 'same'
34	'Decoder-Stage-2-ReLU-1'	ReLU	ReLU
35	'Decoder-Stage-2-Conv-2'	Convolution	128 3x3 convolutions with stride [1 1] and padding 'same'
36	'Decoder-Stage-2-ReLU-2'	ReLU	ReLU
37	'Decoder-Stage-3-UpConv'	Transposed Convolution	64 2x2 transposed convolutions with stride [2 2] and cropping [0 0 0 0]
38	'Decoder-Stage-3-UpReLU'	ReLU	ReLU
39	'Decoder-Stage-3-DepthConcatenation'	Depth concatenation	Depth concatenation of 2 inputs
40	'Decoder-Stage-3-Conv-1'	Convolution	64 3x3 convolutions with stride [1 1] and padding 'same'
41	'Decoder-Stage-3-ReLU-1'	ReLU	ReLU
42	'Decoder-Stage-3-Conv-2'	Convolution	64 3x3 convolutions with stride [1 1] and padding 'same'
43	'Decoder-Stage-3-ReLU-2'	ReLU	ReLU
44	'Final-ConvolutionLayer'	Convolution	3 1x1 convolutions with stride [1 1] and padding 'same'
45	'Softmax-Layer'	Softmax	softmax
46	'Segmentation-Layer'	Pixel Classification Layer	Cross-entropy loss

TABLE A.3
U-Net Architecture for P=16, encoder Depth = 3

N	Layer	Type	Hyperparameters
1	'ImageInputLayer'	Image Input	16x16x5 images with 'zerocenter' normalization
2	'Encoder-Stage-1-Conv-1'	Convolution	64 3x3x5 convolutions with stride [1 1] and padding 'same'
3	'Encoder-Stage-1-ReLU-1'	ReLU	ReLU
4	'Encoder-Stage-1-Conv-2'	Convolution	64 3x3x64 convolutions with stride [1 1] and padding 'same'
5	'Encoder-Stage-1-ReLU-2'	ReLU	ReLU
6	'Encoder-Stage-1-MaxPool'	Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0 0 0]
7	'Encoder-Stage-2-Conv-1'	Convolution	128 3x3x64 convolutions with stride [1 1] and padding 'same'
8	'Encoder-Stage-2-ReLU-1'	ReLU	ReLU
9	'Encoder-Stage-2-Conv-2'	Convolution	128 3x3x128 convolutions with stride [1 1] and padding 'same'
10	'Encoder-Stage-2-ReLU-2'	ReLU	ReLU
11	'Encoder-Stage-2-MaxPool'	Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0 0 0]
12	'Encoder-Stage-3-Conv-1'	Convolution	256 3x3x128 convolutions with stride [1 1] and padding 'same'
13	'Encoder-Stage-3-ReLU-1'	ReLU	ReLU
14	'Encoder-Stage-3-Conv-2'	Convolution	256 3x3x256 convolutions with stride [1 1] and padding 'same'
15	'Encoder-Stage-3-ReLU-2'	ReLU	ReLU
16	'Encoder-Stage-3-MaxPool'	Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0 0 0]
17	'Encoder-Stage-4-Conv-1'	Convolution	512 3x3x256 convolutions with stride [1 1] and padding 'same'
18	'Encoder-Stage-4-ReLU-1'	ReLU	ReLU
19	'Encoder-Stage-4-Conv-2'	Convolution	512 3x3x512 convolutions with stride [1 1] and padding 'same'
20	'Encoder-Stage-4-ReLU-2'	ReLU	ReLU
21	'Encoder-Stage-4-DropOut'	Dropout	50% dropout
22	'Encoder-Stage-4-MaxPool'	Max Pooling	2x2 max pooling with stride [2 2] and padding [0 0 0 0]
23	'Bridge-Conv-1'	Convolution	1024 3x3x512 convolutions with stride [1 1] and padding 'same'
24	'Bridge-ReLU-1'	ReLU	ReLU
25	'Bridge-Conv-2'	Convolution	1024 3x3x1024 convolutions with stride [1 1] and padding 'same'
26	'Bridge-ReLU-2'	ReLU	ReLU
27	'Bridge-DropOut'	Dropout	50% dropout
28	'Decoder-Stage-1-UpConv'	Transposed Convolution	512 2x2x1024 transposed convolutions with stride [2 2] and cropping [0 0 0 0]
29	'Decoder-Stage-1-UpReLU'	ReLU	ReLU
30	'Decoder-Stage-1-DepthConcatenation'	Depth concatenation	Depth concatenation of 2 inputs
31	'Decoder-Stage-1-Conv-1'	Convolution	512 3x3x1024 convolutions with stride [1 1] and padding 'same'
32	'Decoder-Stage-1-ReLU-1'	ReLU	ReLU
33	'Decoder-Stage-1-Conv-2'	Convolution	512 3x3x512 convolutions with stride [1 1] and padding 'same'
34	'Decoder-Stage-1-ReLU-2'	ReLU	ReLU
35	'Decoder-Stage-2-UpConv'	Transposed Convolut	256 2x2x512 transposed convolutions with stride [2 2] and cropping [0 0 0 0]
36	'Decoder-Stage-2-UpReLU'	ReLU	ReLU
37	'Decoder-Stage-2-DepthConcatenation'	Depth concatenation	Depth concatenation of 2 inputs
38	'Decoder-Stage-2-Conv-1'	Convolution	256 3x3x512 convolutions with stride [1 1] and padding 'same'
39	'Decoder-Stage-2-ReLU-1'	ReLU	ReLU
40	'Decoder-Stage-2-Conv-2'	Convolution	256 3x3x256 convolutions with stride [1 1] and padding 'same'
41	'Decoder-Stage-2-ReLU-2'	ReLU	ReLU
42	'Decoder-Stage-3-UpConv'	Transposed Convolution	128 2x2x256 transposed convolutions with stride [2 2] and cropping [0 0 0 0]
43	'Decoder-Stage-3-UpReLU'	ReLU	ReLU
44	'Decoder-Stage-3-DepthConcatenation'	Depth concatenation	Depth concatenation of 2 inputs
45	'Decoder-Stage-3-Conv-1'	Convolution	128 3x3x256 convolutions with stride [1 1] and padding 'same'
46	'Decoder-Stage-3-ReLU-1'	ReLU	ReLU
47	'Decoder-Stage-3-Conv-2'	Convolution	128 3x3x128 convolutions with stride [1 1] and padding 'same'
48	'Decoder-Stage-3-ReLU-2'	ReLU	ReLU
49	'Decoder-Stage-4-UpConv'	Transposed Convolution	64 2x2x128 transposed convolutions with stride [2 2] and cropping [0 0 0 0]
50	'Decoder-Stage-4-UpReLU'	ReLU	ReLU
51	'Decoder-Stage-4-DepthConcatenation'	Depth concatenation	Depth concatenation of 2 inputs
52	'Decoder-Stage-4-Conv-1'	Convolution	64 3x3x128 convolutions with stride [1 1] and padding 'same'
53	'Decoder-Stage-4-ReLU-1'	ReLU	ReLU
54	'Decoder-Stage-4-Conv-2'	Convolution	64 3x3x64 convolutions with stride [1 1] and padding 'same'
55	'Decoder-Stage-4-ReLU-2'	ReLU	ReLU
56	'Final-ConvolutionLayer'	Convolution	3 1x1x64 convolutions with stride [1 1] and padding 'same'
57	'Softmax-Layer'	Softmax	softmax
58	'Segmentation-Layer'	Pixel Classification	Layer Cross-entropy loss with 'non_vegetation', 'Other_vegetation', and 1 other classes

Appendix B

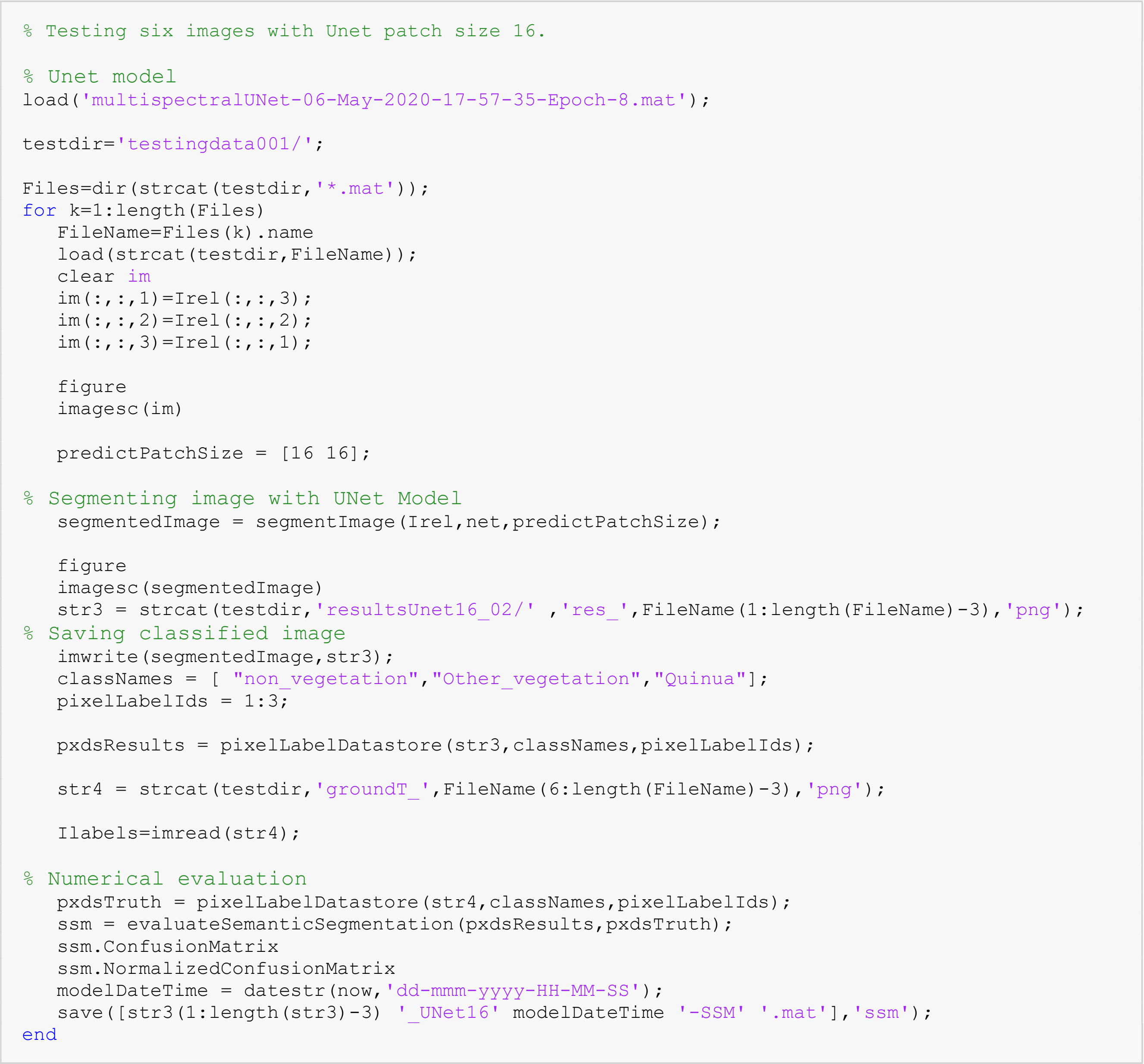
Matlab Code

```
% Training Dataset organization
imds = imageDatastore('train_data.mat','FileExtensions','.mat','ReadFcn',@matReader);
% Classes
classNames = [ "non_vegetation","Other_vegetation","Quinua"];
% Ground Truth data
pixelLabelIds = 1:3;
pxds = pixelLabelDatastore('labeldata',classNames,pixelLabelIds);
% Patch size
imageSize = [16 16 5];
numClasses = 3;
model = 2; %'vgg19'
% Use of SegNet model.
lgraph = segnetLayers(imageSize,numClasses,model);
analyzeNetwork(lgraph)

% Data augmentation
augmenter = imageDataAugmenter('RandRotation',[0 90],'RandXReflection',true);

% Datastore for extracting random patches from pixel label images
dsTrain = randomPatchExtractionDatastore(imds, pxds,[16,16],'PatchesPerImage', 2000000,
'DataAugmentation',augmenter);

% Partitions a subset of observations for training and validation
dsTrainT=partitionByIndex(dsTrain,[1: 1600000]);
dsVal=partitionByIndex(dsTrain,[ 1600001: 2000000]);

inputBatch = preview(dsTrainT);

disp(inputBatch)
disp(lgraph.Layers)

% Preparing training options
initialLearningRate = 0.01;
maxEpochs = 8;
minibatchSize = 128;
l2reg = 0.0001;

options = trainingOptions('sgdm',...
    'InitialLearnRate',initialLearningRate, ...
    'Momentum',0.9,...
    'L2Regularization',l2reg,...
    'MaxEpochs',maxEpochs,...
    'MiniBatchSize',minibatchSize,...
    'LearnRateSchedule','piecewise',...
    'Shuffle','every-epoch',...
    'GradientThresholdMethod','l2norm',...
    'GradientThreshold',0.05, ...
    'Plots','training-progress', ...
    'ValidationData',dsVal, ...
    'ValidationFrequency',2000, ...
    'VerboseFrequency',20);
% Training the SegNet model

modelDateTime = datestr(now,'dd-mmm-yyyy-HH-MM-SS');
[net,info] = trainNetwork(dsTrainT,lgraph,options);

% The trained SegNet model is saved
save(['multispectralSegNet-' modelDateTime '-Epoch-' num2str(maxEpochs)
'.mat'],'net','options');
```

B.1 Matlab code for Segnet Model training considering a 16x16 patch size.

```
% Training Dataset organization
imds = imageDatastore('train_data.mat','FileExtensions','.mat','ReadFcn',@matReader);
% Classes
classNames = [ "non_vegetation","Other_vegetation","Quinua"];
% Ground Truth data
pixelLabelIds = 1:3;
pxds = pixelLabelDatastore('labeldata',classNames,pixelLabelIds);
% Patch size
imageSize = [16 16 5];
numClasses = 3;
encoderDepth = 4;
% Use of Unet model.
lgraph = unetLayers(imageSize,numClasses,'EncoderDepth',encoderDepth)
analyzeNetwork(lgraph)

% Data augmentation
augmenter = imageDataAugmenter('RandRotation',[0 90],'RandXReflection',true);

% Datastore for extracting random patches from pixel label images
dsTrain = randomPatchExtractionDatastore(imds, pxds,[16,16],'PatchesPerImage', 2000000,
'DataAugmentation',augmenter);

% Partitions a subset of observations for training and validation
dsTrainT=partitionByIndex(dsTrain,[1: 1600000]);
dsVal=partitionByIndex(dsTrain,[ 1600001: 2000000]);

inputBatch = preview(dsTrainT);

disp(inputBatch)
disp(lgraph.Layers)

% Preparing training options
initialLearningRate = 0.01;
maxEpochs = 8;
minibatchSize = 128;
l2reg = 0.0001;

options = trainingOptions('sgdm',...
    'InitialLearnRate',initialLearningRate, ...
    'Momentum',0.9,...
    'L2Regularization',l2reg,...
    'MaxEpochs',maxEpochs,...
    'MiniBatchSize',minibatchSize,...
    'LearnRateSchedule','piecewise',...
    'Shuffle','every-epoch',...
    'GradientThresholdMethod','l2norm',...
    'GradientThreshold',0.05, ...
    'Plots','training-progress', ...
    'ValidationData',dsVal, ...
    'ValidationFrequency',2000, ...
    'VerboseFrequency',20);
% Training the UNet model

modelDateTime = datestr(now,'dd-mmm-yyyy-HH-MM-SS');
[net,info] = trainNetwork(dsTrainT,lgraph,options);

% The trained UNet model is saved
save(['multispectralUNet-' modelDateTime '-Epoch-' num2str(maxEpochs)
'.mat'],'net','options');
```

B.2 Matlab code for Unet Model training considering a 16x16 patch size and encoder depth of 3.

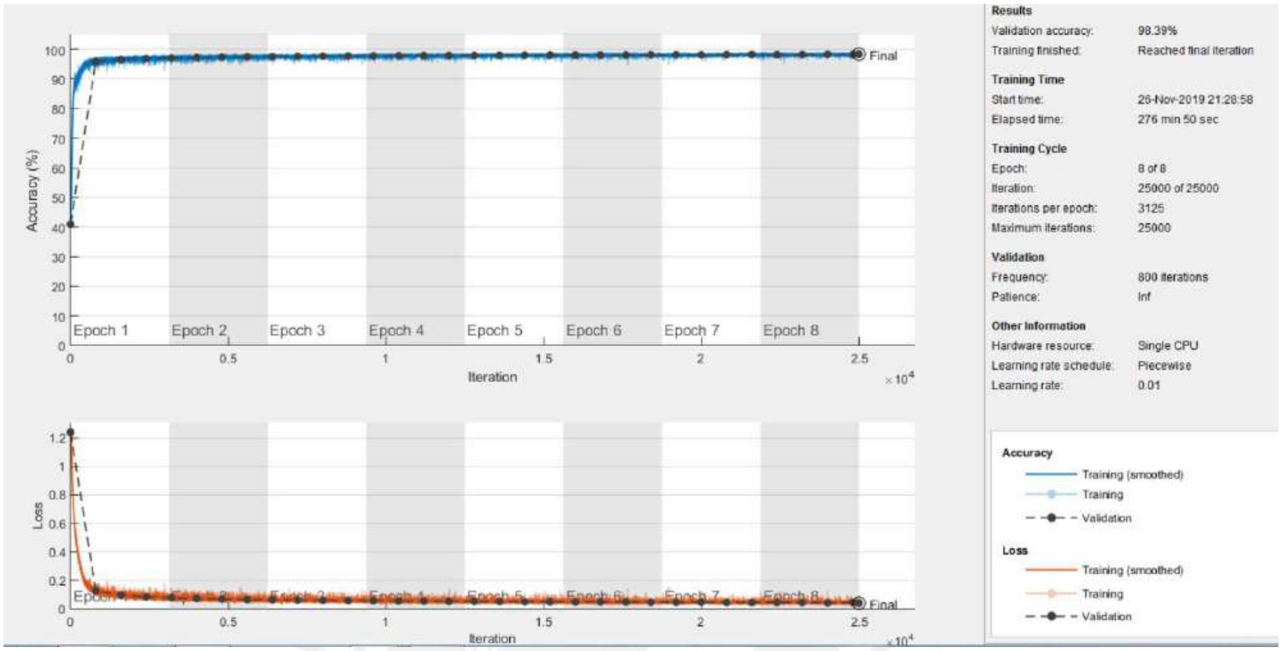


```
% Testing six images with Unet patch size 16.

% Unet model
load('multispectralUNet-06-May-2020-17-57-35-Epoch-8.mat');

testdir='testingdata001/';

Files=dir(strcat(testdir,'*.mat'));
for k=1:length(Files)
   FileName=Files(k).name
   load(strcat(testdir,FileName));
   clear im
   im(:,:,1)=Irel(:,:,3);
   im(:,:,2)=Irel(:,:,2);
   im(:,:,3)=Irel(:,:,1);

   figure
   imagesc(im)

   predictPatchSize = [16 16];

% Segmenting image with UNet Model
   segmentedImage = segmentImage(Irel,net,predictPatchSize);

   figure
   imagesc(segmentedImage)
   str3 = strcat(testdir,'resultsUnet16_02/' ,'res_',FileName(1:length(FileName)-3),'png');
% Saving classified image
   imwrite(segmentedImage,str3);
   classNames = [ "non_vegetation","Other_vegetation","Quinua"];
   pixelLabelIds = 1:3;

   pxdsResults = pixelLabelDatastore(str3,classNames,pixelLabelIds);

   str4 = strcat(testdir,'groundT_',FileName(6:length(FileName)-3),'png');

   Ilabels=imread(str4);

% Numerical evaluation
   pxdsTruth = pixelLabelDatastore(str4,classNames,pixelLabelIds);
   ssm = evaluateSemanticSegmentation(pxdsResults,pxdsTruth);
   ssm.ConfusionMatrix
   ssm.NormalizedConfusionMatrix
   modelDateTime = datestr(now,'dd-mmm-yyyy-HH-MM-SS');
   save([str3(1:length(str3)-3) '_UNet16' modelDateTime '-SSM' '.mat'],'ssm');
end
```

B.3 Matlab code for Unet Model Classification of six test images for evaluation.

APPENDIX C

TRAINING PROGRESS OF DEEP LEARNING MODELS



Results
Validation accuracy: 98.39%
Training finished: Reached final iteration

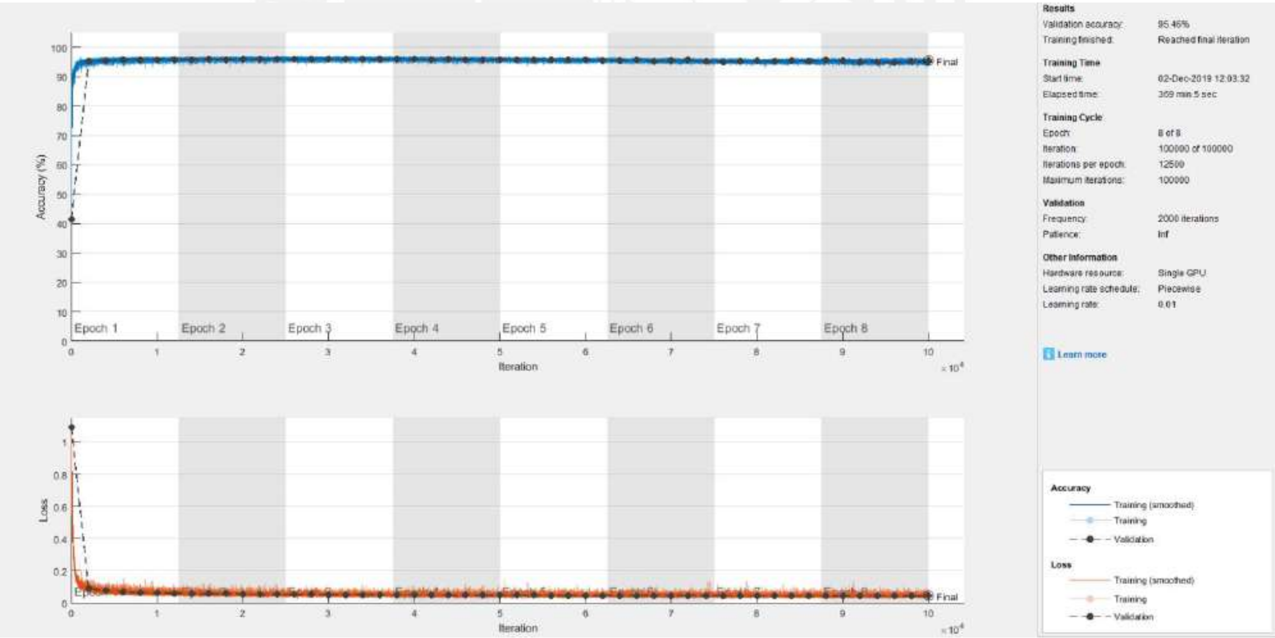
Training Time
Start time: 26-Nov-2019 21:28:58
Elapsed time: 276 min 50 sec

Training Cycle
Epoch: 8 of 8
Iteration: 25000 of 25000
Iterations per epoch: 3125
Maximum iterations: 25000

Validation
Frequency: 800 iterations
Patience: Inf

Other Information
Hardware resource: Single CPU
Learning rate schedule: Piecewise
Learning rate: 0.01

Figure C.1 Training progress for SegNet8



Results
Validation accuracy: 95.46%
Training finished: Reached final iteration

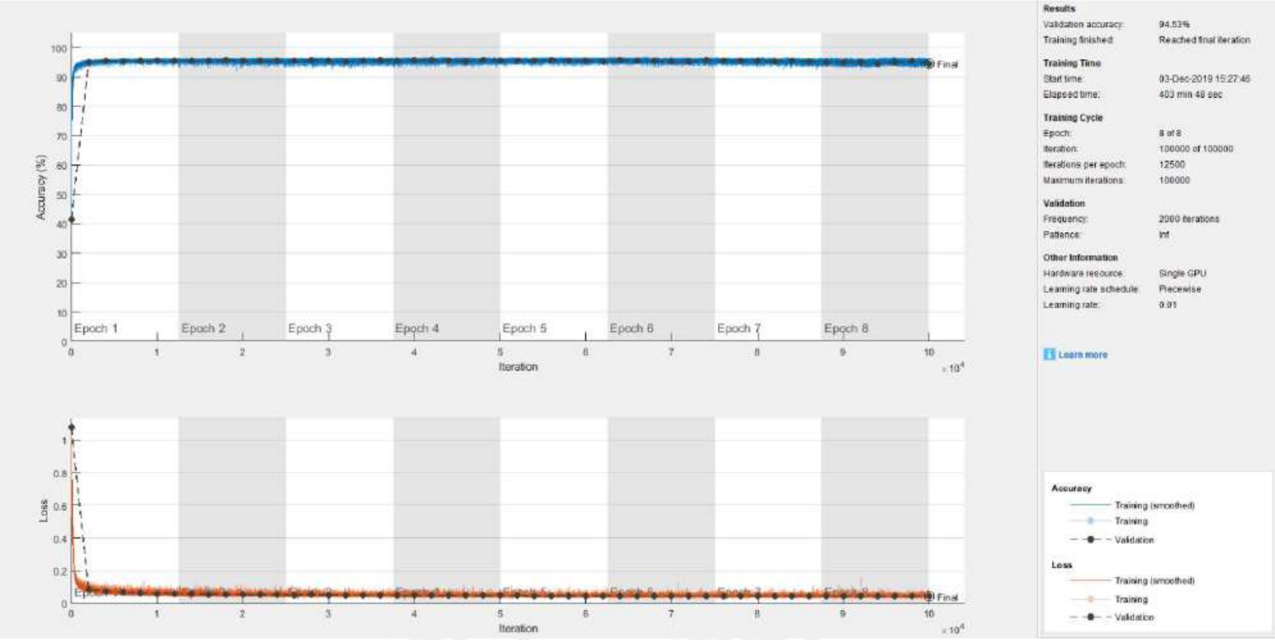
Training Time
Start time: 02-Dec-2019 12:03:32
Elapsed time: 369 min 5 sec

Training Cycle
Epoch: 8 of 8
Iteration: 100000 of 100000
Iterations per epoch: 12500
Maximum iterations: 100000

Validation
Frequency: 2000 iterations
Patience: Inf

Other Information
Hardware resource: Single GPU
Learning rate schedule: Piecewise
Learning rate: 0.01

Figure C.2 Training progress for SegNet12



Results
Validation accuracy: 94.53%
Training finished: Reached final iteration

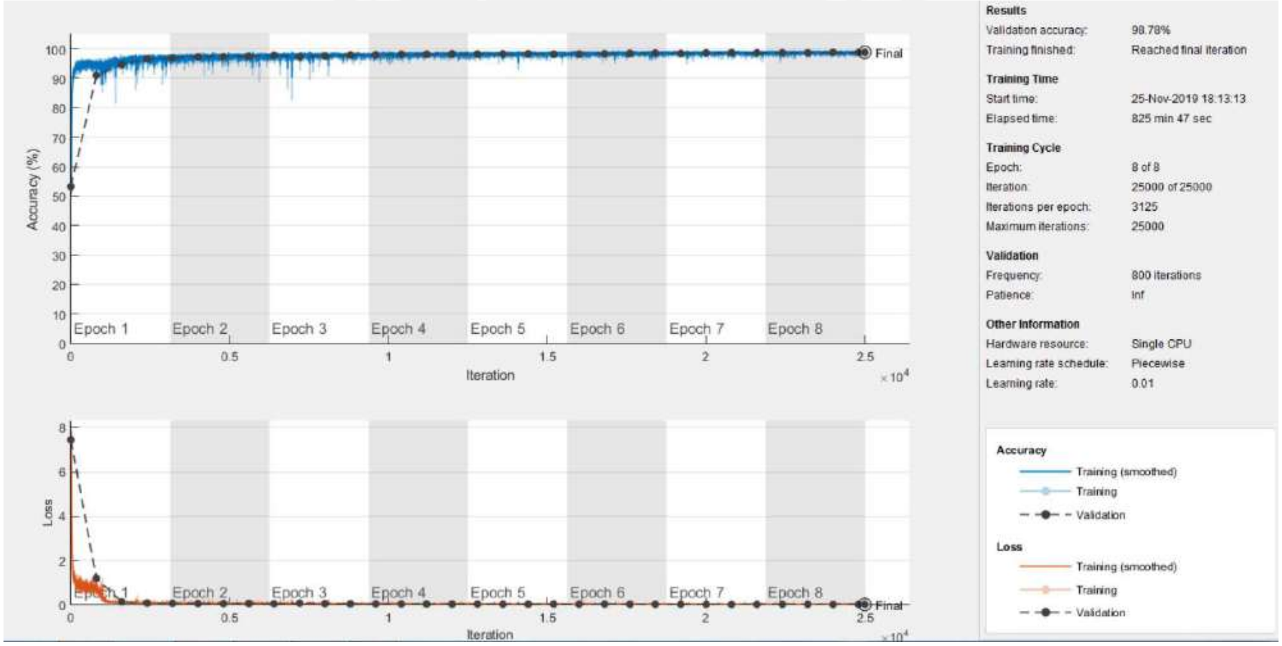
Training Time
Start time: 03-Dec-2019 15:27:46
Elapsed time: 403 min 48 sec

Training Cycle
Epoch: 8 of 8
Iteration: 100000 of 100000
Iterations per epoch: 12500
Maximum iterations: 100000

Validation
Frequency: 2000 iterations
Patience: Inf

Other Information
Hardware resource: Single GPU
Learning rate schedule: Piecewise
Learning rate: 0.01

Figure C.3 Training progress for SegNet116



Results
Validation accuracy: 98.78%
Training finished: Reached final iteration

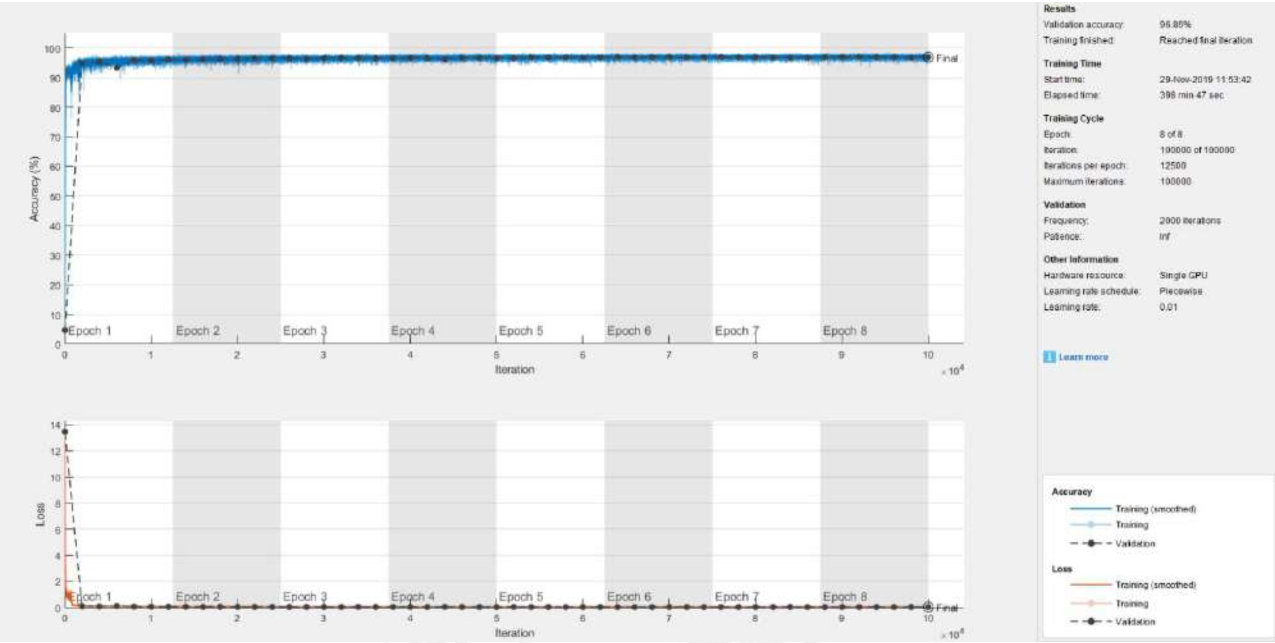
Training Time
Start time: 25-Nov-2019 18:13:13
Elapsed time: 825 min 47 sec

Training Cycle
Epoch: 8 of 8
Iteration: 25000 of 25000
Iterations per epoch: 3125
Maximum iterations: 25000

Validation
Frequency: 800 iterations
Patience: Inf

Other Information
Hardware resource: Single CPU
Learning rate schedule: Piecewise
Learning rate: 0.01

Figure C.4 Training progress for U-Net8



Results

Validation accuracy: 96.85%

Training finished: Reached final iteration

Training Time

Start time: 29-Nov-2019 11:53:42

Elapsed time: 398 min 47 sec

Training Cycle

Epoch: 8 of 8

Iteration: 100000 of 100000

Iterations per epoch: 12500

Maximum iterations: 100000

Validation

Frequency: 2000 iterations

Patience: Inf

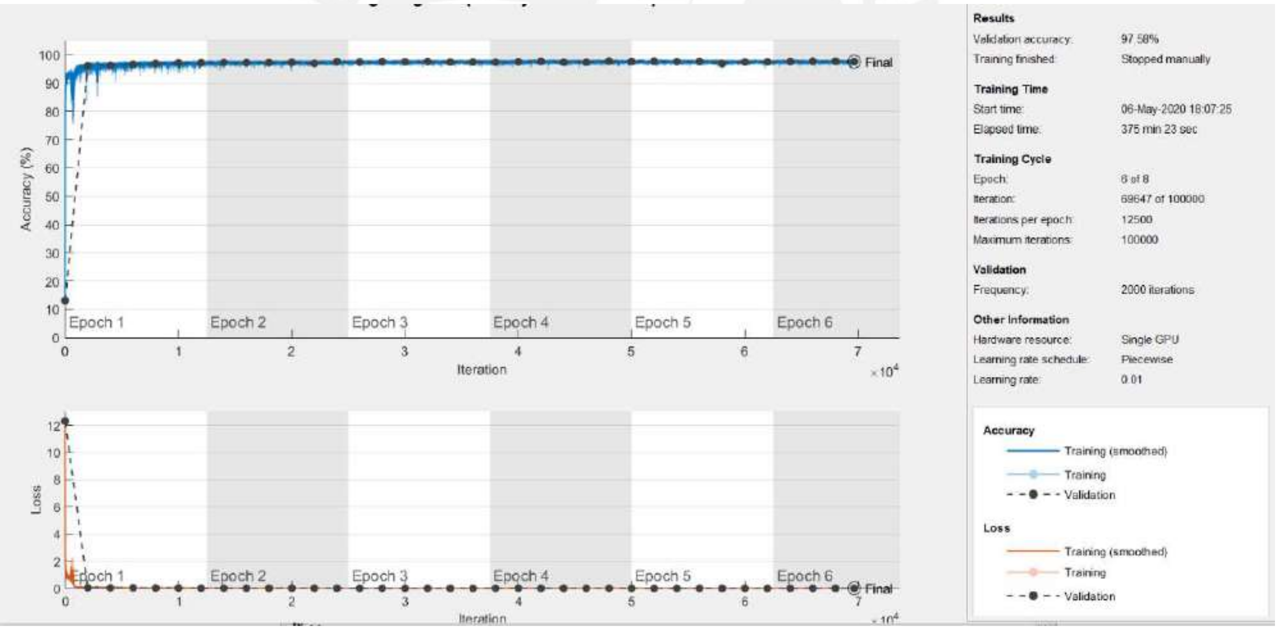
Other Information

Hardware resource: Single GPU

Learning rate schedule: Piecewise

Learning rate: 0.01

Figure C.5 Training progress for U-Net16_3



Results

Validation accuracy: 97.58%

Training finished: Stopped manually

Training Time

Start time: 06-May-2020 18:07:25

Elapsed time: 375 min 23 sec

Training Cycle

Epoch: 6 of 8

Iteration: 69647 of 100000

Iterations per epoch: 12500

Maximum iterations: 100000

Validation

Frequency: 2000 iterations

Other Information

Hardware resource: Single GPU

Learning rate schedule: Piecewise

Learning rate: 0.01

Figure C.6 Training progress for U-Net16_4

APPENDIX D

QUALITATIVE RESULTS

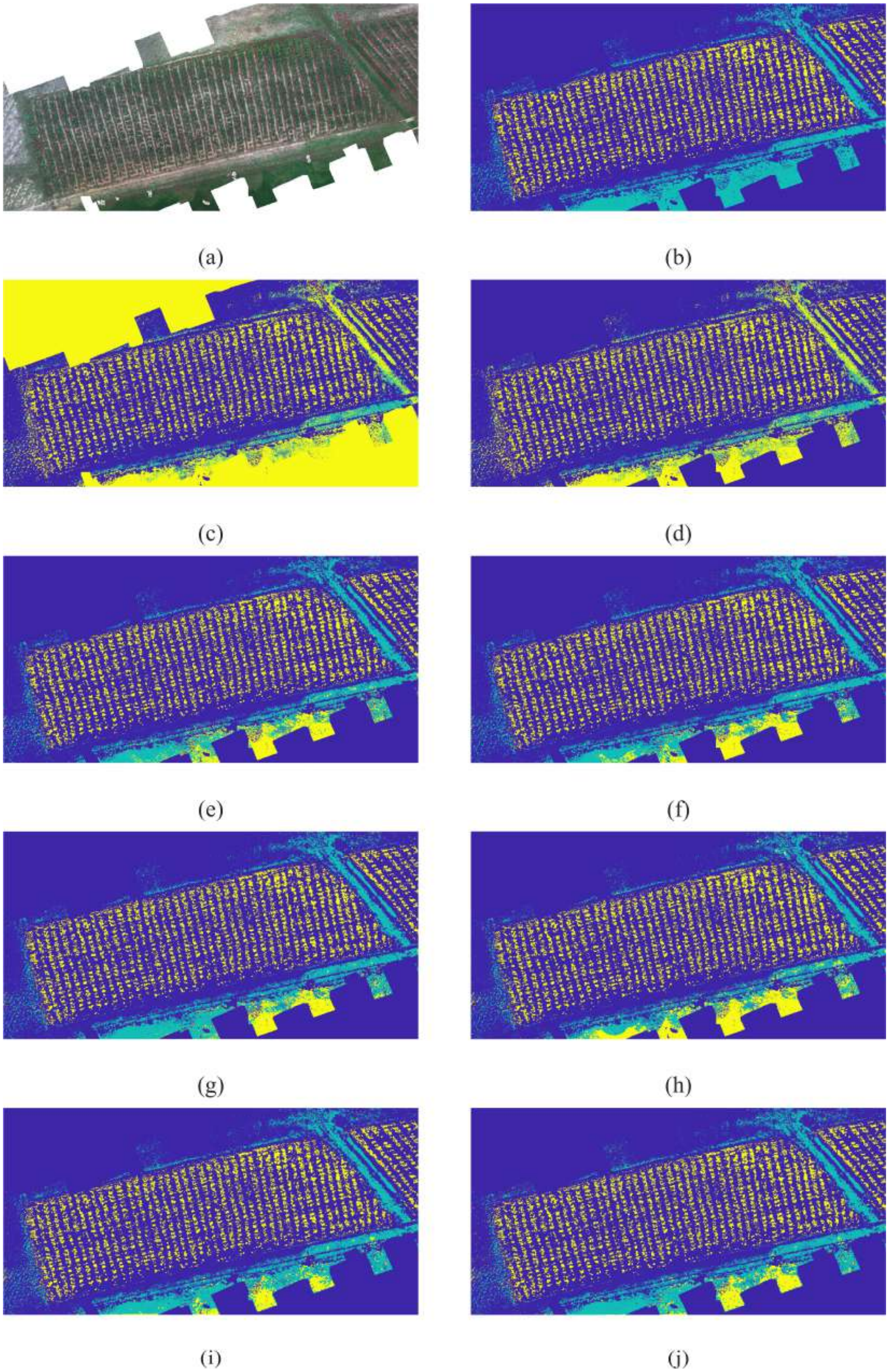


Figure D.1 Qualitative results for testing dataset HYC51 a) RGB image, b) Ground truth, c) SVM, d) Random Forrest, e) SegNet8, f) SegNet12, g) SegNet16, h) U_Net8, i) U_Net16_3 and j) U_Net16_4.

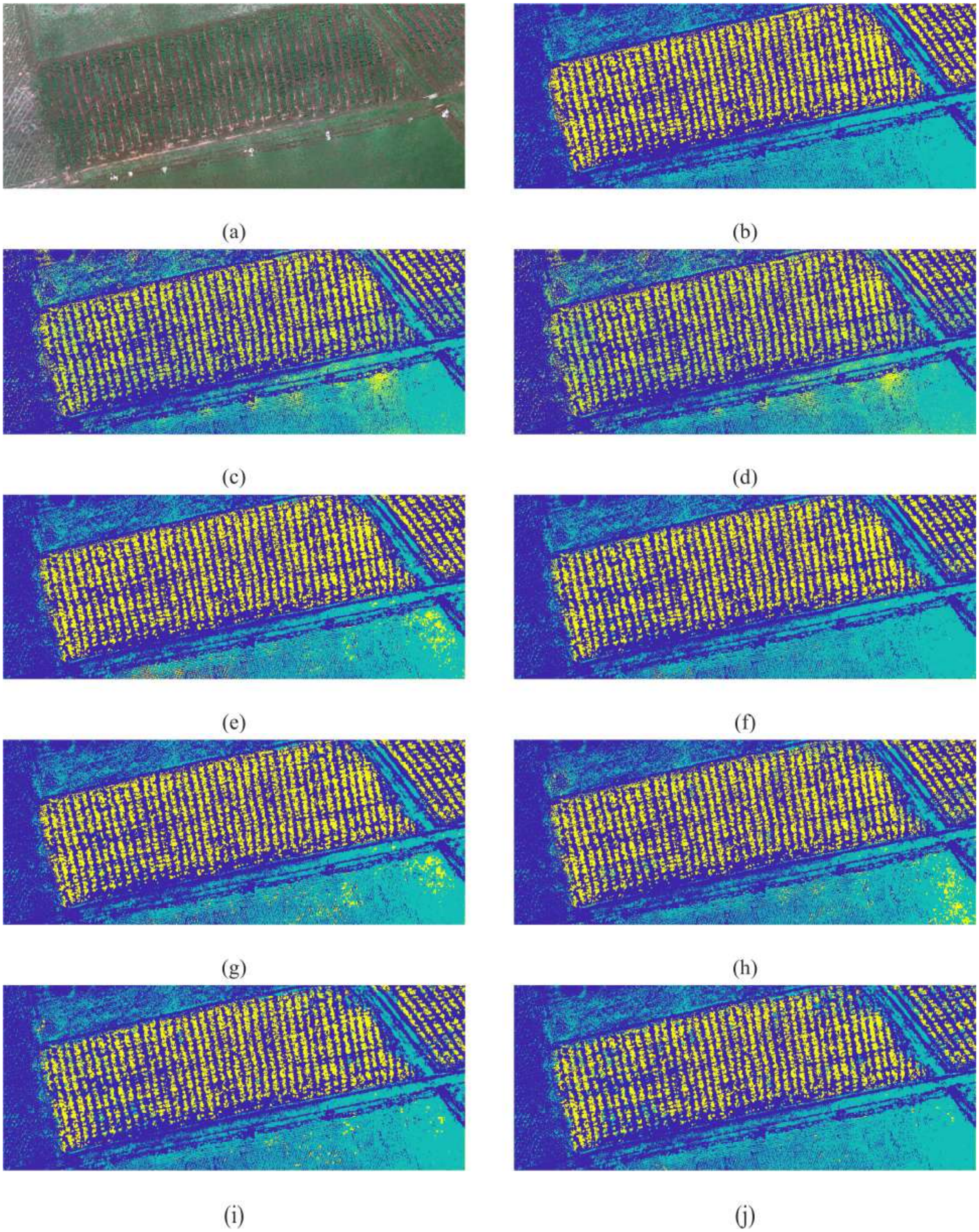


Figure D.2 Qualitative results for testing dataset HYC52 a) RGB image, b) Ground truth, c) SVM, d) Random Forrest, e) SegNet8, f) SegNet12, g) SegNet16, h) U_Net8, i) U_Net16_3 and j) U_Net16_4.

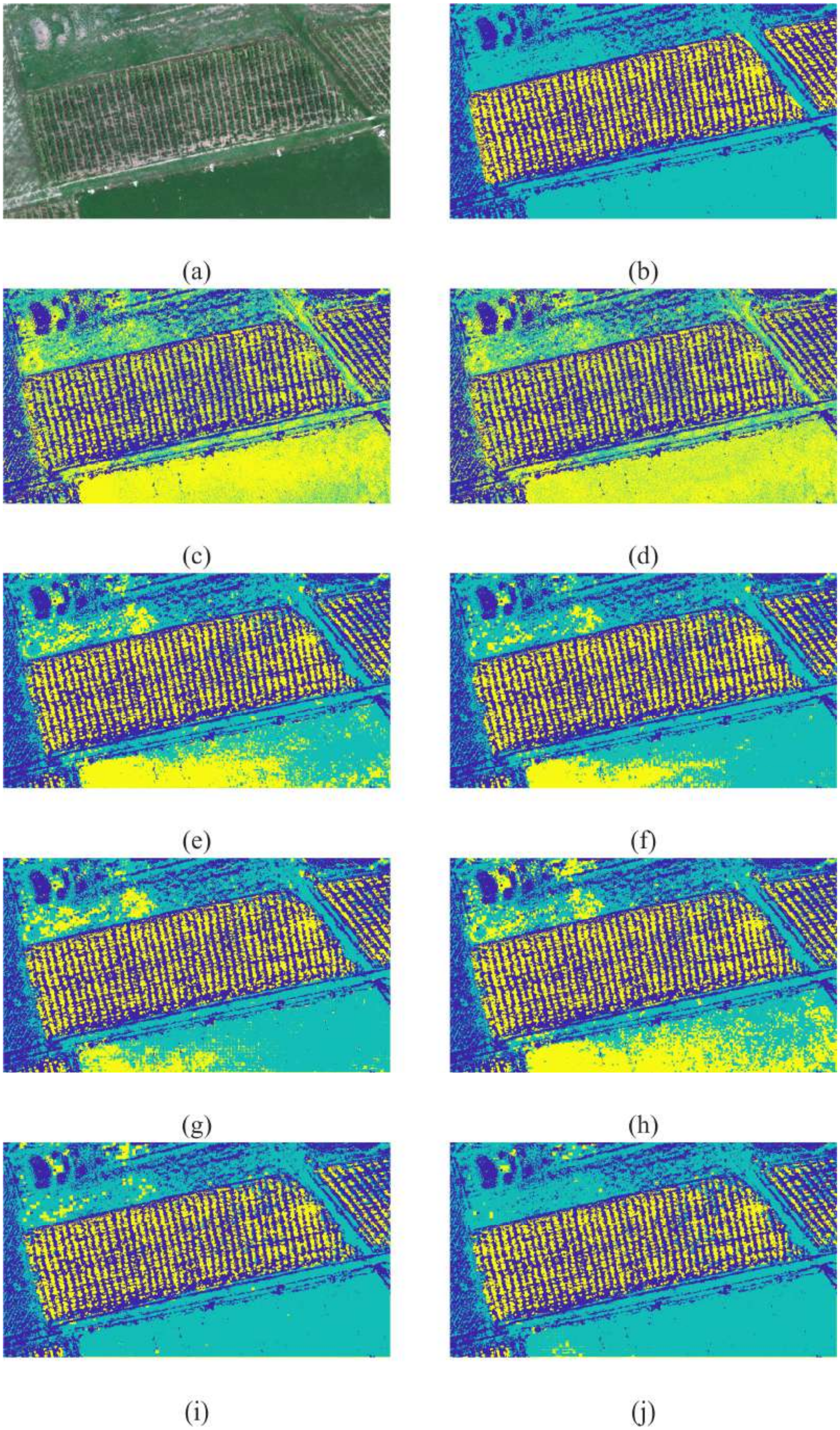

(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)
(j)

Figure D.3 Qualitative results for testing dataset HYC53 a) RGB image, b) Ground truth, c) SVM, d) Random Forrest, e) SegNet8, f) SegNet12, g) SegNet16, h) U_Net8, i) U_Net16_3 and j) U_Net16_4.

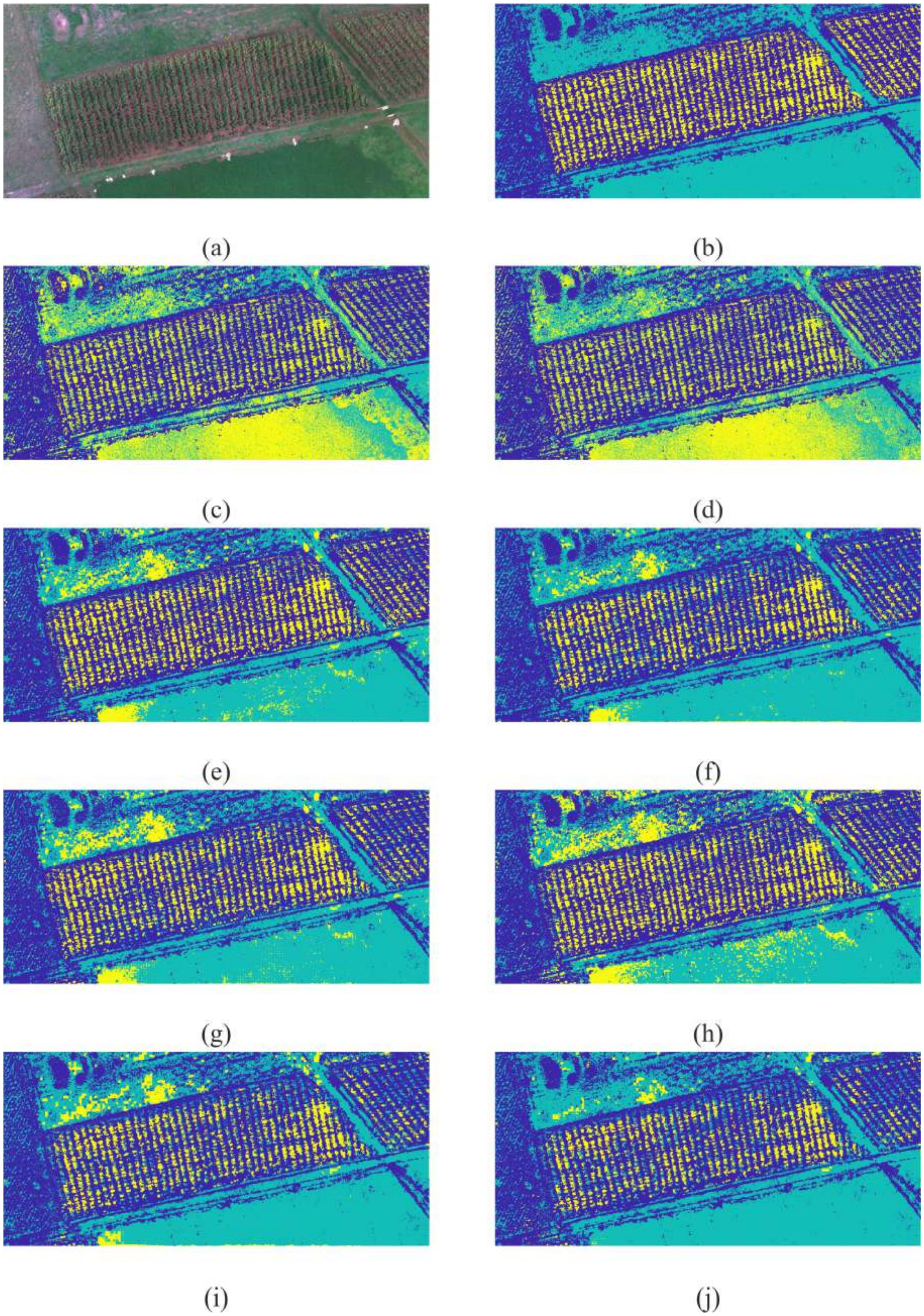


Figure D.4 Qualitative results for testing dataset HYC54 a) RGB image, b) Ground truth, c) SVM, d) Random Forrest, e) SegNet8, f) SegNet12, g) SegNet16, h) U_Net8, i) U_Net16_3 and j) U_Net16_4.

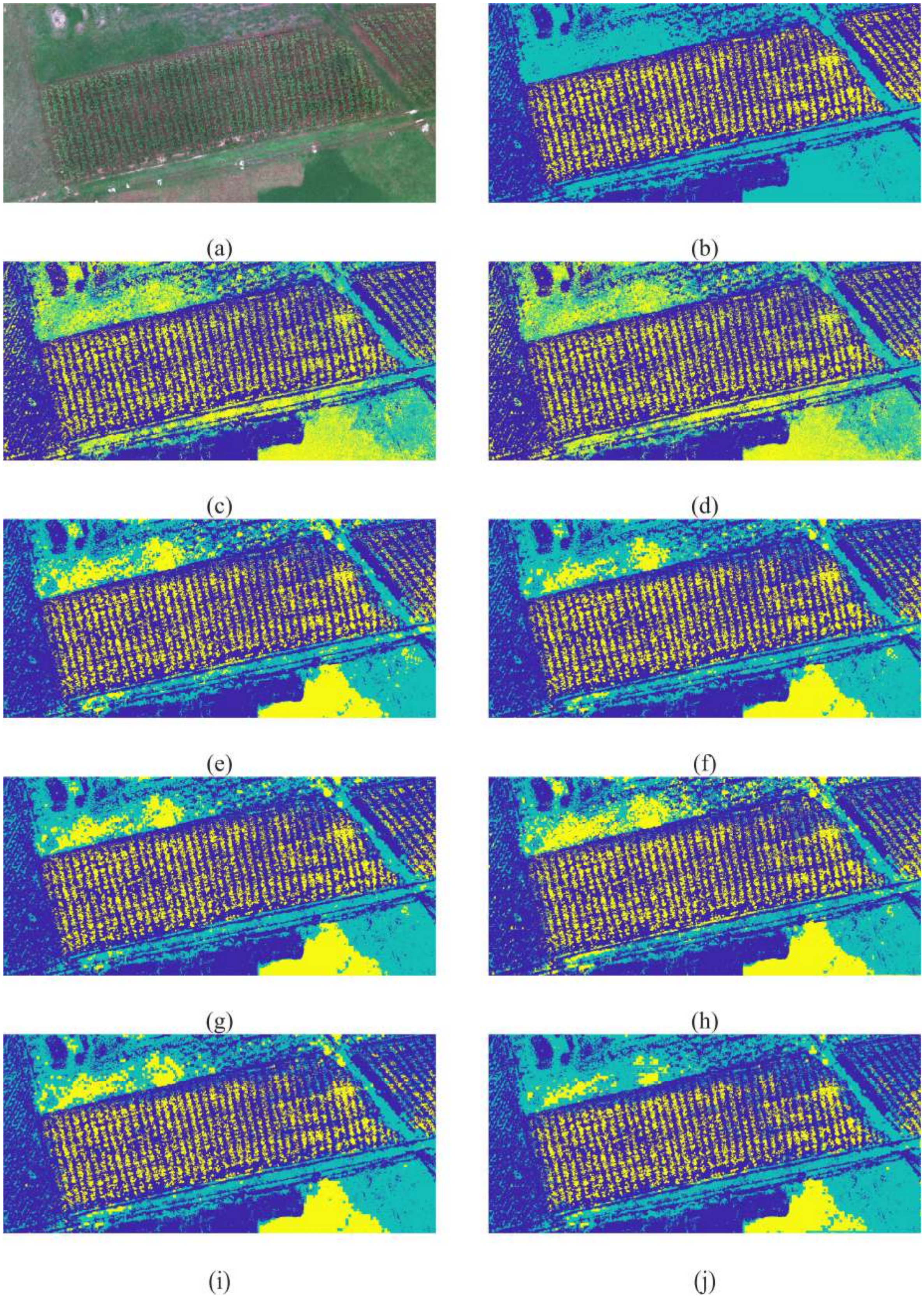


Figure D.5 Qualitative results for testing dataset HYC55 a) RGB image, b) Ground truth, c) SVM, d) Random Forrest, e) SegNet8, f) SegNet12, g) SegNet16, h) U_Net8, i) U_Net16_3 and j) U_Net16_4.

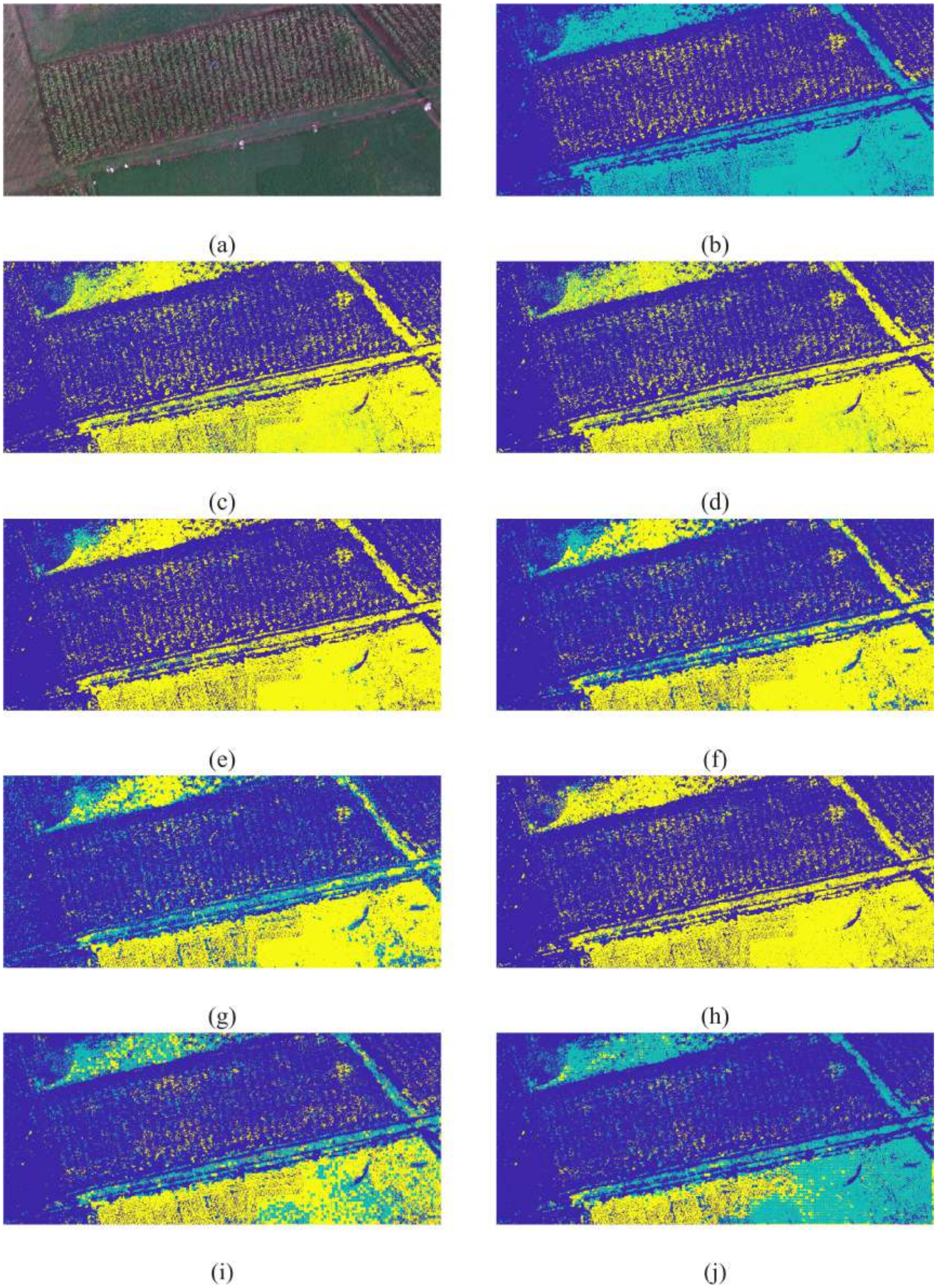


Figure D.6 Qualitative results for testing dataset HYC56 a) RGB image, b) Ground truth, c) SVM, d) Random Forrest, e) SegNet8, f) SegNet12, g) SegNet16, h) U_Net8, i) U_Net16_3 and j) U_Net16_4.