

UNIVERSIDADE FEDERAL DE MINAS GERAIS  
Instituto de Ciências Exatas (ICEX)  
Programa de Pós-graduação em Ciência da Computação

Rafael Marlon Pereira Costa Baeta Carreira

**GEOGRAPHICAL MAPPING OF COFFEE CROPS BY USING CONVOLUTIONAL  
NEURAL NETWORKS**

Belo Horizonte

2017

Rafael Marlon Pereira Costa Baeta Carreira

**GEOGRAPHICAL MAPPING OF COFFEE CROPS BY USING CONVOLUTIONAL  
NEURAL NETWORKS**

**Versão Final**

Dissertação apresentada ao Programa de Pós-Graduação em  
Ciência da Computação da Universidade Federal de Minas Gerais  
como requisito parcial para obtenção do título de Mestre em  
Ciência da Computação

Orientador: Prof. Dr. Jefersson Alex dos Santos

Coorientador: Prof Dr. David Menotti Gomes

Belo Horizonte

2017

© 2017, Rafael Marlon Pereira Costa Baeta Carreira.  
Todos os direitos reservados

Carreira, Rafael Marlon Pereira Costa Baeta.

C314g Geographical mapping of coffee crops by using convolutional networks [manuscrito] / Rafael Marlon Pereira Costa Baeta Carreira — Belo Horizonte, 2017. 67f. il.; 29 cm.

Orientador: Jefersson Alex dos Santos

Coorientador: David Menotti Gomes

Dissertação (mestrado) - Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação.

Referências: p.62-67

1. Computação – Teses. 2. Sensoriamento remoto - Teses. 3. Redes neurais convolucionais – Teses. 4. Café – Cultivo – Teses. I. Santos Jefersson Alex dos. II. Gomes, David Menotti. III. Universidade Federal de Minas Gerais, Instituto de Ciências Exatas, Departamento de Ciência da Computação. IV. Título.

CDU 519.6\*82.10(043)

Ficha catalográfica elaborada pela bibliotecária Irénquer Vismeg Lucas Cruz - CRB 6/819- Universidade Federal de Minas Gerais - ICEx




UNIVERSIDADE FEDERAL DE MINAS GERAIS  
INSTITUTO DE CIÊNCIAS EXATAS  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO


## FOLHA DE APROVAÇÃO


Geographical mapping of coffee crops by using convolutional networks


**RAFAEL MARLON PEREIRA COSTA BAETA CARREIRA**

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

  
PROF. JEFFERSSON ALEX DOS SANTOS - Orientador  
Departamento de Ciência da Computação - UFMG

  
PROF. DAVID MENOTTI GOMES - Coorientador  
Departamento de Informática - UFPR

  
PROF. ARNALDO DE ALBUQUERQUE ARAÚJO  
Departamento de Ciência da Computação - UFMG

  
PROF. RUBENS AUGUSTO CAMARGO LAMPARELLI  
Núcleo Interdisciplinar de Planejamento Energético - UNICAMP

Belo Horizonte, 04 de setembro de 2017.

## **Acknowledgments**

Agradeço primeiramente a Deus por me dar força para concluir mais essa jornada. Aos meus pais por acreditarem em mim. Aos meus avós Ronei e Maria Helena pelo apoio e por serem a minha base, sem vocês esta conquista jamais seria possível, vocês são tudo para mim. Aos meus amigos por estarem ao meu lado em todos os momentos, em especial, Bruno, Edemir, Gabi, Hugo, Keiller, Marcos e Olívio. Aos meus orientadores Jefersson dos Santos e David Menotti por me guiarem com sabedoria.

## Resumo

Nas últimas décadas temos observado um constante crescimento na utilização de imagens de sensoriamento remoto para o monitoramento de atividades e fenômenos na Terra, o que permite o desenvolvimento de diversas aplicações. Dentre as aplicações existentes, a criação de mapas temáticos é uma das mais comuns, pois permite a classificação e análise dos vários objetos que compõe uma imagem podendo ser utilizado para muitos fins, tais como: monitoramento, planejamento e reconhecimento. Mapas temáticos podem ser construídos de forma manual ou por modelos treinados através de aprendizagem supervisionada. Neste tipo de aprendizagem, o sistema é treinado para aprender diferentes padrões através da utilização de amostras rotuladas fornecidas pelo usuário. Nesse sentido, nesta dissertação, um método de geração de mapas temáticos foi desenvolvido para o reconhecimento de colheitas de café visando auxiliar na obtenção de dados dessa cultura agrícola. Pois, apesar de sua grande importância na economia do país e de Minas Gerais, a obtenção de dados ainda é realizada de forma manual. O método desenvolvido neste trabalho baseia-se na combinação de redes neurais de convolução em múltiplas escalas sendo a escolha das redes neurais para o desenvolvimento deste projeto atribuída ao seu desempenho superior aos métodos tradicionais propostos em visão computacional e também por ainda não ser amplamente utilizada em tarefas relacionadas à área agrícola. A utilização de uma abordagem multi-escala está relacionada à variação do tamanho dos padrões encontrados em imagens de satélite e visa tornar o método mais robusto ao permitir que características distintas sejam aprendidas em cada uma das escalas e usadas de forma complementar.

Palavras-chave: Sensoriamento remoto, Classificação, Redes neurais de convolução

## **Abstract**

In the last decades we have observed a constant growth in the use of remote sensing images for the monitoring of activities and phenomena on Earth allowing the development of several applications. Among the existing applications, the creation of thematic maps is one of the most common, since it allows the classification and analysis of the various objects that composes an image and can be used for many purposes, such as: monitoring, planning and recognition. Thematic maps are, usually, generated manually or by the use of models trained by supervised learning. In this type of learning, the system is trained to learn different patterns by using labeled samples provided by the user. In this sense, in this dissertation, a thematic map generation method was developed for the recognition of coffee crops in order to obtain data from this crop. For, despite its great importance in the country's economy and Minas Gerais, data collection is still performed manually. The method developed in this work is based on the combination of convolutional neural networks in multiple scales and the choice by neural networks for the development of this project is attributed to the fact that its performance is superior to the traditional methods proposed in computer vision and also not yet be widely used in tasks related to the agricultural area. The use of a multi-scale approach is related to the variation of the size of the patterns found in satellite images and aims to make the method more robust by allowing distinct features to be learned at each of the scales and used in a complementary way.

**Keywords:** Remote sensing, Coffee classification, Convolutional neural networks.

## Summary

<b>1 Introduction</b> .....	<b>8</b>
<b>1.1 Objective and Contributions</b> .....	<b>10</b>
<b>1.2 Organization of the text</b> .....	<b>11</b>
<b>2 . Background</b> .....	<b>12</b>
<b>2.1 Remote sensing</b> .....	<b>12</b>
<b>2.2 Convolutional Neural Networks</b> .....	<b>14</b>
2.2.1 Neural networks.....	14
2.2.2 Loss function .....	16
2.2.3 Learning algorithm .....	17
2.2.4 Convolutional Neural Networks Architecture.....	19
2.2.4.1 Convolutional layer .....	20
2.2.4.2 Pooling layer .....	22
<b>3. Related Work</b> .....	<b>23</b>
<b>3.2 Spatial feature extraction</b> .....	<b>24</b>
<b>3.3 Coffee crop mapping</b> .....	<b>25</b>
<b>4. Methodology</b> .....	<b>29</b>
<b>4.1 Object representation</b> .....	<b>30</b>
<b>4.2 Learning</b> .....	<b>33</b>
<b>4.3 Fusion</b> .....	<b>38</b>
<b>4.4 Prediction</b> .....	<b>38</b>
<b>5. Experimental evaluation</b> .....	<b>39</b>
5.1.1 Dataset .....	40
5.1.2 Evaluated architectures.....	46
5.1.3 Assessment of results .....	46
<b>5.2 Results and discussion</b> .....	<b>49</b>
5.2.1 Multiple × Individual Scales .....	49
5.2.2 Comparison to the baselines .....	58
<b>6. Conclusions and future work</b> .....	<b>60</b>
<b>Bibliography</b> .....	<b>62</b>

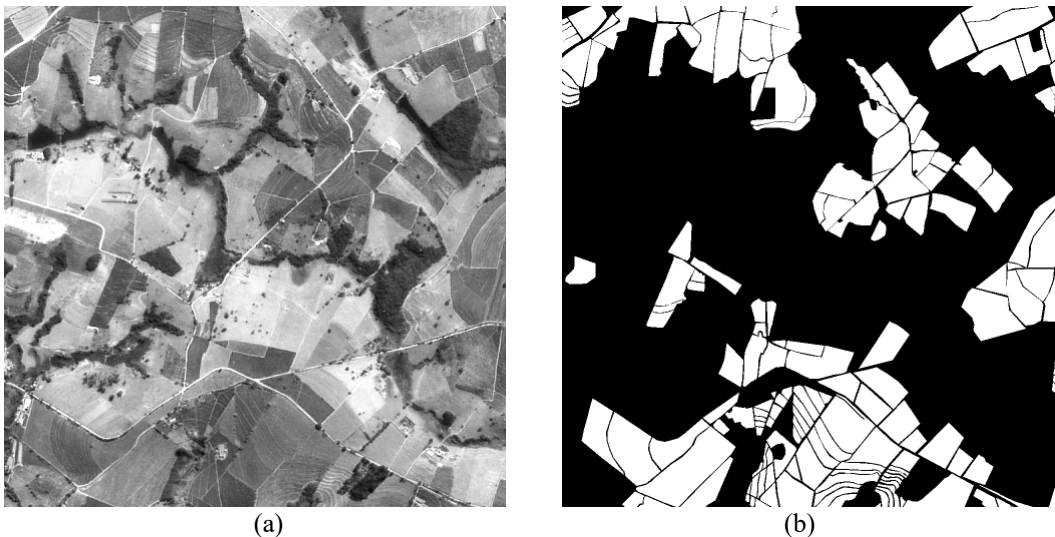


## 1 Introduction

In the last decades, we have observed an increasing in the use of remote sensing images to monitor activities and phenomena on Earth allowing the development of several applications, such as: urban planning, environmental monitoring, [Almeida et al., 2014], [Berger et al., 2013] environmental disasters [Dong and Shan, 2013]. This growth is due to the constant evolution in the quality of the image sensors and an easier access to this type of image.

A remote sensing image (RSI) is formed by the responses of interactions between electromagnetic energy and the various materials that compose the Earth's surface. These responses are obtained by the use of sensors on board satellites or some type of aerial vehicle [Meneses and Almeida, 2012].

Among the diverse applications that can be developed with the use of RSI, the creation of thematic maps is one of the most common, since it allows the classification and analysis of the various objects that compose an image and can be used for many purposes, such as: monitoring, planning and recognition. Thematic maps are images constructed in order to identify the category in which each object belongs in the image and these images, usually, are generated by the use of models trained by supervised learning that is a process in which the system is trained to learn patterns by using samples labeled by the user. In Figure 1.1b, we can see the thematic map generated from the spatial information presented in Figure 1.1a.



**Figure 1.1:** Example of a thematic map using a RSI: (a) a scene taken over Monte Santo de Minas County (State of Minas Gerais, Brazil) and (b) a thematic map that indicates coffee regions (white) and non-coffee (black)

The cultivation of coffee is a very important economic activity in Brazil. According to the Ministry of Agriculture [Ministério da agricultura, 2017], Brazil is the largest exporter of coffee and the second largest consumer of the product being this one of the main export sectors of the country. In December 2016, the product accounted for 9.8 % of Brazilian exports. The coffee park is estimated to be 2.22 million hectares and is distributed in 15 states and 98.6% of the national production are concentrated at Minas Gerais, Espírito Santo, São Paulo, Bahia, Rondônia, Paraná, Rio de Janeiro, Goiás and Mato Grosso.

Thus, in order to maintain the quality and quantity of the product, it is necessary to develop efficient and effective techniques for the management of its crops.

Currently, data on coffee in Brazil is mainly obtained by mean of manual survey applied to producers, cooperatives and public agencies [da Silveira et al., 2017]. This procedure makes obtaining the data in short time very hard, moreover, it has a high cost [Ippoliti-Ramilo et al., 1999]. In this way, the extraction of data by satellite images becomes an efficient way to obtain information and allow the development of several approaches for the recognition and classification of coffee areas [Souza et al., 2016], [Santos et al., 2010], [Faria et al., 2012], [Faria et al., 2014], [Chemura et al., 2016], [Ferreira et al., 2016].

The identification of coffee areas is not a trivial task. Some intrinsic issues faced in this activity are the fact that this product is cultivated in different areas with different climates, reliefs, altitudes and latitudes, which allow the production of various types of coffee producing a wide variety of patterns. The characteristics of the relief can lead to the occurrence of shadows that modifies the encoding of the spectral information obtained by the satellites, reducing or even eliminating them [Zhou et al., 2009]. Moreover, the growth of coffee does not occur in a seasonal way which allows the existence of plantations of different ages. In addition to the challenges provided by the identification of coffee, we have those related to remote sensing images. When working with remote sensing images we have to deal with relatively large images and some images may have millions of pixels and still contain thousands of time series. These properties may interfere with the performance of machine learning algorithms, even those given as state of the art (e.g., Neural Networks, Support Vector Machine), because most of the proposed approaches are not scalable. Furthermore, high resolution images are usually composed of a large amount of objects of different patterns and sizes which makes choosing an ideal scale to deal with such differences a very difficult task which also strongly influences the performance of the algorithms.

Other factors of great impact are the mixtures of pixels (pixel that contains information of different objects, due to low resolution), noises and corrupted bands (hyperspectral images).

Finally, we have to ensure the efficiency of the algorithms to handle a large amount of data, so that we can use them in real contexts, because some types of data must be analyzed almost in real time, such as earthquakes and tsunamis.

## **1.1 Objective and Contributions**

In this work, convolutional neural networks are employed to perform coffee crop mapping. This approach is modeled as a supervised learning problem that aims at assigning a label for each pixel in the image, which is a process known as semantic segmentation. In other words, the main objective of the approach proposed in this work is to recognize coffee crops in remote sensing images to generate a thematic map that consists of regions classified as coffee and non-coffee.

The proposed approach is based on the combination of convolution neural networks (CNN) in multiple scales. Neural networks are used for the development of this project for the reason that its performance has been superior to the great majority of the traditional methods proposed in the most diverse areas, such as iris recognition [Silva et al., 2015], [Luz and Menotti, 2015], vehicles [Menotti et al., 2014], and also in previous works for classification of remote sensing images [Castelluccio et al., 2015], [Penatti et al., 2015], [Nogueira et al., 2017].

The use of a multi-scale approach is related to the variation of the size of the patterns found in satellite images and aims to make the method more robust by allowing distinct features to be learned at each of the scales and used in a complementary way and, so that the different networks are combined a fusion is used at the decision level through majority voting.

## **1.2 Organization of the text**

This dissertation was organized as follows: Chapter 2 is a brief explanation of the main background concepts required to follow the remainder chapters in this work. Chapter 3 is an overview of the current semantic segmentation and multi-scale classification on coffee crops. Chapter 4 presents the details of the proposed approaches. Chapter 5 shows the corresponding evaluation protocols and experimental results of the proposed methods. We conclude this work in Chapter 6 with some remarks and future directions of this research.

## 2 . Background

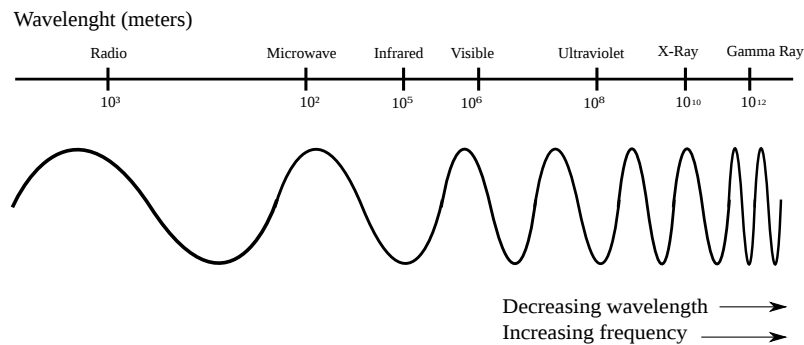
A thematic map is an image constructed in order to identify the category of objects of the image. These images, usually, are generated by the use of models trained by supervised learning that is a task in which the system is trained to learn patterns by using samples labeled by the user. In this work, the process of creating thematic maps is composed of the following steps: data acquisition, features extraction and classification.

This chapter introduces fundamental concepts about remote sensing and data acquisition in Section 2.1 and describes the feature extraction by convolutional neural networks (CNN) architecture in Section 2.2.

### 2.1 Remote sensing

The concept of remote sensing can be defined in a simple and scientific way as: “Remote sensing is the science which aims to obtain images from the earth surface through the detection and quantitative measurement of the answers of the interaction of electromagnetic radiation with terrestrial materials.” [Meneses and Almeida, 2012].

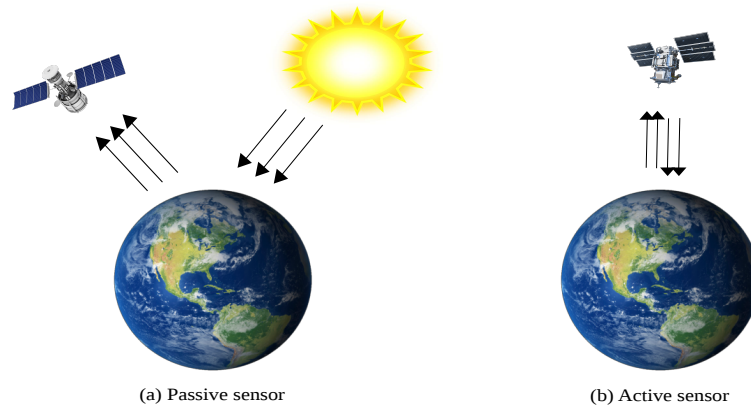
The electromagnetic radiation can be decomposed into several regions depending on the characteristics of the electromagnetic wave, as you can see in Figure 2.1 each region of the electromagnetic wave can provide different types of information which allows the development of diverse applications [Zhou and Wei, 2016], [Wang et al., 2017], [Yokoya et al., 2014].



**Figure 2.1:** Eletromagnetic spectrum

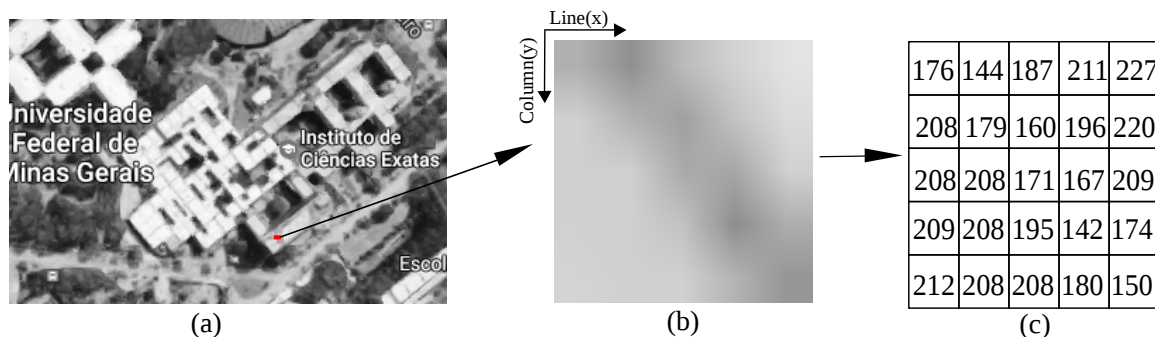
The reflected or emitted electromagnetic radiation of an object is obtained by a device called sensor that can be classified, according to its characteristics, as passive or active.

A sensor is called passive when it uses an external stimulus to obtain data, which can be an energy reflected or emitted by the Earth's surface. Unlike passive sensors, an active sensor produces its own energy and uses it to perform data collection on Earth (Figure 2.2).



**Figure 2.2:** Example of passive and active sensors. (a) A passive sensor that receives the electromagnetic radiation emitted by the sun and reected by the Earth. (b) An active sensor that emits its own signal and receives it back

The output of the data collected from a sensor is usually a digital image obtained from the observed region. The digital image ( $H \times W$ ) can be dened as a discrete representation of a real scene formed by  $H \times W$  pixels, where  $H$ ,  $W$  is the height and width of the image, respectively. Each pixel  $p$  is expressed as a vector representing a measure taken from a region. An example of digital image of remote sensing covering an urban area is shown in Figure 2.3.



**Figure 2.3:** Digital image (a) With zoomed area of the group of pixels in gray values (b) and corresponding digital values

Since each region of the electromagnetic wave can be represented digitally, it is quite common for satellite images to be composed of overlays of several digital images that are referred

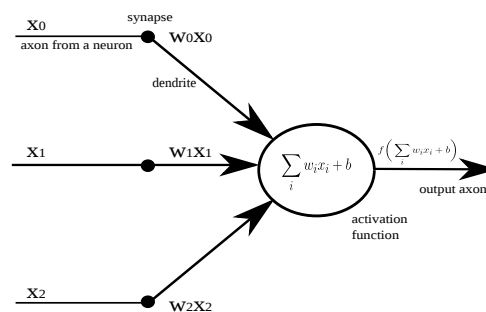
to as band or channel. This overlay of bands originates different types of images, a fairly common image type formed by channel overlays are the RGB images which is composed by the overlap of the red (R), green (G) and blue (B) band regions. The stacking of many bands also originates the multispectral and hyperspectral images where the difference between them consists of the number of channels composing the image, multispectral images is usually composed of 3 to 10 wider bands whereas hyperspectral images are constituted by hundreds of small bands.

In this work, the images used are composed of three channels: red, green and a near infrared channels. The near-infrared channel was used instead of the blue channel because it helps to emphasize the differences between plants.

## 2.2 Convolutional Neural Networks

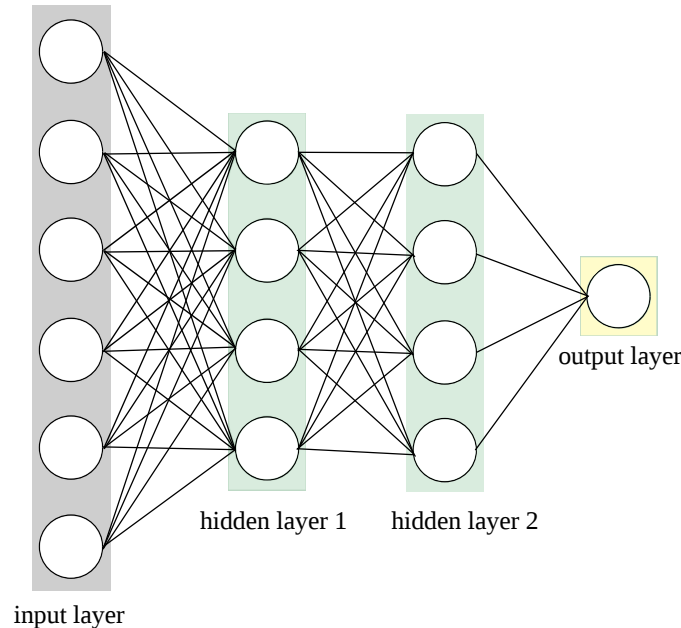
### 2.2.1 Neural networks

The Artificial Neural Network (ANN) is an architecture inspired by biological neural systems, more specifically, the brain. As the brain, an artificial neural network is made up of neurons. The neurons that compose an artificial neural network (Figure 2.4) simulate the functioning of biological neurons, and as these neurons, receive signals (synapses) from other neurons (I.e.,  $x_0$ ) by the dendrites, these signals (I.e.,  $w_0$ ) interact with the receiver neuron (I.e.,  $w_0x_0$ ) and the sum of the interactions ( $\sum_i w_i x_i + b$ ) are used to calculate the activation rate of these neuron by an activation function  $f$ .



**Figure 2.4:** Example of artificial neuron

A neural network is built by the combination of neurons that are grouped together originating layers and, the neurons of one layer can only connect to neurons of other layers. In Figure 2.5, an example of ANN called Multi-Layer perceptron (MLP) is shown.



**Figure 2.5:** Example of artificial neural network

This ANN is composed of 4 layers known as fully connected layer. In this type of layer, neurons of two adjacent layers are completely paired. In the above example, the gray layer represents the input, the two green layers are the hidden layers and the last layer (yellow) is the neural network output which can be a class score (rank), real value numbers or a target of real value (regression)

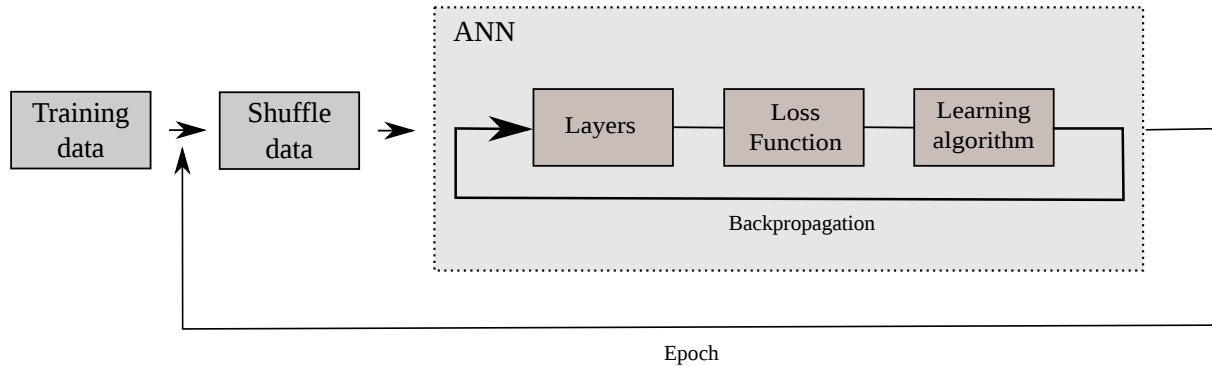
The aim of an artificial neural network is to learn features by adjusting the weights ( $W$ ) values to distinguish data that is not linearly separable. The learning of these features can be achieved by supervised learning, which is a technique that uses previously labeled data called training data to analyze the performance of the algorithm and adjust it to become more effective.

During the learning process the training data is used several times as input by the ANN that is initialized with random weights. The dot product of each sample ( $x_i$ ) of the training data with the ANN weights ( $w_i$ ) will generate predictions ( $y_i$ ) these predictions will be compared to the real labels ( $y_i$ ) to measure network error by a function called loss function.

This function is used as a guide to the learning algorithm that aims to minimize it by adjusting the weights of the whole network by a process called backpropagation. All training data



is sent to the neural network and when all data is used an epoch is completed. At the end of an epoch, if the network did not reach an acceptable loss value, the process can be repeated by randomly reorganizing the training data and resubmitting it to the neural network. The complete process can be seen in Figure 2.6.



**Figure 2.6:** Example of artificial neural network

### 2.2.2 Loss function

A loss function is used to quantify the quality of any particular set of weights  $W$ . A low loss value indicates that a  $W$  configuration must produce predictions for examples  $x_i$  consistent with their groundtruth labels  $y_i$ . There are many loss functions, such as: mean squared error (MSE), cross entropy, exponential, each with its advantages and limitations. Among them, one of the most commonly used loss function is crossentropy, which is dened as:

$$L(W) = -\frac{1}{N} \sum_{i=1}^N [y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)] \quad (2.1)$$

where  $N$  is the number of examples,  $y_i$  is the groundtruth and  $\hat{y}_i$  is the probability assigned by the classifier to a given class. The main advantage of cross-entropy is attributed to its way to penalty wrong outputs, that is, outputs that are very wrong are heavily penalized while outputs close to the expected class have an error close to 0. The loss function works jointly with the learning algorithm and its outputs are used to guide when to stop training. The learning algorithm aims to minimize the loss function ( $L(W)$ ) by updating the weight values ( $W$ ).

### 2.2.3 Learning algorithm

There are several learning algorithms for training Neural Networks (Gradient Descent, Newton Method, Conjugative Gradient, Quasi Newton, Levenberg Marquardt).

The Newton Method, Quasi Newton, Levenberg Marquardt are second order methods, since they use the exact or approximate Hessian matrix and its inverse, which is a matrix with partial second derivatives, that is computationally expensive to calculate, which restricts the use of these algorithms for small datasets. The conjugate gradient also uses the Hessian matrix, however, it avoids its inverse which reduces computational cost.

Among these algorithms, the gradient descent (GD) is the most common used since it is a first order method that uses information from the gradient vector to calculate the updates that make it a fast algorithm and the best choice to handle large amount of data.

The gradient descent is a way to minimize the loss function  $L(\theta)$  parameterized by the parameters of a model  $\theta \in \mathbb{R}^d$ , the parameters are updated in the opposite direction of the gradient of the loss function  $\nabla_{\theta}L(\theta)$  [Ruder, 2016], the opposite direction is used, because, the gradient points in the direction of the highest increase of  $L(\theta)$  and the aim is to minimize this function.

The size of the update step is determined by the learning rate  $\eta$  and an appropriate choice of this parameter is a difficult task. A large learning rate can hinder convergence and cause fluctuations of the loss function around the minimum or even diverge, while a small leads to slow convergence.

To perform the updates, training data is used by the learning algorithm to adjust the weights and the way that the data is sent to the learning algorithm can give rise to three variants of the gradient descent:

- **Batch gradient descent:** It uses all the data to do an update on the network which makes the learning process more precise, however, it is very slow.

$$\theta = \theta - \eta \nabla_{\theta} L(\theta) \quad (2.2)$$

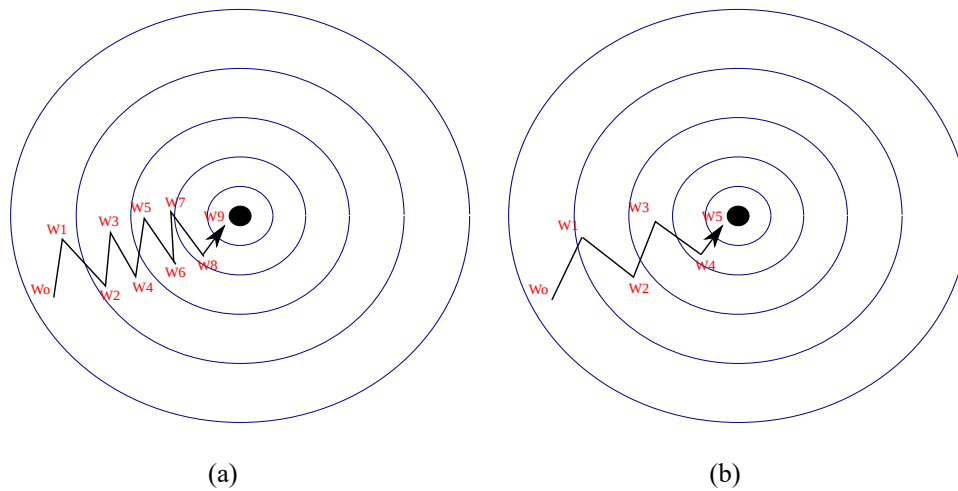
- **Stochastic gradient descent (SGD):** It updates the network for each training sample  $x_i$  and label  $y_i$ , this approach makes learning faster, but unstable.

$$\theta = \theta - \eta \nabla_{\theta} L(\theta; x_i; y_i) \quad (2.3)$$

- **Mini-batch gradient descent:** it is the half term between batch gradient descent and stochastic gradient descent. In this approach the update is performed from small portions of training examples, called, mini-batches

$$\theta = \theta - \eta \nabla_{\theta} L(\theta; x_{i:i+n}; y_{i:i+n}) \quad (2.4)$$

A common problem encountered by the learning algorithm is to minimize highly non-convex error functions. In these functions, there are many suboptimal local minima, and in this type of point, there exist dimensions that slope up, slopes down and a plateau with the same error which hinders the gradient descent algorithm to find a better point. In these regions, the gradient is close to zero in all dimensions that make the gradient descent algorithm to oscillate. To deal with this, a method called momentum can be used to help the gradient descent algorithm to follow a relevant direction (Figure 2.7).



**Figure 2.7:** Example of SGD (a) without momentum (b) with momentum

The idea of momentum is to add a fraction of the previous step to the current update vector (Equation 2.5):

$$\begin{aligned} \mathcal{V}_t &= \mathcal{V}_t + \eta \nabla_{\theta} L(\theta) \\ \theta &= \theta - \mathcal{V}_t \end{aligned} \quad (2.5)$$

This addition is accumulated when the parameters are updated in the right direction and reduced when the updates change direction. The use of the momentum makes the convergence faster while reducing the oscillation.

#### **2.2.4 Convolutional Neural Networks Architecture**

Convolutional Neural Networks (CNNs), as the MLP, are deep learning architectures typically composed of multiple layers that can learn data-driven features and classifiers while adjusting learning, in processing time, based on network accuracy.

Since the layers of an MLP are fully connected, depending on the size of the input and the number of layers, there will be so many parameters to train that it will become impractical and even if it is not intractable, a large number of parameters can lead to the overfitting, that occurs when the architecture memorizes the training data, which makes very hard for the network to recognize an unseen sample.

CNN is an ANN that uses shared weights which drastically reduces the number of parameters, computational cost and the overfitting effect. The shared weights use also allows a greater number of layers which permit to generate higher level features that are more representative. Since the encoding of spatial features in an efficient and robust way is the key to the generation of discriminatory models, the feature learning step, which can be stated as a technique that learns a transformation from raw input to a representation which improves separability of the class, is a great advantage of CNNs when compared to conventional methods.

In fact, multiple layers (responsible for encoding spatial resources automatically) learn adaptive and specific resource representations in a hierarchical, data-dependent manner. Thus, low-level descriptors are learned in the early layers of the network and high-level features in the deepest. This process learns all viable data information, which creates robust features and classifiers.

In Figure 2.8, we can see an example of CNN and the common layers found in this architecture, that is, pooling, convolution and fully connected layers. The addition of fully-connected layers is a way to learn nonlinear combinations of high-level features.

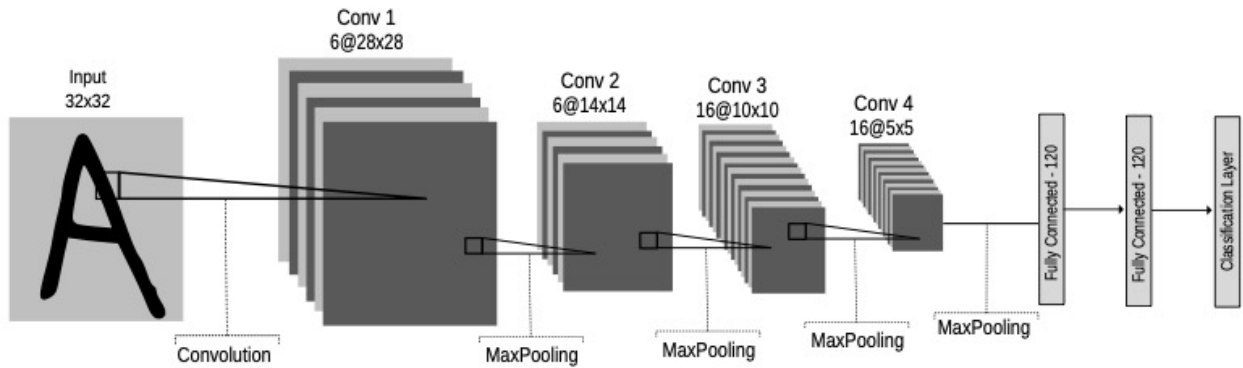


Figure 2.8: CNN proposed by [LeCun et al., 1998]

### 2.2.4.1 Convolutional layer

The convolutional layer is the most important layer of a convolutional neural network. Unlike the fully-connected layer, where all neurons in one layer are connected to all neurons in the previous layers, the convolutional layers connect each neuron to a local region (filter) of the previous layers. Every filter is spatially small and extends through the full depth of the input volume, the convolution process is done by sliding each filter across the width and height to compute dot products between the filter and the input values. The output of this process is called feature map and its size is controlled by three hyperparameters: **depth**, **stride** and **zero-padding**.

- **Depth**: controls the number of filters used in the convolution and each filter will learn a different feature
- **Stride**: determines the size of the filter step. For example, if the stride is 1, the filter will move one pixel at a time
- **Zero-padding**: it wraps the input volume with zeros helping to control the spatial size of the output volume. The spatial size of the output volume can be calculated as follows:

$$\text{Output}_{\text{volume}} = \frac{V - F + 2P}{S + 1} \quad (2.6)$$

where  $V$  is the volume size,  $F$  filter size,  $S$  stride applied and  $P$  the amount of zeropadding used. If the output of Equation 2.6 is not an integer, then, the value of any parameter cannot be used for the input.

An example of convolution is shown in Figure 2.9.

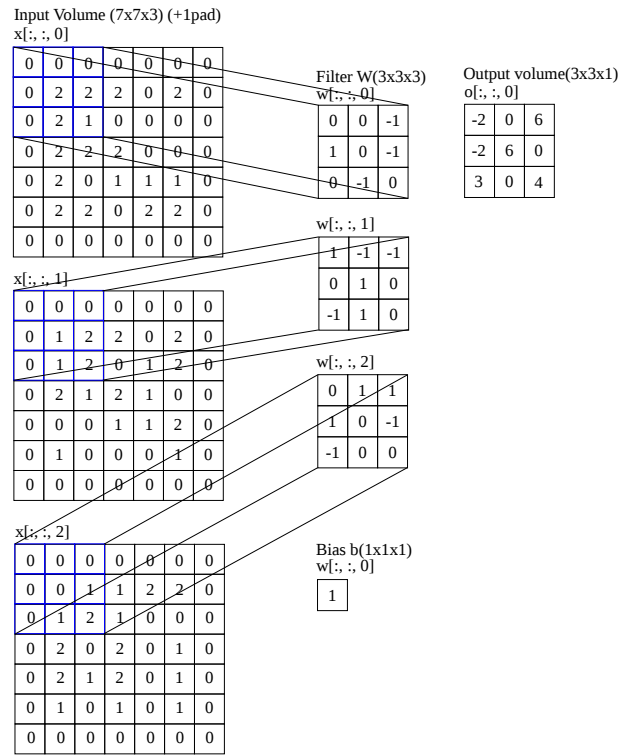


Figure 2.9: Example of convolution

where high values after this operation means that the filter pattern is similar to some region in the image (Figure 2.10)

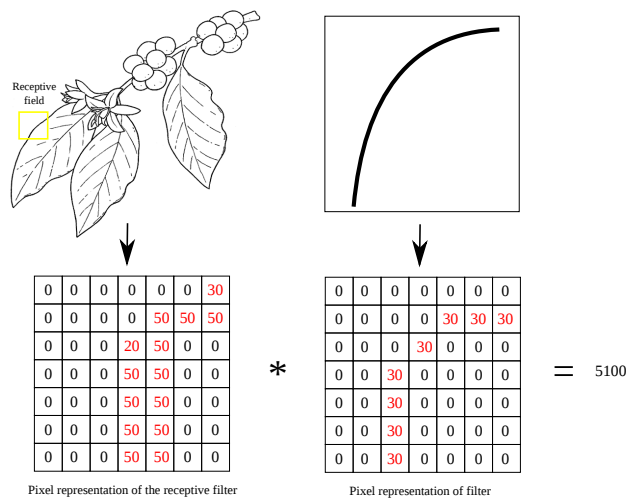


Figure 2.10: Example of pattern recognition

### 2.2.4.2 Pooling layer

A pooling layer is usually inserted after a convolutional layer and the aim of this layer is to reduce the dimensionality of each feature map, reducing the number of parameters and computation in the network maintaining the most important information and avoiding overfitting. The main concept of this layer is that the exact location of a feature is not so important than its rough location in relation to other features which ensures a translational invariance. The pooling layer operates by sliding a filter along the height and width, applying a nonlinear function. As well as convolutional layer, the steps are controlled by the stride and it is also possible to use zero-padding, the most common use is a pooling layer with 2x2 size filter and stride 2, which downsamples every depth slice of the input by 2, discarding 75% of the activations. As a good practice, pooling layers should not contain large filters because large filters can cause destructive effects. There are many functions used in the pooling layer, among them, the max pooling is the most common, this function consist in to extract the maximum value in each subregion (Figure 2.11).

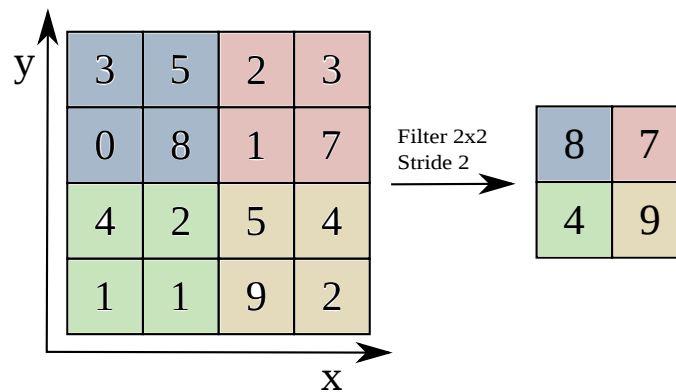
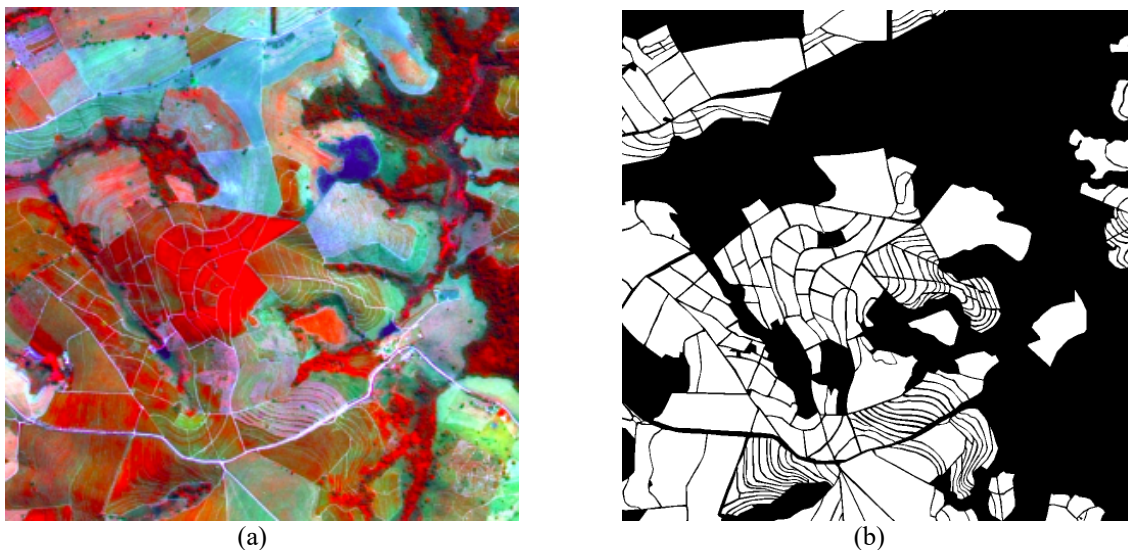


Figure 2.11: Example of max pooling

### 3. Related Work

Semantic segmentation A common process used in remote sensing area is the image segmentation. This process aims to separate different objects that belong to an image in relation to some property, such as texture, pixel values, shape, among others. The segmentation process is called semantic segmentation when the image is divided into objects that belong to some category (class) with some semantic meaning such as coffee and non-coffee.

In remote sensing, this is an important process since we are interested in specific objects (classes), in this way, a good segmentation algorithm, that is, that does not have a high misclassification rate, is essential to help in further analysis. To ensure the quality of a segmentation method, evaluations are performed, manually or in an automatic and supervised way. In the manual way, a human expert in the area, in which the segmentation method was applied, visually analyzes the result of the segmentation, while, in the supervised way, a manually segmented image denoted groundtruth is compared with the result of the segmentation obtained by the method. In Figure 3.1, we can see an example of a segmented image in two different classes: coffee and noncoffee



**Figure 3.1:** Example of semantic segmentation (a) original image and (b) original image classified into coffee (white) and non-coffee (black)

Over the years, with the increase in the availability and quality of satellite images, many works related to semantic segmentation have appeared. In [Slavkovikj et al., 2015] it was



proposed an approach that uses a 220-band hyperspectral image as input to a deep learning architecture known as convolutional neural network (CNN) to segment a hyperspectral image into 16 different classes. In this approach, the features are learned by the use of the spectral pixel and its neighbors that are used in order to capture spectral and spatial information at the same time.

In contrast to [Slavkovikj et al., 2015], in [Makantasis et al., 2015], the hyperspectral image was first subjected to a dimensionality reduction by an algorithm called PCA [Jolliffe, 2002] that extracts a reduced image representation maintaining a minimum of 95 % of the initial information. This reduced image is then used by a CNN architecture to extract relevant features.

In [Yao et al., 2016], they created a framework to perform the segmentation task on high-resolution images, this structure consists of a combination of high-level features, which is a feature with a high level of abstraction, learned by stacked discriminative sparse autoencoder and supervised feature transferring using labeled tiles.

### **3.2 Spatial feature extraction**

The development of algorithms for spatial extraction information is a hot research topic in the remote sensing community [Benediktsson et al., 2013]. It is mainly motivated by the recent accessibility of high spatial resolution data provided by new sensor technologies.

Even though many visual descriptors have been proposed or successfully used for remote sensing image processing [Yang and Newsam, 2008; dos Santos et al., 2010; Bouchiha and Besbes, 2013], some applications demand more specific description techniques. As an example, very successful low-level descriptors in computer vision applications do not yield suitable results for coffee crop classification, as shown in [dos Santos et al., 2014]. Despite this, higher accuracy rates can be obtained by the combination of complementary descriptors that exploits late fusion learning techniques. Following this trend, many approaches have been proposed for selection of spatial descriptors in order to find suitable algorithms for each application [Faria et al., 2014; Cheriyyadat, 2014; Tokarczyk et al., 2015].

Cheriyyadat [2014] proposed a feature learning strategy based on Sparse Coding, which learned features from well-known datasets are used for building detection in larger image sets. Faria et al. [2014] proposed a new method for selecting descriptors and pattern classifiers based on

rank aggregation approaches. Tokarczyk et al. [2015] proposed a boosting-based approach for the selection of low-level features for very-high resolution semantic classification. Tsai and Chen [2017] proposed a spectral domain method to extract structural features of row-planted coffee fields using the Fourier transform.

Despite the fact that the use of Neural Network-based approaches for remote sensing image classification is not recent [Barsi and Heipke, 2003], its massive use is recently motivated by the study on deep learning-based approaches that aims at the development of powerful application-oriented descriptors.

Many works have been proposed to learn spatial feature descriptors [Firat et al., 2014; Hung et al., 2014; Xie et al., 2014; Zhang et al., 2015]. Firat et al. [2014] proposed a method that combines Markov Random Fields with CNNs for object detection and classification in high-resolution remote sensing images. Hung et al. [2014] applied CNNs to learn features and detect invasive weed. In [Xie et al., 2014], the authors presented an approach to learn features from Synthetic Aperture Radar (SAR) images. Zhang et al. [2015] proposed a deep feature learning strategy that exploits a pre-processing saliency filtering. Moreover, new effective hyperspectral and spatio-spectral feature descriptors [Romero et al., 2014; Midhun et al., 2014; Chen et al., 2014; Tuia et al., 2015] have been developed mainly boosted by the deep learning growth in recently years.

### **3.3 Coffee crop mapping**

As mentioned in Chapter 1, the recognition of coffee crops is not a trivial task and this fact is related to the great amount of patterns found in this areas. This diversity originates mainly due to differences in climates, relief and the fact that the coffee crops is not seasonal, which allows plants of different ages within the same crop.

To deal with the coffee recognition problem, in [Santos et al., 2010], a genetic programming (GP) approach was proposed. This approach combines the similarities of descriptors to recognize coffee crops. In this approach, the entire image is partitioned into sub-images and for each sub-image features are extracted by using descriptors.

Some of these sub-images that belong to different classes (coffee and non-coffee) are used as sample, for GP training, which aims to discover similarity functions for each class. These

similarity functions are used to classify the test sub-images that are classified based on the similarities encountered with the training set images. After classification of all sub-images, the classes assigned to these are used as seeds that will guide the watershed-based algorithm [Lotufo and Falcao, 2002] to segment the image semantically.

Faria et al. [2012] built a framework for fusion of classifiers using support vector machine (SVM) [Hearst et al., 1998], this framework is composed of  $C$  classifiers constructed by combining a set of learning methods (e.g. Decision Tree, Naive Bayes) with a set of image descriptors (e.g. Color Histogram). The performance of each classifier was computed in a validation set  $V$  and stored in a matrix  $M_v$  that can be used to train fusion technique that requires prior training (e.g., SVM).

In addition of [Faria et al., 2012] approach, in [Faria et al., 2014] the authors proposed a rank aggregation method to select the most suitable classifiers based on both the diversity and the effectiveness performance of classifiers. In [Ferreira et al., 2016], a boosting-based approach was proposed that uses different types (i.e., very high spatial and hyperspectral imagery) of data in a complementary way to improve the recognition of coffee. For the spatial image, the image is segmented and for each segmented region a combination of descriptors is used to generate features, whereas in the hyperspectral image only the pixel signature is used. The images are mapped to create a feature with both images information to train the boosting that is used to classify the entire image.

A very common problem in remote sensing is the great number of information of different shapes and sizes that composes the images, which makes the choice of a suitable scale to extract relevant information a very difficult task. To deal with this problem, some approaches use a resource called context window that uses the neighborhood around a pixel of some interest class to extract features.

To correctly extract features, the size of the neighborhood must be adjusted to best fit the shape of the class. For example, suppose a class of small objects, if a large context window is used, this window may contain other objects that are not related to the class of the object. These objects may become noise making it difficult to extract relevant features [Nogueira et al., 2016], however, if the object contains shapes larger than the context window, information may become incomplete or useless, which justifies the use of a larger window as we can see in the spatial image of IEEE GRSS Data Fusion Contest 20141 in Figure 3.2.



**Figure 3.2:** Example of use of different sizes context windows. Blue dashed box is an example of a large context window, while the red dashed box is an example of a small context window.

To avoid both, some methods use a multi-scale approach. In [Nogueira et al., 2015], it was proposed a method called Cascade Convolutional Neural Network Model (CCNN), which is a hierarchical model composed of three levels that employ the same architecture for each level, differentiating only in the classification layer and training data. In this method, an input is sent to the first level that can classify the input or not. If an input is not classified, it is subdivided into four parts and resized to the size of the input of the first level and sent to the second level that does the same process of the first level. If there are still unclassified data, these are submitted to the last level that will classify the remaining inputs into some class.

Dos Santos et al. [2013] proposed a method called multiscale classifier (MSC) to classify remote sensing images based on the Adaboost [Schapire, 1999] algorithm that creates a strong classifier of weak classifiers. In this approach, the image is segmented on several scales using the algorithm from [Guigues and Le Men, 2003] algorithm, each segmentation level is composed of a set of regions that are used to calculate features using different types of descriptors. These features are used to construct an  $F(p)$  classifier that is a linear combination  $MSC(p)$  of  $T$  weak classifiers  $h_t(p)$  each related to a specific scale and feature type.

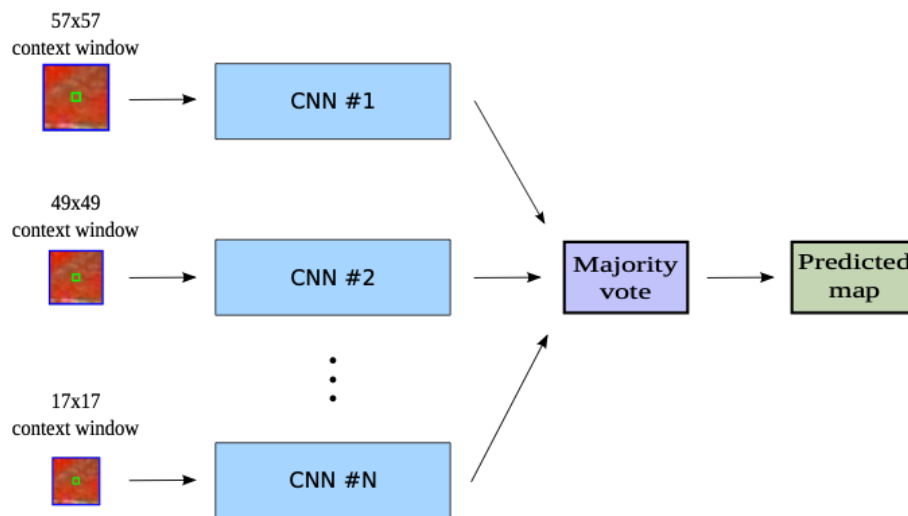
The final classifier is constructed at the end of  $N$  iterations where, at each iteration, each learner creates a weak classifier that decreases the expected classification error of the combination. The error is decreased by using training data where each sample has a weight, which is the same for all data at the beginning. At each iteration, the weight of misclassified samples is increased, which forces weak learners to focus on hard samples in the next iteration. The algorithm then selects the weak classifier which further decreases the error.

A drawback of the MSC is that the method does not guarantee the representation of all the scales in the final result. To solve this problem, the authors proposed a variation called Hierarchical

multiscale classifiers (HMSC), which is constructed using a hierarchy that relates each scale of the model. The HMSC is constructed by selecting weak classifiers for each scale, starting from highest to lowest scales. Each weak learner is trained with only the samples related to the current scale and at the end of each step, only the most difficult samples are selected, limiting the training set used in the next step. For each scale, the weak learner produces a set  $H_\lambda$  of weak classifiers. The HMSC is a combination of the set of weak classifiers  $S_\lambda(p)$  selected for each  $\lambda$  scale.

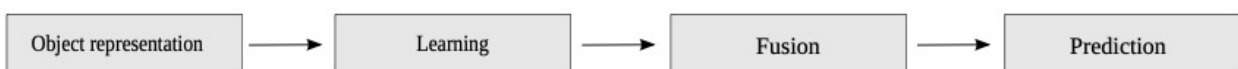
## 4. Methodology

Our approach for creating thematic maps was based on the notion of context windows proposed in [dos Santos et al., 2012] and consists in combining Convolutional Neural Networks (CNNs) that work with different scales. The proposed method extracts, at the same time, context windows of different sizes of a region that belongs to an image to send, as input, to different CNN scales. Each CNN output class probability vectors that are combined to generate only one vector. From this vector, the class that has the highest probability is selected as the predicted class. It was projected to work with a binary class mapping scenario that contains the classes: coffee and non-coffee. Figure 4.1 illustrates an example of the proposed methodology to combine different CNNs in this work.



**Figure 4.1:** The multiscale CNN structure was designed to receive as input, at the same time, context windows of a very high spatial resolution (VHS) image with different size relative to the same region. Each classifier produces a vector that contains a probability for the input to be coffee or non-coffee that sums 1. The probability vectors are fused by the sum of all vectors generating a single resulting vector, the value of the highest probability class in this vector is selected to be the predicted class.

The proposed methodology is composed of four main steps: object representation, learning, fusion and prediction (Figure 4.2).

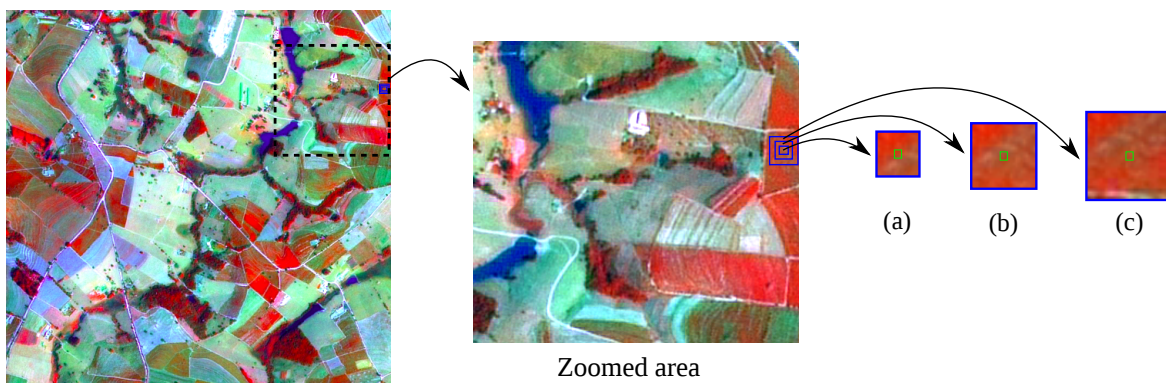


**Figure 4.2:** Steps of the proposed methodology

In the process of object representation, the image was subdivided into several context windows. Some context windows are used as input for the CNN to build a data model in the learning step that will learn how to extract representative features to distinguish classes. Next, the fusion step was used to combine the output of each CNN at the decision level by majority voting, giving rise to the multi-scale approach that was used in the last step, to predict coffee crops.

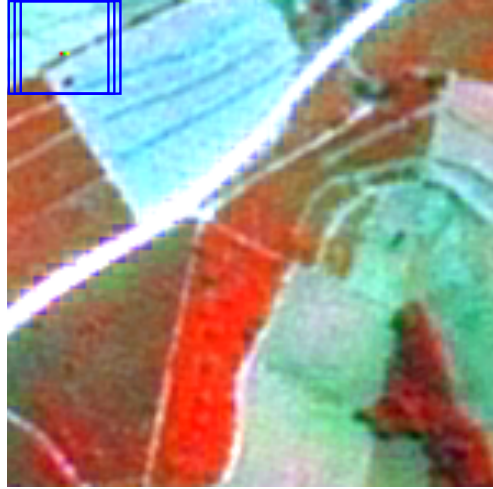
#### 4.1 Object representation

In remote sensing, one of the most common way for classification is at pixel level, which assigns each pixel to a class based on their values. As mentioned in Section 2.1, in high resolution images there are many objects rich in detail, in this way, the use of only one pixel to represent an object belonging to a specific class may not be sufficient to provide relevant information. A common way to deal with this problem is to use a context window that is a set of pixels around a central pixel that aim to extract different and complementary information as we can see in Figure 4.3.



**Figure 4.3:** Example of context window use

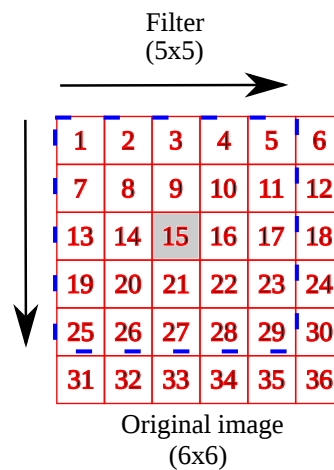
Since all pixels of the image were used, it was allowed overlapping context windows (Figure 4.4).



**Figure 4.4:** Examples of 3 context window with overlap

Since a context window is designed as an area around a central pixel, when the center pixel belongs to some extremity of the image, it may not contain neighbors in some direction. To handle such cases, it was decided to padding the image by a mirroring scheme.

For example, suppose an image of  $6 \times 6$  pixels and we want to use a context windows with size  $5 \times 5$  pixels. In this case, without padding it is possible to use only 4 pixels to extract features (Figure 4.5).



**Figure 4.5:** Example of context window use without padding



The padding was done by calculating the size that the image must have to support the use of the context window of the desired size (Equation 4.1). This new dimension was then achieved by the image by the mirroring of the extremities.

$$width_{new} = width_{old} + \left\lfloor \frac{filter_{width}}{2} \right\rfloor * 2 \quad (4.1)$$

$$height_{new} = height_{old} + \left\lfloor \frac{filter_{height}}{2} \right\rfloor * 2$$

For the above example, for all pixels to be used, the dimensions of the image should be  $10 \times 10$  pixels. This new dimension can be achieved by adding 2 rows and 2 columns on each side of the image. The extremities mirroring is created by inverting the column or line positions of the original image (Figure 4.6).

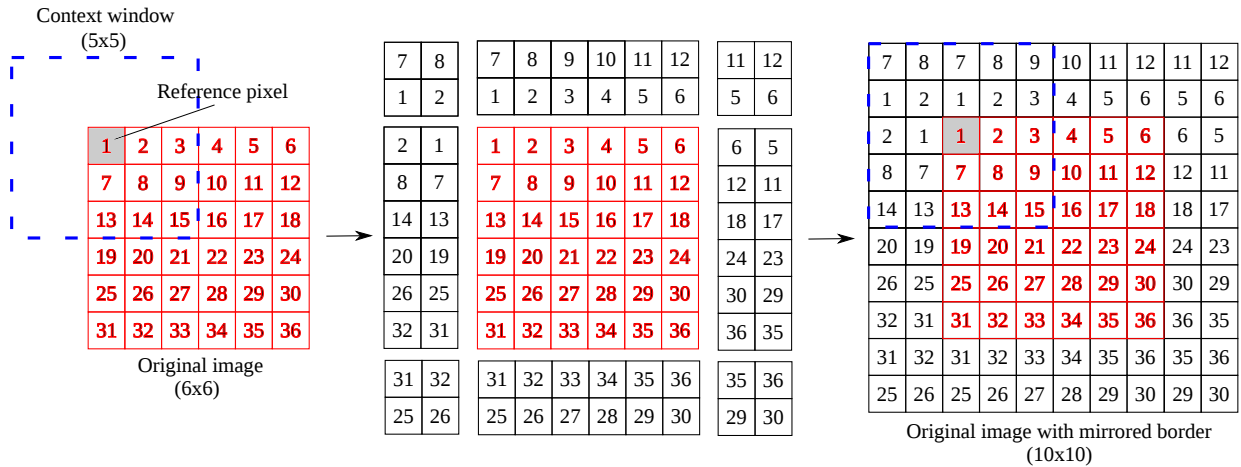


Figure 4.6: Example of context window use with padding

To ensure that each context window follows the same distribution, all data ( $X$ ) was normalized to mean 0 and standard deviation 1  $\mathcal{N}(0,1)$  and this process was performed by applying the following equation in the original image:

$$X_{Normalized} = \frac{X - \mu}{\sigma} \quad (4.2)$$

where  $\mu$ ,  $\sigma$  is the mean and standard deviation of the training data, respectively.

## 4.2 Learning

It is a great advantage of CNNs when compared to conventional methods, the use of multiple layers in a hierarchical manner (Section 2.2.4) to learn extracting representative features to recognize unseen data. The features were obtained by combining convolutional layers and pooling layers.

In the convolutional layer, different image patterns called feature maps were extracted by the filters, these feature maps were reduced by the pooling layer that preserves the location of the relevant features and makes them invariant. As the network becomes deeper, smaller, more representative and abstract the features become. The final features generated by CNN are those of the last layer that are connected, usually, by a fully connected layer that will learn to relate them.

The extracted features of a learned model are used to recognize unseen samples related to the classes in which the model was trained. Initially, the extracted features are not representative enough to correctly recognize unseen samples. In this way, the feature extraction was tuned by a training process that is used to adjust the weights of the CNN to make extracted features more representative. This process was performed by constructing a training data with  $N$  samples that are composed of a pair  $(x_i, y_i)$  where  $x_i$  is a context windows and  $y_i$  is it label. The constructed training data were randomly arranged and were used as input to each CNN separately.

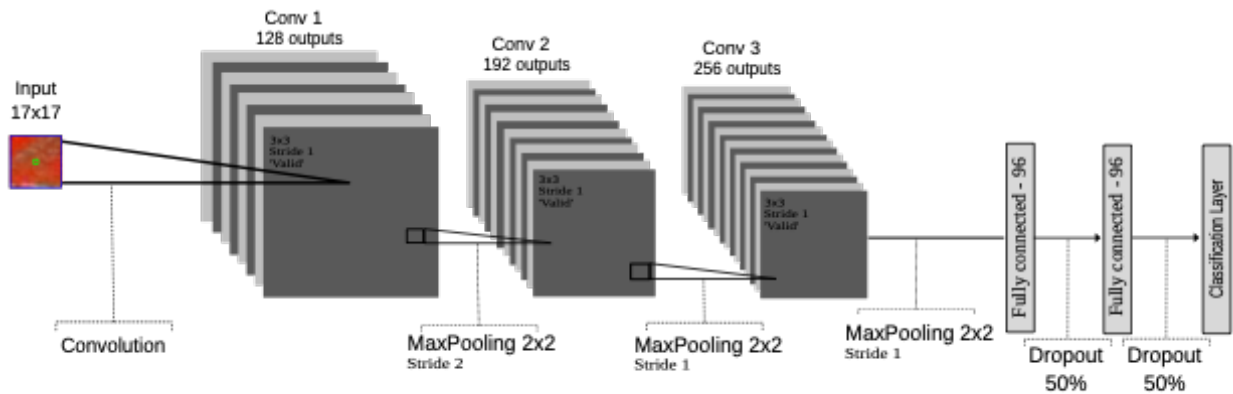
The CNN error was calculated using a validation data that is a set of unseen samples that is used to see the generalization capability of the network. The training process was performed on each CNN separately and was maintained until the error rate remains stable with close validation and training errors. This process was done because, if the validation error is high while the training error is low, this indicates the occurrence of overfitting, i.e., the CNN has memorized the training data.

In this work, several CNNs were proposed, each of them constructed to use a specific context window size that was chosen based on the average size of the patterns found in the coffee crops. For the datased used in this work, each pixel is equal to 2.5m, so a cropsiz of  $17 \times 17$  is equivalent to an area of  $1806.25m^2$ .

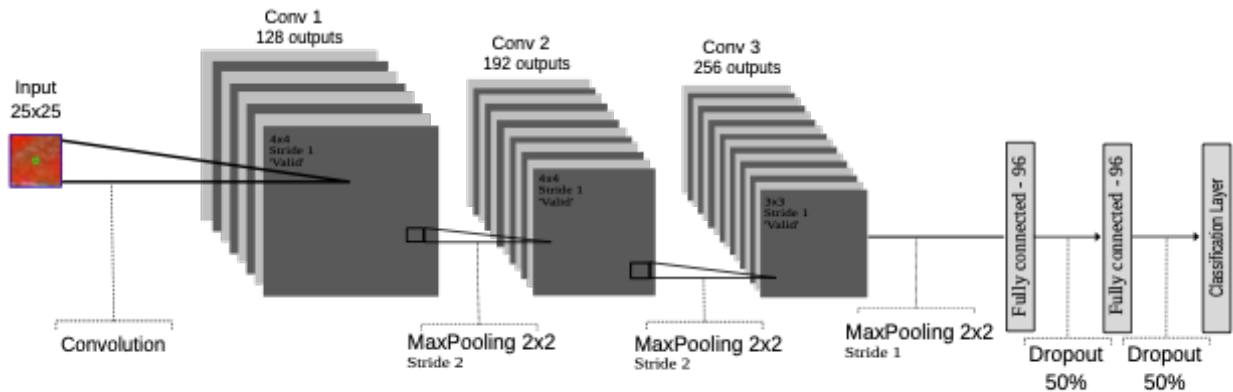
In Table 4.1, it is summarized each CNN architecture used in this work and, Figures 4.7 - 4.16 show each one of the architectures.

Blocks (conv + pool)	17 x 17 (1806.25m <sup>2</sup> )	25 x 25 (3906.25m <sup>2</sup> )	33 x 33 (6806.25m <sup>2</sup> )	41 x 41 (10506.25m <sup>2</sup> )	49 x 49 (15006.25m <sup>2</sup> )	57 x 57 (20306.25m <sup>2</sup> )
3	x	x	x			
4			x	x	x	
5					x	x
6						x

**Table 4.1:** Summary of architectures used



**Figure 4.7:** CNN # 1: architecture with  $17 \times 17$  pixels context windows as input and 3 blocks of convolution and maxpooling



**Figure 4.8:** CNN # 2: architecture with  $25 \times 25$  pixels context windows as input and 3 blocks of convolution and maxpooling

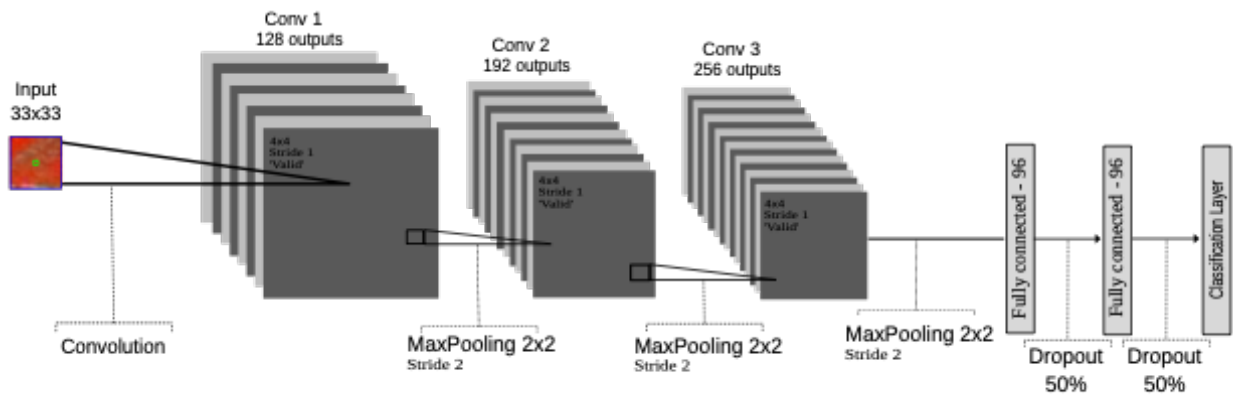


Figure 4.9: CNN # 3: architecture with  $33 \times 33$  pixels context windows as input and 3 blocks of convolution and maxpooling

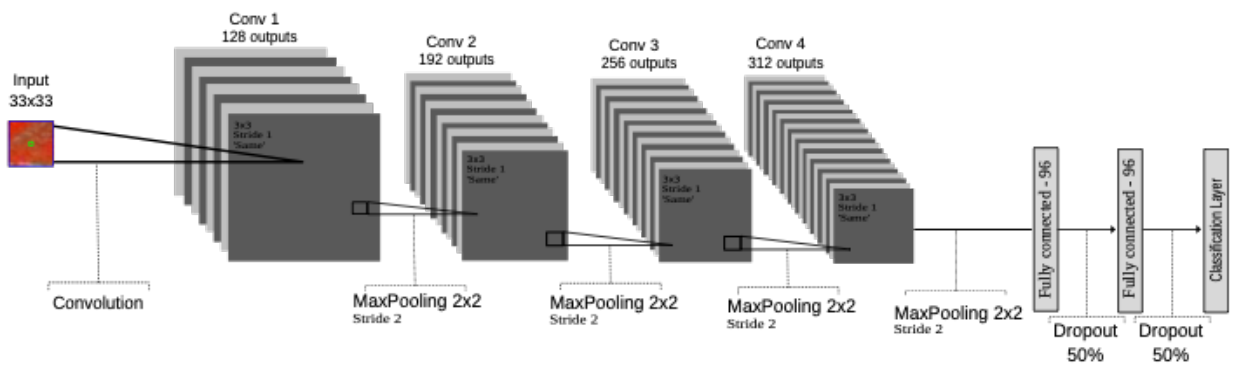


Figure 4.10: CNN # 4: architecture with  $41 \times 41$  pixels context windows as input and 3 blocks of convolution and maxpooling

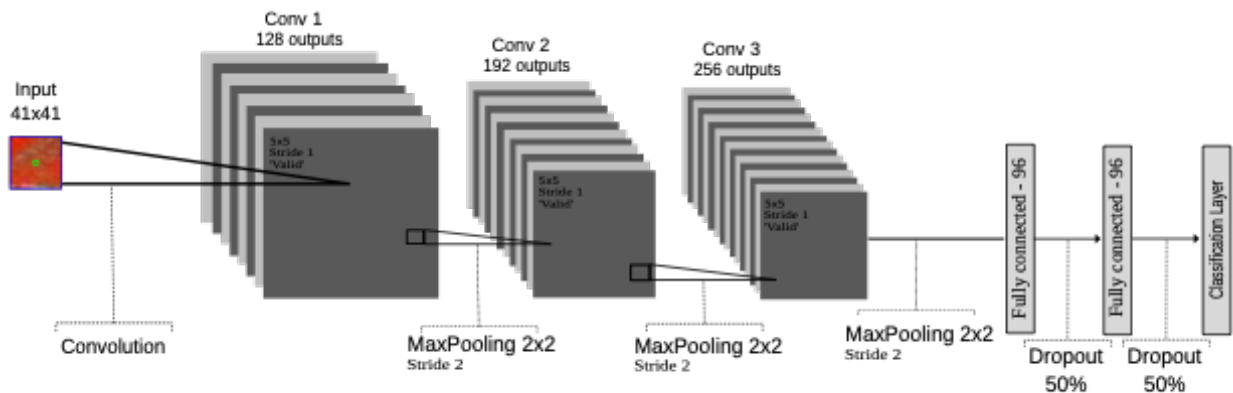


Figure 4.11: CNN # 5: architecture with  $33 \times 33$  pixels context windows as input and 4 blocks of convolution and maxpooling

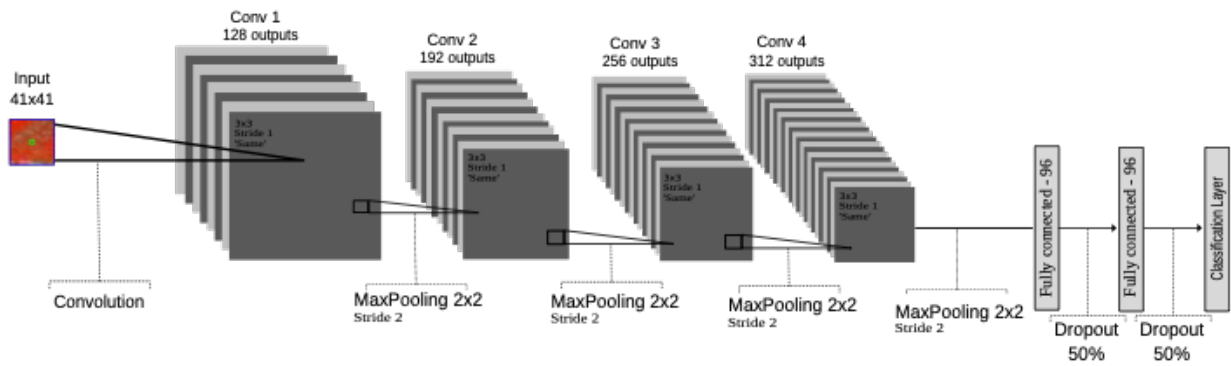


Figure 4.12: CNN # 6: architecture with  $41 \times 41$  pixels context windows as input and 4 blocks of convolution and maxpooling

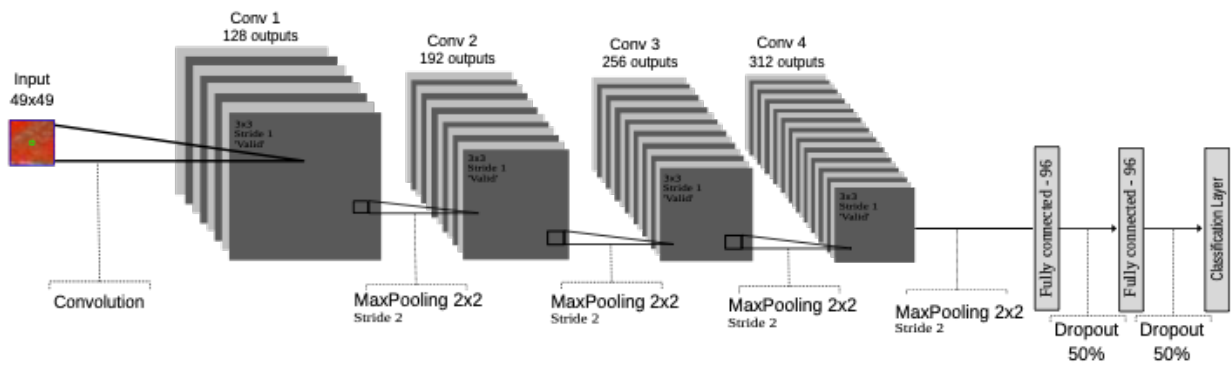


Figure 4.13: CNN # 7: architecture with  $49 \times 49$  pixels context windows as input and 4 blocks of convolution and maxpooling

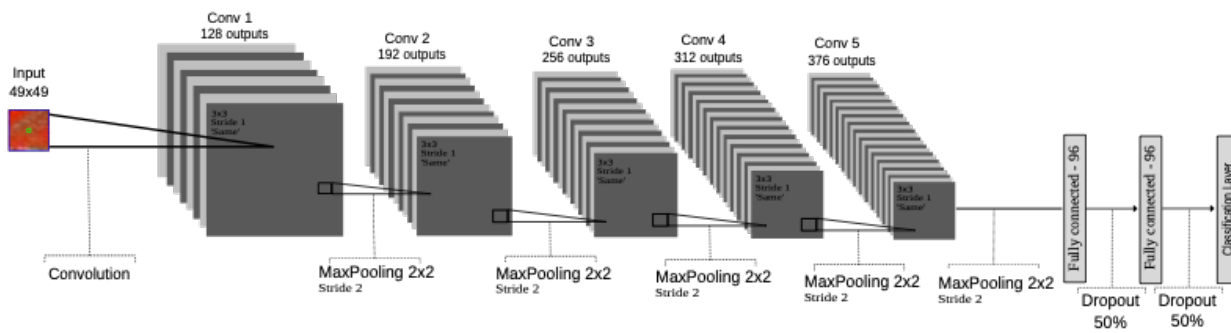
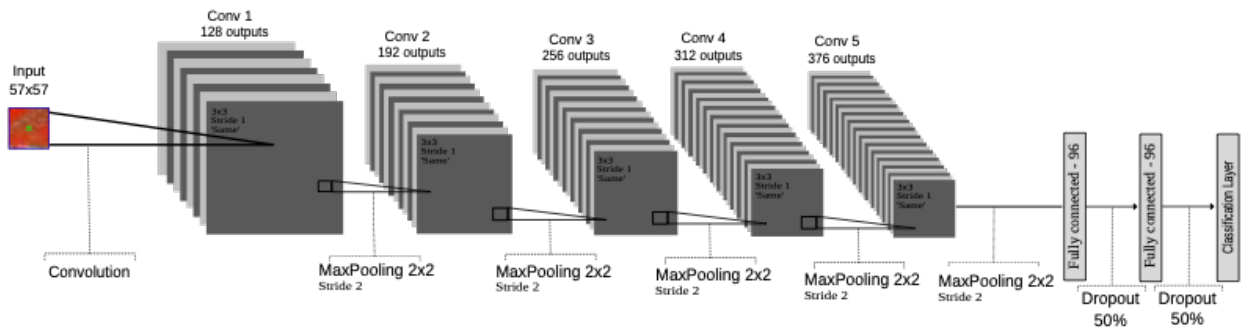
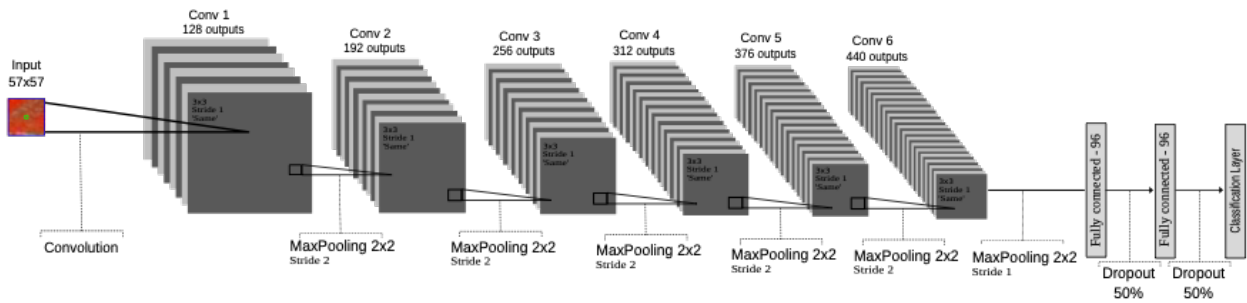


Figure 4.14: CNN # 8: architecture with  $49 \times 49$  pixels context windows as input and 5 blocks of convolution and maxpooling



**Figure 4.15:** CNN # 9: architecture with  $57 \times 57$  pixels context windows as input and 5 blocks of convolution and maxpooling



**Figure 4.16:** CNN # 10: architecture with  $57 \times 57$  pixels context windows as input and 6 blocks of convolution and maxpooling

The architectures proposed in this work were based in [Nogueira et al., 2015] that also proposed an approach to recognize coffee crops. In contrary of [Nogueira et al., 2015] that used only one CNN to build a three stage network to deal with the scale problem, in this work, we used a combination of several CNNs that extracts different and complementary information at distinct scales. Each CNN architecture used in this work was built based on a specific context window size. In this way, objects with more complex patterns required large window size. Consequently, the large context window size was, more complex the required CNN was, that is, more layers, filtering and pooling operations.

### 4.3 Fusion

The fusion was performed by using the last layer (classification layer) output. Given a set of CNNs, each CNN provides a probability vector  $P$  with length equal the number of classes, the vectors are summed generating a resultant vector, within the class with the highest value selected as the class of the pixel to be predicted (Figure 4.17).

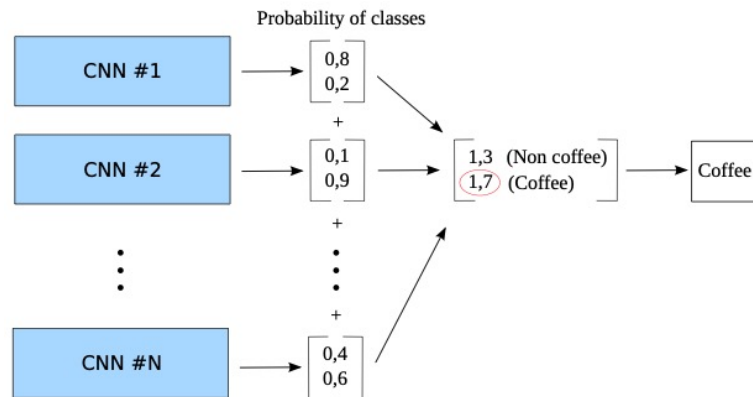


Figure 4.17: Example of CNN combination

### 4.4 Prediction

Once all CNNs were trained, the scales were used to build the multi-scale approach and the thematic map generation process was done by extracting context windows for each selected scale in the image to be classified and sent to their respective networks. Each network generates a probability vector that contains the probability of the context window belong to some class. The probability vectors are combined to generate a resultant vector in which the class with largest probability value determines the classification of a pixel of the image. The prediction process is shown in Figure 4.18 for a fusion of three CNNs.

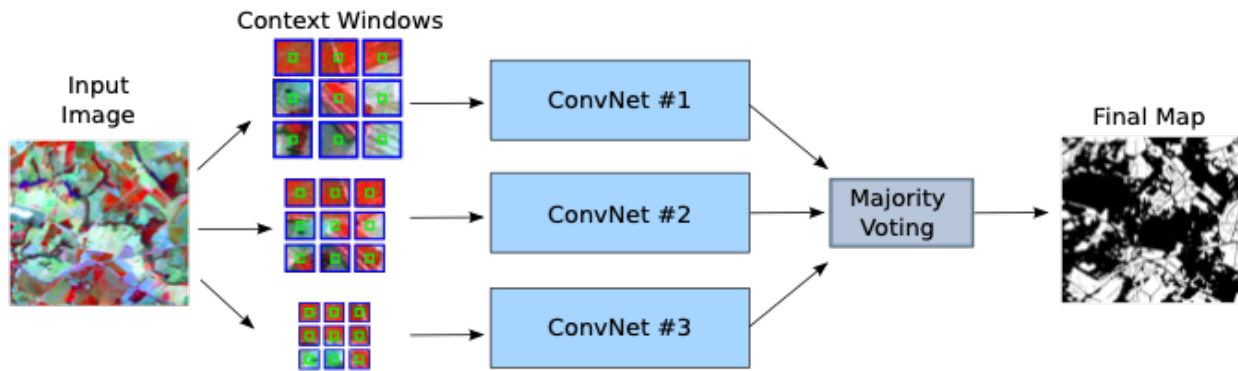


Figure 4.18: Example of prediction process.

## 5. Experimental evaluation

In this section, we present the experiments that were performed to validate our method. We have carried out experiments in order to address the following research questions: (1) is multiple scale fusion more effective than individual CNNs for semantic segmentation of coffee crops? (2) What is the best combination of architectures? (3) Are the proposed methods effective in the coffee crop recognition problem when compared to the baselines?

### 5.1 Setup

- Evaluation metric: we used the Overall Accuracy and Cohen's Kappa described in Section 5.1.3.
- Feature extraction: we used several CNNs to work with different scales to encode spatial information about the classes coffee and non-coffee.
- Training: All CNN were trained using SGD with an initial learning rate 0.001 and momentum 0.9, the learning rate is decreased in an exponential way using a decay rate of 0.1. The training data were shued in each epoch and divided in k mini-batches of 250 samples and the weights were initialized using xavier [Glorot and Bengio, 2010].
- Baselines: We compared the proposed method against two approaches that follows the traditional three-main-step strategy: (i) segmentation, (ii) feature extraction and, (iii) classification. These approaches, named here as MSC-Boost and HMSC-Boost, described in Section 2.1, are based on boosting of classifiers and combine features from multiple segmentation scales [dos Santos et al., 2012]. In our experiments, both approaches were



implemented to consider features extracted from five segmentation scales. The main difference between them is that MSC-Boost consider all regions segmented over the segmented scales while HMSC-Boost starts from the coarse regions and use the other scales in sequence as refinement steps. We have used the same engineered features of the original paper dos Santos et al. [2012]. For a better comparison, we also included results with a SVM [Hearst et al., 1998] with radial basis function (RBF) [Broomhead and Lowe, 1988] and the best engineered descriptor in the best segmentation scale as reported in [dos Santos et al., 2012].

- Implementation details: The proposed approach was implemented by using the Tensorow [Abadi et al., 2016] framework. This framework is more suitable due to its support to parallel programming using CUDA, a NVIDIA parallel programming based on graphics processing units. The complete set of experiments was performed on a 64 bits Intel i7 4960X machine with 3.6GHz of clock and 64GB of RAM memory. We used the following GPUs: a GeForce GTX770 with 4GB of internal memory and a GeForce GTX Titan X with 12GB of memory, both under a 7.5 CUDA version. Ubuntu version 14.04.3 LTS was used as operating system. The CNNs and their parameters were adjusted by considering a full set of experiments based on [Nogueira et al., 2015].

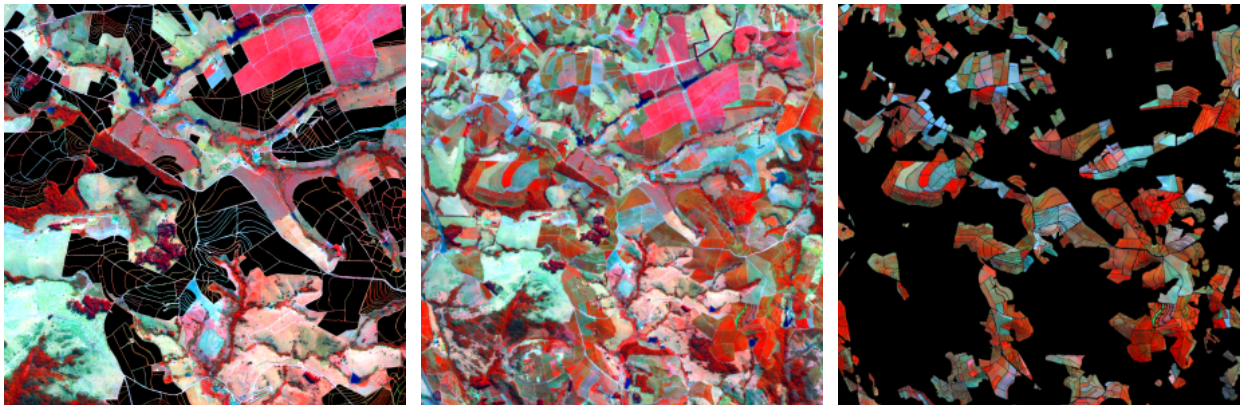
### 5.1.1 Dataset

The dataset used in this work is a composition of scenes taken by using the Satellite Pour l'Observation de la Terre (SPOT) 5, which offers a higher resolution of 2.5 to 5 meters in panchromatic mode in 2005 over Monte Santo de Minas county, State of Minas Gerais, Brazil. The images were obtained through collaboration with the researcher in agriculture from the Núcleo Interdisciplinar de Planejamento Energético da Unicamp (NIPE), Dr. Rubens Lamparelli, who maintains direct contact with the Cooperativa de Cafeicultores de Guaxupé (Cooxupé).

The area covered by the image is a traditional place of coffee crops, characterized by its mountainous terrain. In addition to common issues in the area of pattern recognition in remote sensing images, these factors add further problems that must be taken into account. In mountainous areas, spectral patterns tend to be affected by the topographical differences and by interferences

generated by shadows. This dataset provides an ideal environment for multi-scale analysis, since the variations in topography require the cultivation of coffee in different crop sizes.

Another problem is that coffee is not an annual crop. This means that, in the same area, there are crops of different ages. In terms of classification, we have several completely different patterns representing the same class while some of these patterns are much closer to other classes (Figure 5.1).



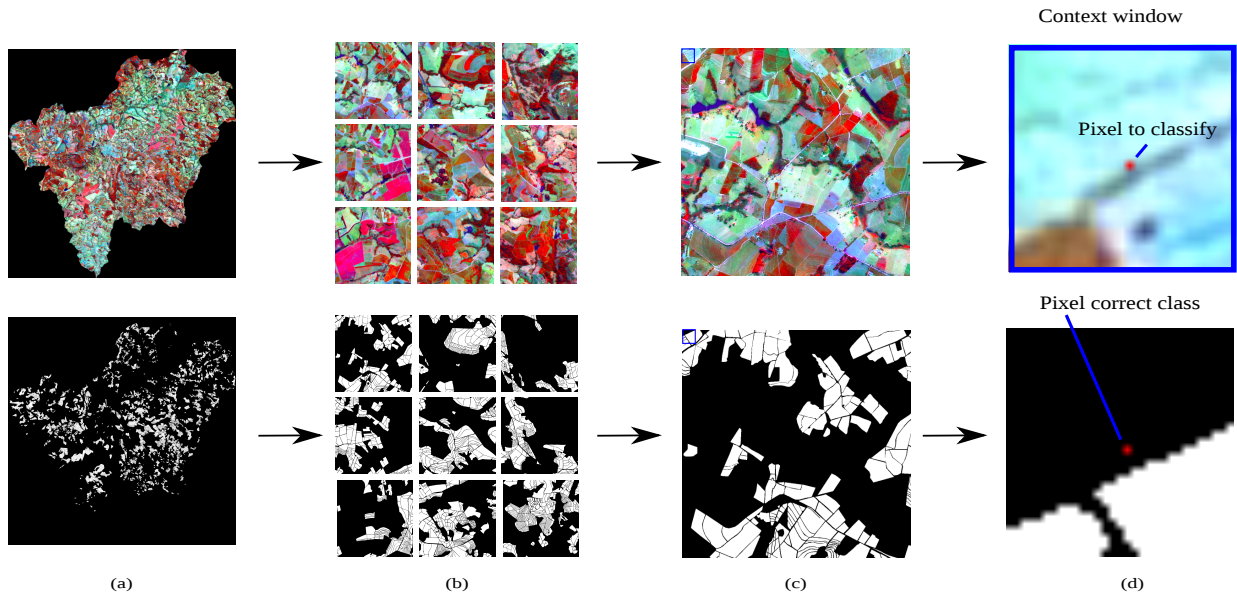
(a) Non-coffee

(b) Original image

(c) Coffee

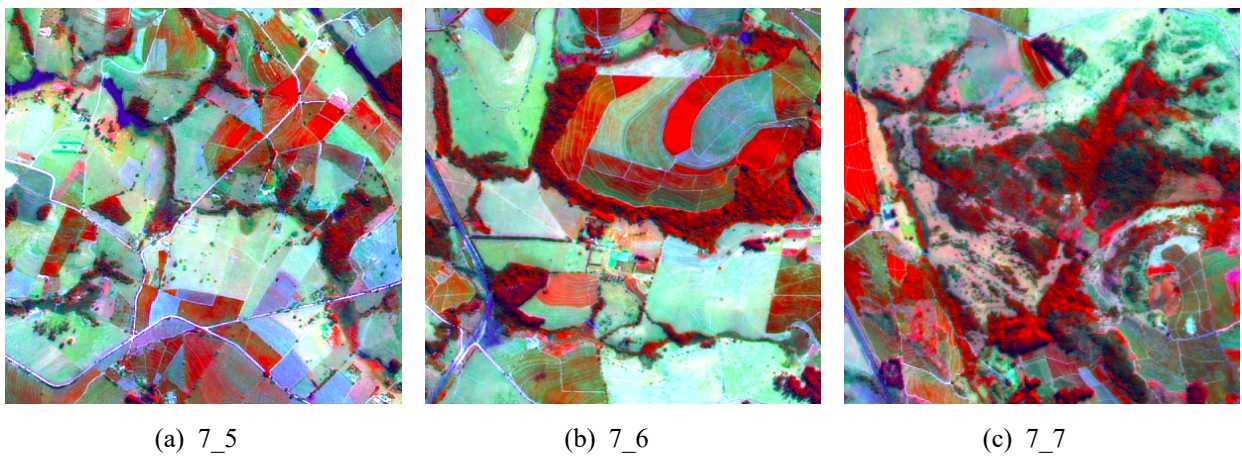
**Figure 5.1:** Intravariance class in dataset

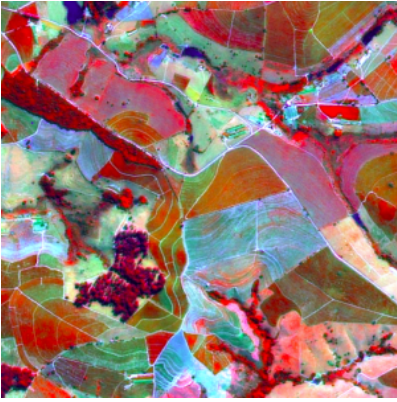
The dimensions of the image used are  $3000 \times 3000$  pixels and to allow the comparison with the baselines, the dataset was divided into a grid of  $3 \times 3$ , generating 9 subimages with dimensions equal to  $1000 \times 1000$  pixels (Figure 5.2).



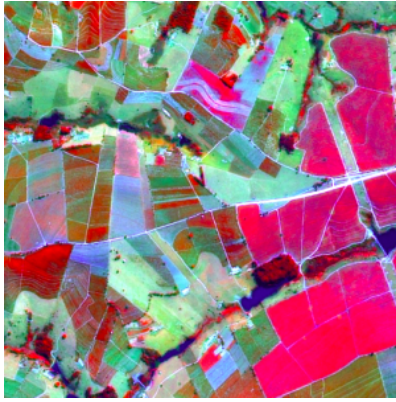
**Figure 5.2:** (a) Original image (b) Original image divided into sub-images of the same size. (c) Example of subimage and a train sample (d) Train sample  $x_i$  and its groundtruth  $y_i$

In Figures 5.3 and 5.4 are show each image and its groundtruth, respectively.

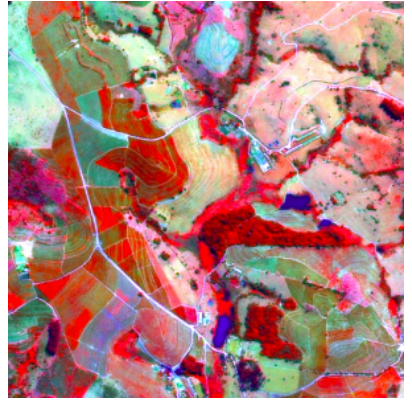




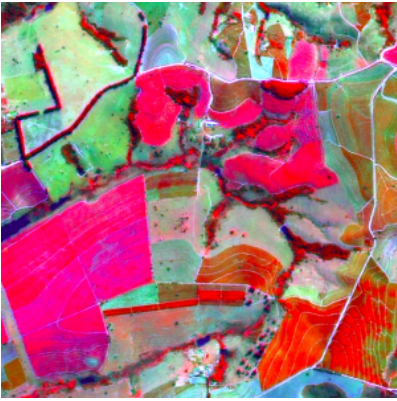
(d) 8\_5



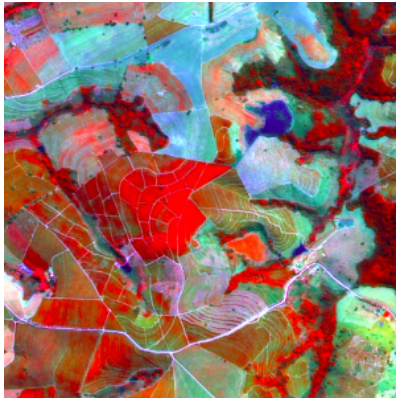
(e) 8\_6



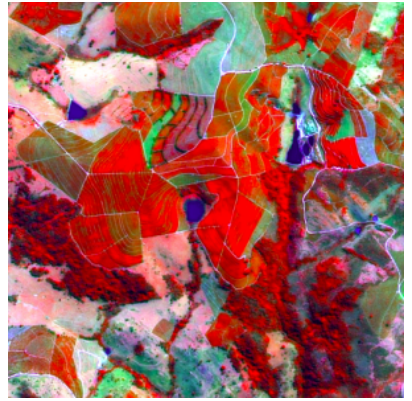
(f) 8\_7



(g) 9\_5

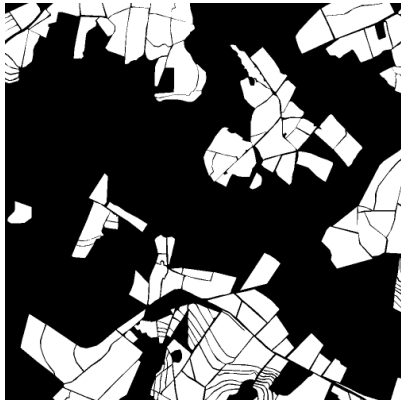


(h) 9\_6



(i) 9\_7

**Figures 5.3:** Subimages of the original images



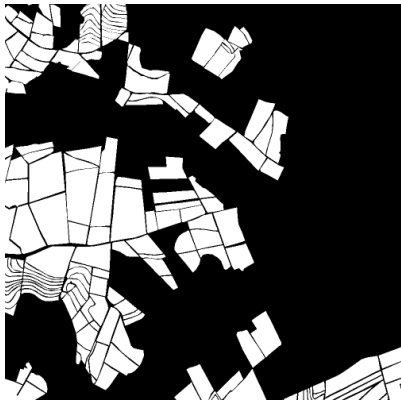
(a) 7\_5



(b) 7\_6



(c) 7\_7



(d) 8\_5



(e) 8\_6



(f) 8\_7



(g) 9\_5



(h) 9\_6



(i) 9\_7

Figures 5.4: Subimages groundtruth of the original images

The informations about the class distribution in each subimage are shown in Table 5.1.

Image	Subimage 1	Subimage 2	Subimage 3	Subimage 4	Subimage 5
Coffee	32,71%	26,52%	14,61%	29,57%	38,66%
Non-Coffee	67,29%	73,48%	85,39%	70,43%	61,34%

Image	Subimage 6	Subimage 7	Subimage 8	Subimage 9	Original Image
Coffee	25,98%	23,06%	42,44%	35,49%	29,89%
Non-Coffee	74,02%	76,94%	57,56%	64,51%	70,11%

**Table 5.1:** Class distribution for each subimage

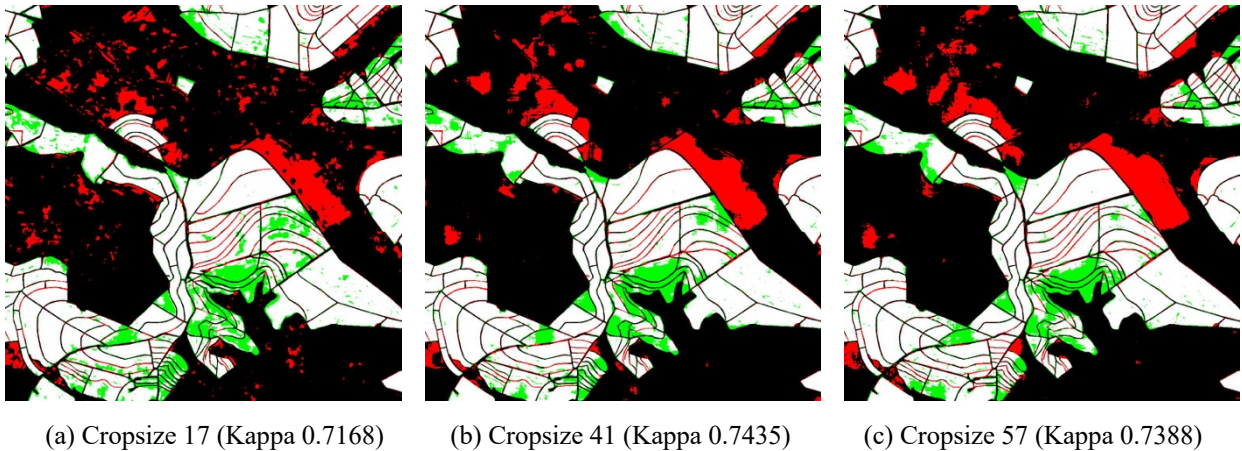
In the experiments, we used 9 different sets of 1 million pixels each to be used for training and classification (prediction step). The results of the experiments described in the following sections are obtained from all combinations (Table 5.15) of the 9 subimages used (6 for training and 3 for classification).

Instance	Train						Test		
1	8_5	9_6	7_7	7_6	7_5	8_7	8_6	9_7	9_5
2	8_7	9_7	7_7	9_6	8_6	8_5	7_6	7_5	9_5
3	8_7	9_6	7_5	8_5	9_5	7_6	9_7	7_7	8_6
4	9_6	9_7	7_5	7_7	8_5	7_6	9_5	8_6	8_7
5	8_7	8_5	9_5	7_5	7_7	8_6	7_6	9_6	9_7
6	7_5	8_7	9_5	8_5	9_6	7_7	8_6	7_6	9_7
7	8_6	8_7	9_7	8_5	7_6	7_5	9_5	7_7	9_6
8	9_7	7_7	8_5	7_6	7_5	9_5	8_6	8_7	9_6
9	8_7	9_5	8_5	7_5	7_7	9_7	8_6	9_6	7_6
10	7_5	8_7	9_6	8_5	8_6	9_5	9_7	7_7	7_6

**Table 5.2:** Instances used to evaluate the proposed approach

### 5.1.2 Evaluated architectures

To construct the proposed approach, several fusions of different CNN architectures were evaluated and each architecture was constructed with respect to a context window size. In this work, we used context windows of sizes 17, 25, 33, 41, 49 and 57 pixels. Context windows with size 17 resulted in a poor classification, while sizes greater than 41 did not offer significant improvement and still made training slower.



**Figure 5.5:** Comparative of small, medium and large cropsizes, where black, white, red, green are true negative (TN), true positive (TP), false positive (FP) and false negative (FN), respectively.

### 5.1.3 Assessment of results

To analyze the results, we computed the overall accuracy and kappa index by using a confusion matrix for each test image.

A confusion matrix is a widely used table in supervised learning that allows an easy analysis of the performance of the algorithm in relation to the prediction results and the expected result. Each column represents the instances in a predicted class while each row represents the instances in an actual class. An example of confusion matrix with 2 classes can be seen in Table 5.3

True label	Predicted Label			
	N =100	Non-coffee	Coffee	Total
Non-coffee		2 (TN)	18 (FP)	20
Coffee		5 (FN)	75 (TP)	80
Total		7	93	100

**Table 5.3:** Example of confusion matrix with 2 classes

Each position in confusion matrix has a definition and is related with the current class and the predicted class:

- TN - True negative - the classifier predicted no coffee and the current class is no coffee
- FN - False negative - the classifier predicted no coffee, but, the current class is coffee
- TP - True positive - the classifier predicted coffee and the current class is coffee
- FP - False positive - the classifier predicted coffee, but, the current class is no coffee

The overall accuracy is defined as the sum of true positive and true negative samples divided by the total number of samples (Equation 5.1).

$$OA = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (5.1)$$

Kappa index is an effective index which compares the accuracy of a trained classifier with the accuracy of a random classifier, commonly used in the RSI classification [dos Santos et al., 2012]. Experiments in different areas show that Kappa can have several interpretations and these guidelines could be different depending on the application. However, Landis and Koch [1977] characterize Kappa values above 0.80 as “almost perfect agreement”, 0.60 to 0.79 as a “substantial agreement”, 0.40 to 0.59 as a “moderate agreement” and below 0.40 as “poor agreement”. Negative Kappa index means that there is no agreement between classified data and verification data (Table 5.4).



Kappa Index	Agreement
< 0	Less than chance agreement
0 - 0.39	Poor agreement
0.40 - 0.59	Poor agreement
0.60 - 0.79	Substantial agreement
0.80 - 0.99	Almost perfect agreement

**Table 5.4:** Kappa index interpretation

The Kappa index (Equation 5.2) is used jointly with overall accuracy (OA), because OA measures the total agreement of all predictions that make this metric less robust to handle unbalanced datasets, since OA can hide poor performance from a classifier in some minority class. In the example of coffee in Table 5.3, we have an unbalanced dataset where the coffee class is equivalent to four times the non-coffee class. In this example, the non-coffee class was not well classified, but the classifier still had high precision (0.77). Meanwhile, the kappa metric has penalized this case of unbalance resulting in a poor result (0.04) for the algorithm.

$$\text{kappa} = \frac{OA - \text{random}_{ACC}}{1 - \text{random}_{ACC}} \quad (5.2)$$

where the  $\text{random}_{ACC}$  is defined as:

$$\text{random}_{ACC} = \frac{(TN + FP) \times (TN + FN) + (FN + TP) \times (FP + TP)}{(TP + TP + FN + FP)^2} \quad (5.3)$$

## 5.2 Results and discussion

### 5.2.1 Multiple $\times$ Individual Scales

In this section, we compared the classification results obtained by using individual scales represented by all CNN architectures (CNN #1 to CNN #10 ) against the combination of scales by using the proposed fusion scheme. Table 5.15 presents the classification results.

Instance #1											
Train images	8 5, 9 6, 7 7, 7 6, 7 5, 8 7										
	17x17 (3 blocks) CNN #1		17x17 (3 blocks) CNN #2		17x17 (3 blocks) CNN #3		17x17 (4 blocks) CNN #4		17x17 (3 blocks) CNN #5		
Test images	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	
8 6	0,865	0,717	0,868	0,723	0,868	0,709	0,874	0,736	0,871	0,729	
9 7	0,870	0,709	0,873	0,715	0,870	0,719	0,876	0,723	0,870	0,709	
9 5	0,919	0,769	0,918	0,764	0,919	0,755	0,928	0,795	0,918	0,768	
Mean	0,885	0,732	0,886	0,734	0,886	0,728	0,892	0,751	0,886	0,735	
Std	0,030	0,032	0,027	0,026	0,028	0,024	0,031	0,039	0,028	0,030	
	41x41 (4 blocks) CNN #6		49x49 (4 blocks) CNN #7		49x49 (5 blocks) CNN #8		57x57 (5 blocks) CNN #9		57x57 (6 blocks) CNN #10		
Test images	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	
8 6	0,877	0,744	0,853	0,701	0,872	0,730	0,875	0,739	0,868	0,725	
9 7	0,880	0,732	0,868	0,710	0,868	0,704	0,877	0,723	0,867	0,702	
9 5	0,931	0,804	0,912	0,763	0,925	0,784	0,925	0,787	0,922	0,780	
Mean	0,896	0,760	0,877	0,725	0,888	0,739	0,892	0,749	0,886	0,735	
Std	0,030	0,039	0,030	0,034	0,032	0,041	0,028	0,033	0,031	0,040	

**Table 5.5:** Results of instance # 1 for all architectures

Instance #2											
Train images	8 7, 9 7, 7 7, 9 6, 8 6, 8 5										
	17x17 (3 blocks) CNN #1		17x17 (3 blocks) CNN #2		17x17 (3 blocks) CNN #3		17x17 (4 blocks) CNN #4		17x17 (3 blocks) CNN #5		
Test images	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	
7 6	0,910	0,773	0,910	0,776	0,886	0,728	0,907	0,773	0,911	0,780	
7 5	0,836	0,622	0,838	0,632	0,818	0,607	0,840	0,638	0,845	0,649	
9 5	0,915	0,762	0,925	0,792	0,895	0,729	0,931	0,809	0,928	0,801	
Mean	0,887	0,719	0,891	0,733	0,866	0,688	0,893	0,740	0,895	0,743	
Std	0,044	0,084	0,046	0,088	0,042	0,070	0,047	0,090	0,044	0,082	
	41x41 (4 blocks) CNN #6		49x49 (4 blocks) CNN #7		49x49 (5 blocks) CNN #8		57x57 (5 blocks) CNN #9		57x57 (6 blocks) CNN #10		
Test images	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	
7 6	0,921	0,802	0,912	0,784	0,918	0,793	0,918	0,793	0,916	0,791	
7 6	0,848	0,653	0,849	0,653	0,842	0,632	0,850	0,656	0,852	0,663	
9 5	0,938	0,825	0,935	0,819	0,939	0,824	0,938	0,824	0,937	0,822	
Mean	0,902	0,760	0,899	0,752	0,900	0,749	0,902	0,758	0,902	0,759	
Std	0,047	0,094	0,045	0,087	0,051	0,103	0,046	0,089	0,044	0,084	

**Table 5.6:** Results of instance # 2 for all architectures

Instance #3											
Train images	8 7, 9 6, 7 5, 8 5, 9 5, 7 6										
	17x17 (3 blocks) CNN #1		17x17 (3 blocks) CNN #2		17x17 (3 blocks) CNN #3		17x17 (4 blocks) CNN #4		17x17 (3 blocks) CNN #5		
Test images	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	
8 6	0,843	0,704	0,874	0,722	0,869	0,709	0,869	0,720	0,869	0,710	
9 7	0,858	0,725	0,932	0,739	0,942	0,777	0,942	0,780	0,936	0,755	
9 5	0,854	0,710	0,861	0,712	0,871	0,730	0,871	0,734	0,867	0,725	
Mean	0,852	0,713	0,889	0,724	0,894	0,739	0,894	0,744	0,891	0,730	
Std	0,008	0,011	0,038	0,014	0,041	0,035	0,041	0,031	0,039	0,023	
	41x41 (4 blocks) CNN #6		49x49 (4 blocks) CNN #7		49x49 (5 blocks) CNN #8		57x57 (5 blocks) CNN #9		57x57 (6 blocks) CNN #10		
Test images	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	
8 6	0,879	0,730	0,867	0,708	0,872	0,715	0,873	0,716	0,871	0,712	
9 7	0,947	0,792	0,934	0,754	0,947	0,794	0,947	0,789	0,948	0,792	
9 5	0,878	0,745	0,861	0,715	0,879	0,748	0,881	0,752	0,873	0,736	
Mean	0,901	0,755	0,887	0,726	0,899	0,752	0,901	0,752	0,897	0,747	
Std	0,040	0,032	0,041	0,025	0,041	0,040	0,041	0,037	0,044	0,041	

**Table 5.7:** Results of instance # 3 for all architectures

Instance #4											
Train images	9 6, 9 7, 7 5, 7 7, 8 5, 7 6										
	17x17 (3 blocks) CNN #1		17x17 (3 blocks) CNN #2		17x17 (3 blocks) CNN #3		17x17 (4 blocks) CNN #4		17x17 (3 blocks) CNN #5		
Test images	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	
9 5	0,919	0,767	0,920	0,770	0,912	0,760	0,929	0,804	0,919	0,778	
8 6	0,854	0,691	0,851	0,686	0,845	0,680	0,855	0,701	0,852	0,686	
8 7	0,901	0,734	0,891	0,710	0,882	0,701	0,896	0,735	0,870	0,717	
Mean	0,891	0,730	0,888	0,722	0,880	0,714	0,893	0,747	0,880	0,727	
Std	0,034	0,038	0,035	0,044	0,034	0,041	0,037	0,053	0,035	0,047	
	41x41 (4 blocks) CNN #6		49x49 (4 blocks) CNN #7		49x49 (5 blocks) CNN #8		57x57 (5 blocks) CNN #9		57x57 (6 blocks) CNN #10		
Test images	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	
9 5	0,931	0,809	0,924	0,793	0,928	0,797	0,930	0,806	0,930	0,805	
8 6	0,860	0,710	0,842	0,676	0,857	0,703	0,861	0,713	0,854	0,697	
8 7	0,902	0,746	0,891	0,725	0,899	0,739	0,905	0,756	0,904	0,752	
Mean	0,898	0,755	0,886	0,731	0,894	0,747	0,898	0,758	0,896	0,751	
Std	0,036	0,050	0,042	0,058	0,036	0,048	0,035	0,047	0,038	0,054	

**Table 5.8:** Results of instance # 4 for all architectures

Instance #5											
Train images	8 7, 8 5, 9 5, 7 5, 7 7, 8 6										
	17x17 (3 blocks) CNN #1		17x17 (3 blocks) CNN #2		17x17 (3 blocks) CNN #3		17x17 (4 blocks) CNN #4		17x17 (3 blocks) CNN #5		
Test images	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	
7 6	0,911	0,778	0,923	0,805	0,925	0,807	0,923	0,805	0,906	0,769	
9 6	0,817	0,630	0,823	0,636	0,836	0,660	0,829	0,649	0,815	0,624	
9 7	0,864	0,700	0,870	0,712	0,867	0,699	0,869	0,710	0,860	0,695	
Mean	0,864	0,703	0,872	0,718	0,876	0,722	0,873	0,721	0,860	0,696	
Std	0,047	0,074	0,050	0,084	0,045	0,076	0,047	0,079	0,045	0,073	
	41x41 (4 blocks) CNN #6		49x49 (4 blocks) CNN #7		49x49 (5 blocks) CNN #8		57x57 (5 blocks) CNN #9		57x57 (6 blocks) CNN #10		
Test images	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	
7 6	0,927	0,815	0,925	0,811	0,923	0,808	0,926	0,812	0,928	0,816	
9 6	0,838	0,669	0,827	0,645	0,829	0,652	0,836	0,664	0,831	0,653	
9 7	0,877	0,728	0,859	0,687	0,870	0,712	0,874	0,719	0,873	0,717	
Mean	0,881	0,737	0,871	0,714	0,874	0,724	0,879	0,732	0,877	0,728	
Std	0,045	0,073	0,050	0,086	0,047	0,079	0,045	0,075	0,049	0,082	

**Table 5.9:** Results of instance # 5 for all architectures

Instance #6											
Train images	7 5, 8 7, 9 5, 8 5, 9 6, 7 7										
	17x17 (3 blocks) CNN #1		17x17 (3 blocks) CNN #2		17x17 (3 blocks) CNN #3		17x17 (4 blocks) CNN #4		17x17 (3 blocks) CNN #5		
Test images	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	
8 6	0,854	0,693	0,872	0,734	0,873	0,735	0,877	0,743	0,865	0,720	
7 6	0,904	0,753	0,911	0,777	0,914	0,782	0,914	0,786	0,903	0,762	
9 7	0,875	0,718	0,875	0,724	0,875	0,719	0,876	0,725	0,865	0,704	
Mean	0,878	0,721	0,886	0,745	0,887	0,745	0,889	0,751	0,878	0,729	
Std	0,025	0,030	0,021	0,028	0,023	0,032	0,022	0,031	0,022	0,030	
	41x41 (4 blocks) CNN #6		49x49 (4 blocks) CNN #7		49x49 (5 blocks) CNN #8		57x57 (5 blocks) CNN #9		57x57 (6 blocks) CNN #10		
Test images	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	
8 6	0,880	0,750	0,876	0,743	0,876	0,740	0,878	0,745	0,879	0,747	
7 6	0,920	0,799	0,908	0,772	0,917	0,790	0,919	0,795	0,919	0,794	
9 7	0,880	0,734	0,863	0,699	0,871	0,711	0,874	0,716	0,878	0,726	
Mean	0,893	0,761	0,882	0,738	0,888	0,747	0,890	0,752	0,892	0,756	
Std	0,023	0,034	0,023	0,037	0,025	0,040	0,025	0,040	0,023	0,035	

**Table 5.10:** Results of instance # 6 for all architectures

Instance #7											
Train images	8 6, 8 7, 9 7, 8 5, 7 6, 7 5										
	17x17 (3 blocks) CNN #1		17x17 (3 blocks) CNN #2		17x17 (3 blocks) CNN #3		17x17 (4 blocks) CNN #4		17x17 (3 blocks) CNN #5		
Test images	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	
9 5	0,905	0,734	0,907	0,743	0,908	0,748	0,916	0,769	0,910	0,758	
7 7	0,935	0,751	0,944	0,784	0,937	0,761	0,945	0,792	0,929	0,741	
9 6	0,804	0,600	0,812	0,614	0,810	0,614	0,812	0,618	0,804	0,605	
Mean	0,881	0,695	0,888	0,714	0,885	0,707	0,891	0,726	0,881	0,701	
Std	0,068	0,083	0,068	0,089	0,067	0,081	0,070	0,095	0,067	0,084	
	41x41 (4 blocks) CNN #6		49x49 (4 blocks) CNN #7		49x49 (5 blocks) CNN #8		57x57 (5 blocks) CNN #9		57x57 (6 blocks) CNN #10		
Test images	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	
9 5	0,918	0,781	0,916	0,769	0,917	0,773	0,922	0,784	0,917	0,772	
7 7	0,944	0,788	0,944	0,790	0,936	0,759	0,943	0,778	0,936	0,760	
9 6	0,812	0,618	0,797	0,590	0,812	0,618	0,815	0,625	0,808	0,611	
Mean	0,891	0,729	0,886	0,716	0,888	0,717	0,893	0,729	0,887	0,714	
Std	0,070	0,096	0,078	0,110	0,067	0,086	0,068	0,090	0,069	0,090	

**Table 5.11:** Results of instance # 7 for all architectures

Instance #8											
Train images	9 7, 7 7, 8 5, 7 6, 7 5, 9 5										
	17x17 (3 blocks) CNN #1		17x17 (3 blocks) CNN #2		17x17 (3 blocks) CNN #3		17x17 (4 blocks) CNN #4		17x17 (3 blocks) CNN #5		
Test images	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	
8 6	0,847	0,682	0,857	0,702	0,841	0,673	0,856	0,703	0,844	0,679	
8 7	0,897	0,730	0,897	0,731	0,892	0,725	0,906	0,760	0,899	0,742	
9 6	0,825	0,642	0,824	0,637	0,820	0,632	0,819	0,632	0,815	0,623	
Mean	0,857	0,684	0,859	0,690	0,851	0,677	0,860	0,698	0,853	0,681	
Std	0,036	0,044	0,037	0,048	0,037	0,047	0,044	0,065	0,043	0,060	
	41x41 (4 blocks) CNN #6		49x49 (4 blocks) CNN #7		49x49 (5 blocks) CNN #8		57x57 (5 blocks) CNN #9		57x57 (6 blocks) CNN #10		
Test images	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	
8 6	0,853	0,695	0,845	0,682	0,859	0,705	0,861	0,712	0,866	0,720	
8 7	0,910	0,769	0,905	0,761	0,909	0,756	0,914	0,775	0,916	0,781	
9 6	0,827	0,646	0,817	0,628	0,827	0,642	0,829	0,650	0,826	0,643	
Mean	0,863	0,703	0,856	0,690	0,865	0,701	0,868	0,712	0,869	0,715	
Std	0,043	0,062	0,045	0,067	0,042	0,057	0,043	0,063	0,045	0,069	

**Table 5.12:** Results of instance # 8 for all architectures

Instance #9											
Train images	8 7, 9 5, 8 5, 7 5, 7 7, 9 7										
	17x17 (3 blocks) CNN #1		17x17 (3 blocks) CNN #2		17x17 (3 blocks) CNN #3		17x17 (4 blocks) CNN #4		17x17 (3 blocks) CNN #5		
Test images	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	
8 6	0,851	0,687	0,869	0,723	0,863	0,711	0,870	0,727	0,862	0,711	
9 6	0,825	0,637	0,835	0,658	0,838	0,665	0,831	0,652	0,826	0,643	
7 6	0,916	0,783	0,921	0,797	0,922	0,800	0,920	0,799	0,912	0,780	
Mean	0,864	0,702	0,875	0,726	0,874	0,725	0,874	0,726	0,867	0,711	
Std	0,047	0,074	0,043	0,070	0,043	0,069	0,045	0,074	0,043	0,069	
	41x41 (4 blocks) CNN #6		49x49 (4 blocks) CNN #7		49x49 (5 blocks) CNN #8		57x57 (5 blocks) CNN #9		57x57 (6 blocks) CNN #10		
Test images	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	
8 6	0,869	0,724	0,846	0,684	0,861	0,706	0,865	0,717	0,874	0,736	
9 6	0,834	0,658	0,831	0,656	0,830	0,650	0,831	0,653	0,835	0,661	
7 6	0,926	0,811	0,911	0,781	0,917	0,786	0,924	0,806	0,922	0,803	
Mean	0,876	0,731	0,863	0,707	0,869	0,714	0,873	0,726	0,877	0,733	
Std	0,047	0,077	0,043	0,065	0,044	0,068	0,047	0,077	0,044	0,071	

**Table 5.13:** Results of instance # 9 for all architectures

Instance #10											
Train images	7 5, 8 7, 9 6, 8 5, 8 6, 9 5										
	17x17 (3 blocks) CNN #1		17x17 (3 blocks) CNN #2		17x17 (3 blocks) CNN #3		17x17 (4 blocks) CNN #4		17x17 (3 blocks) CNN #5		
Test images	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	
9 7	0,868	0,709	0,876	0,723	0,869	0,712	0,872	0,719	0,866	0,705	
7 7	0,922	0,709	0,939	0,755	0,931	0,739	0,936	0,761	0,937	0,760	
7 6	0,906	0,766	0,906	0,765	0,910	0,777	0,909	0,775	0,910	0,778	
Mean	0,899	0,728	0,907	0,748	0,903	0,743	0,906	0,752	0,904	0,748	
Std	0,028	0,033	0,031	0,022	0,031	0,033	0,032	0,030	0,036	0,038	
	41x41 (4 blocks) CNN #6		49x49 (4 blocks) CNN #7		49x49 (5 blocks) CNN #8		57x57 (5 blocks) CNN #9		57x57 (6 blocks) CNN #10		
Test images	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	Acc	Kappa	
9 7	0,877	0,726	0,861	0,694	0,875	0,723	0,874	0,718	0,873	0,716	
7 7	0,947	0,790	0,937	0,764	0,937	0,758	0,944	0,781	0,942	0,776	
7 6	0,919	0,796	0,906	0,768	0,909	0,772	0,920	0,796	0,917	0,790	
Mean	0,915	0,771	0,901	0,742	0,907	0,751	0,913	0,765	0,911	0,761	
Std	0,035	0,039	0,038	0,041	0,031	0,025	0,036	0,041	0,035	0,039	

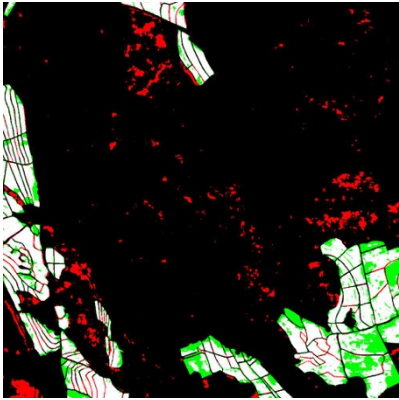
Table 5.14: Results of instance # 10 for all architectures

Scale	Overall acc. (%)	Kappa (k)
CNN #1 (17 × 17 – 3blocks)	87.57 ± 1.61	0.713 ± 0.026
CNN #2 (25 × 25 – 3blocks)	88.41 ± 1.33	0.725 ± 0.029
CNN #3 (33 × 33 – 3blocks)	88.02 ± 1.20	0.719 ± 0.021
CNN #4 (41 × 41 – 3blocks)	87.94 ± 1.21	0.720 ± 0.023
CNN #5 (33 × 33 – 4blocks)	88.66 ± 1.29	0.736 ± 0.025
CNN #6 (41 × 41 – 4blocks)	89.16 ± 1.26	0.746 ± 0.024
CNN #7 (49 × 49 – 4blocks)	88.08 ± 1.44	0.724 ± 0.028
CNN #8 (49 × 49 – 5blocks)	88.73 ± 1.19	0.734 ± 0.025
CNN #9 (57 × 57 – 5blocks)	89.10 ± 4.02	0.743 ± 0.022
CNN #10 (57 × 57 – 6blocks)	88.91 ± 4.13	0.740 ± 0.021
Combination architectures 3 blocks	89.00 ± 4.00	0.743 ± 0.018
Combination architectures 4 blocks	89.20 ± 2.00	0.749 ± 0.062
Combination architectures 5 blocks	89.20 ± 2.30	0.747 ± 0.061
Combination best architectures	89.60 ± 2.20	0.755 ± 0.062
Combination all architectures	89.70 ± 4.20	0.759 ± 0.062

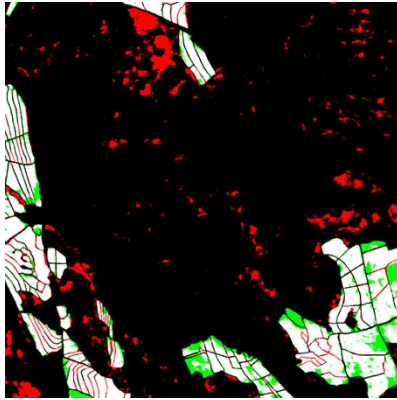
Table 5.15: Classification using CNNs over different scales and the combined results.

According to the results, one can observe that the combination of scales achieved better maps than the best individual scales. We can suppose that the combination improved the results by exploiting the diversity of individual CNNs in different scales. Overall, we could observe that the voting scheme fusion created an intermediate result among each scale, as expected.

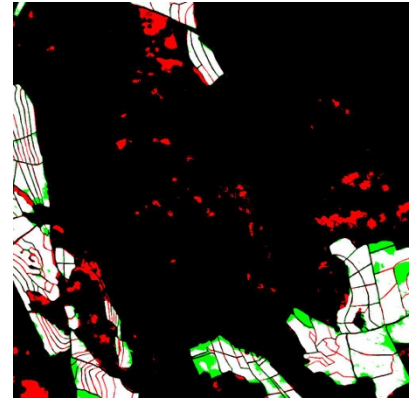
We showed an example of result for each single scale in Figure 5.6. Note, for example, the reduction of false positives (red) and false negatives (green) pixels as the scale increases. In contrast, smaller scales are better at recognizing non-coffee paths between coffee regions.



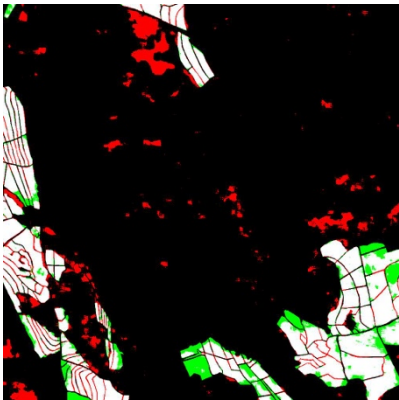
(a) CNN #1



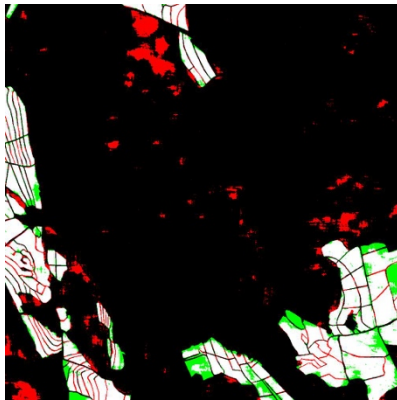
(b) CNN #2



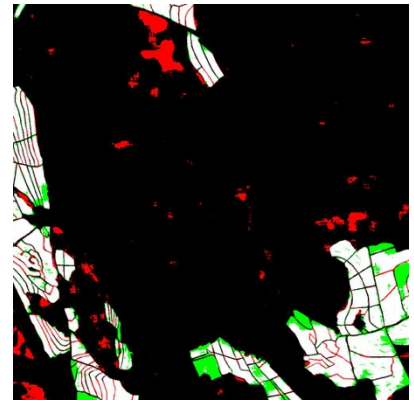
(c) CNN #3



(d) CNN #4

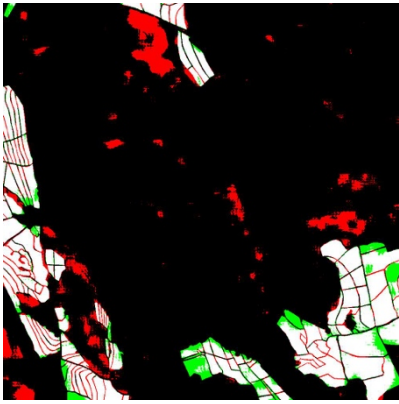


(e) CNN #5

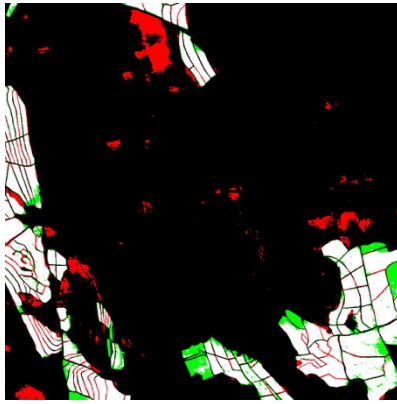


(f) CNN #6

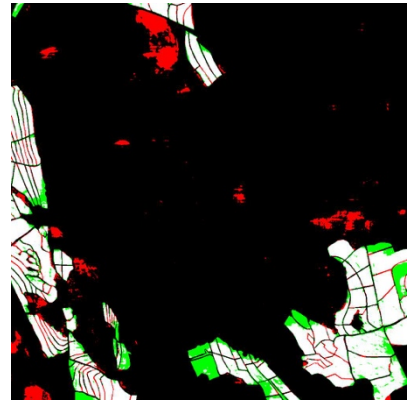




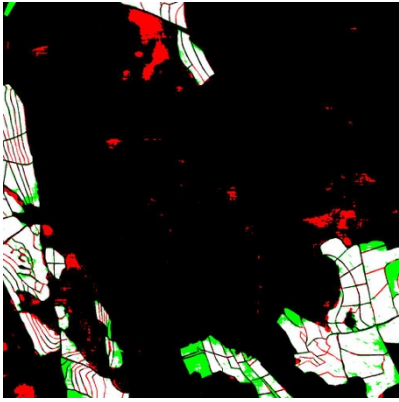
(g) CNN #7



(h) CNN #8



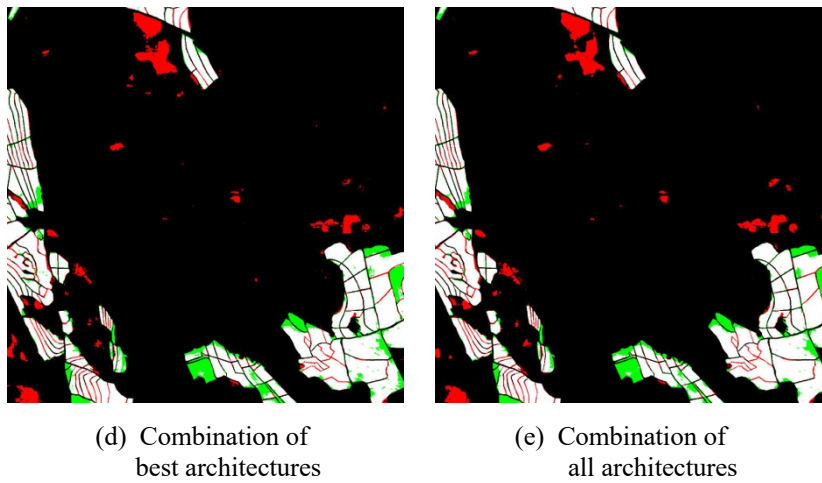
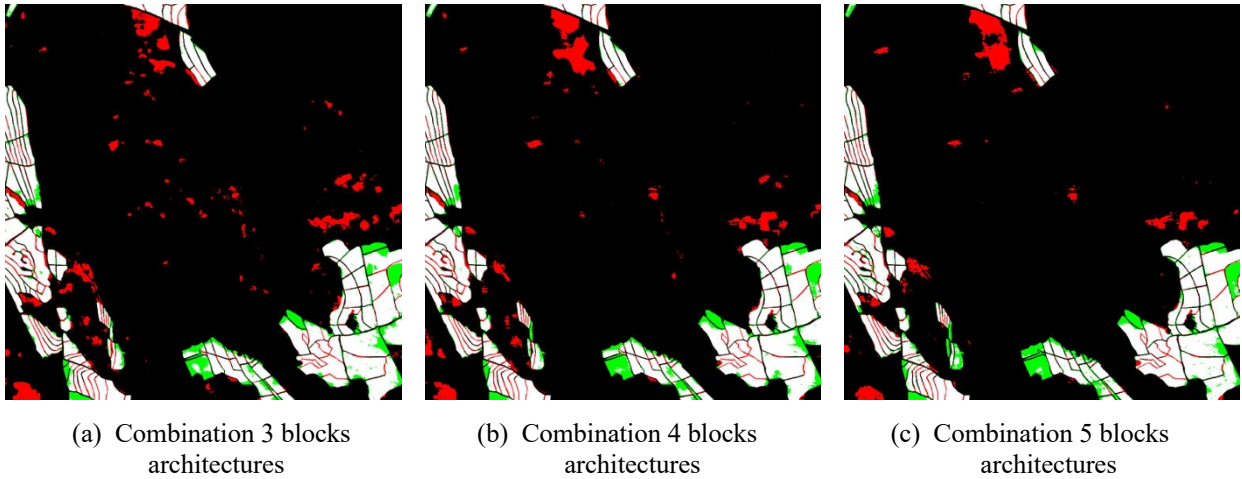
(i) CNN #9



(j) CNN #10

**Figure 5.6:** Results for each single scale. Pixels correctly classified are shown in white (true positive) and black (true negative) while misclassified pixels are displayed in red (false positive) and green (false negative).

In Figure 5.7, we show an example result for combining scales. As we can see, the combination of scales uses characteristics of each architecture. Note the reduction of false positives (red pixels) as we increase the number of combined scales. In addition, some non-coee paths between cultures have been maintained, which is a characteristic of small scales.



**Figure 5.7:** Results for combined scales. Pixels correctly classified are shown in white (true positive) and black (true negative) while misclassified pixels are displayed in red (false positive) and green (false negative).

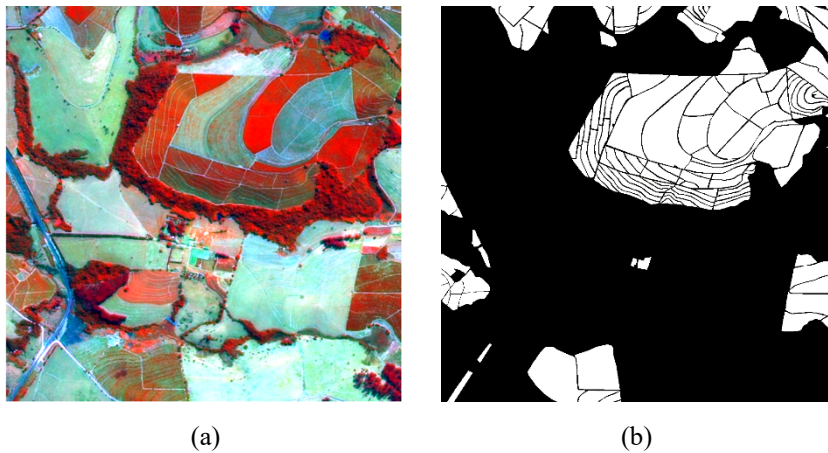
### 5.2.2 Comparison to the baselines

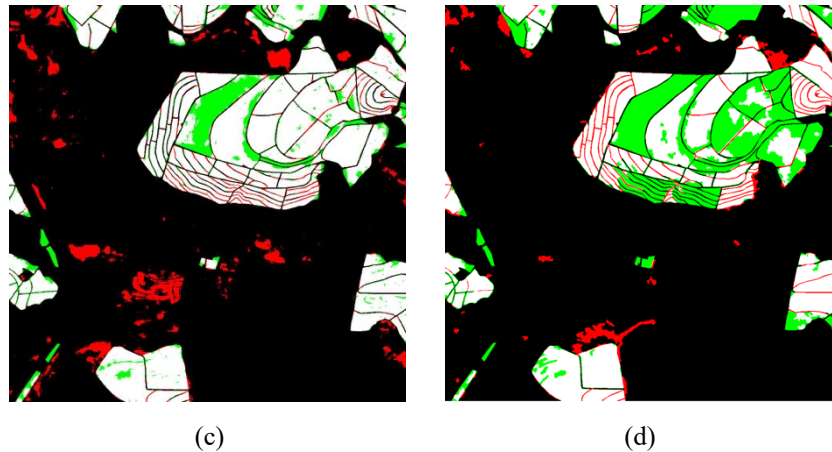
In Table 5.16 we presented the results for the proposed approach and the baselines.

<b>Approach</b>	<b>Overall Acc. (%)</b>	<b>Kappa (<math>\kappa</math>)</b>
SVM (RBF)	$80.09 \pm 1.58$	$0.748 \pm 0.025$
MSC-Boost	$82.28 \pm 1.60$	$0.780 \pm 0.025$
HMSC-Boost	$82.69 \pm 1.68$	$0.788 \pm 0.024$
Ours	$89.60 \pm 2.20$	$0.755 \pm 0.062$

**Table 5.16:** Classification results comparing the proposed approach against the baselines.

As it can be observed, the proposed approach overcame the results of the baseline only in one metric. Despite this, the proposed approach is more robust to recognize, correctly, the coffee class which is the class more important. Figure 5.8 illustrates an example of results comparing the proposed method against the HSMC-Boost baseline.





**Figure 5.8:** Example of results: (a) input image, (b) ground truth, (c) the proposed approach, and (d) HMSC-Boost. Pixels correctly classified are shown in white (true positive) and black (true negative) while the errors are displayed in red (false positive) and green (false negative).

One can observe that the main difference in these examples is that the proposed approach produced less false negative than the baseline. On the other hand, our approach produced more false positives

Overall, our approach seems to be promising in reducing two main problems found in the baselines: (1) to discriminate recently planted coffee crops; and (2) to detect paths between the crops. As pointed by dos Santos et al. [2012], most of the HMSC-Boost classification errors are related to confusion caused by recently planted coffee crops, which usually appear in light blue in the composition of colors displayed. The proposed approach achieved better results in those areas. Moreover, it was more effective in assigning the class “non-coffee” to the paths between crops, as can be also observed in Fig. 5.8. The more the number of “black lines” between coffee crops the more accurate was the classification of paths.

The regions in red in Fig. 5.8(c) indicates most of the false positives produced by the proposed approach are due to dense native vegetation canopy. We believe the misclassified pixels can be better classified by including largest context windows in the process. Also, these pixels are easier to remove by using some post-processing approaches than the misclassified regions produced by HSMC-Boost and other segmentation-based methods found in the literature. Note that the proposed approach misclassifies some very small groups or even isolated pixels.

## 6. Conclusions and future work

This work addressed the use of RSI to recognize coffee crops to build thematic maps and to answer the research questions related to the combination of scales, the choice of architectures and the effectiveness of the proposed approach in relation to baselines. For this purpose, we proposed the CNN-based approach that extracts, from a RSI, context windows of different scales at the same time. In Section 5.2, we evaluated the proposed approach using different architectures that are used individually and combined in a coffee scene, referred to in Section 5.1.1, which demonstrated a significant improvement using the Kappa index and overall accuracy metrics when we combine different scales and more robustness to recognize coffee crops compared to the baselines. Our approach extracted features described in Section 3.2 from the same region with different context windows and architectures that produced different characteristics that helps in the correct identification of the classes by the combination at the decision level. The creation of the final thematic maps consisted of classifying each non-labeled region by combining the class probability of each CNN into a resulting vector and selecting the highest probability as the predicted class.

According to the results, it was observed that the combination of scales brings better maps than the best individual scale. From this, we can suppose that the combination improves the results by exploring the diversity of individual CNNs at different scales, and the best result is achieved by combining all the architectures. Compared with baselines, the proposed approach is more robust to correctly recognize the class of coffee that is the most important class and appears to be promising in reducing two main problems encountered in baselines: (1) to discriminate recently planted coffee crops; and (2) detect paths between the crops.

As future work we plan to:

- analyze the influence of scale size on the recognition of each class separately for multiclass problems, which can help build more effective architectures that can work in a complementary way.
- use different and efficient fusion schemes. The majority voting, which was used as a fusion method in this work, takes into account only the final result of each architecture. In this way, other methods of fusion can be used to make better use of the information of each network, such as: combination of layers of characteristics, specialty of each architecture, among others.

- validate the proposed approach in another applications containing different data types.
- apply post-processing algorithms to help smoothing classification errors such as: Markov Random Field (MRF) [Rozanov, 1982] and Conditional Random Field (CRF) [Laerty et al., 2001].

## Bibliography

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., et al. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467.

Almeida, J., dos Santos, J. A., Alberton, B., Torres, R. d. S., and Morellato, L. P. C. (2014). Applying machine learning based on multiscale classifiers to detect remote phenology patterns in cerrado savanna trees. *Ecological Informatics*, 23:49--61.

Barsi, A. and Heipke, C. (2003). Artificial neural networks for the detection of road junctions in aerial images. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(3/W8):113--118.

Benediktsson, J., Chanussot, J., and Moon, W. (2013). Advances in very-high-resolution remote sensing [scanning the issue]. *Proceedings of the IEEE*, 101(3):566-569.

Berger, C., Voltersen, M., Hese, S., Walde, I., and Schmullius, C. (2013). Robust extraction of urban land cover information from hsr multi-spectral and lidar data. *Selected Topics in Applied Earth Observations and Remote Sensing, IEEE Journal of*, 6(5):2196--2211.

Bouchiha, R. and Besbes, K. (2013). Comparison of local descriptors for automatic remote sensing image registration. *Signal, Image and Video Processing*, 9(2):463--469.

Broomhead, D. S. and Lowe, D. (1988). Radial basis functions, multi-variable functional interpolation and adaptive networks. Technical report, DTIC Document.

Castelluccio, M., Poggi, G., Sansone, C., and Verdoliva, L. (2015). Land use classification in remote sensing images by convolutional neural networks. arXiv preprint arXiv:1508.00092

Chemura, A., Mutanga, O., and Dube, T. (2016). Separability of coee leaf rust infection levels with machine learning methods at sentinel-2 msi spectral resolutions. *Precision Agriculture*, pages 1--23.

Chen, Y., Lin, Z., Zhao, X., Wang, G., and Gu, Y. (2014). Deep learning-based classification of hyperspectral data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*.

Cheriyadat, A. M. (2014). Unsupervised feature learning for aerial scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, 52(1):439--451.

da Silveira, L. S., Valente, D. S. M., de Carvalho Pinto, F. d. A., and Santos, F. L.

(2017). Estudos de casos de classificação de áreas cultivadas com café por meio de descritores de textura. *Coe Science*, 11(4):502--511.

Dong, L. and Shan, J. (2013). A comprehensive review of earthquake-induced building damage detection with remote sensing techniques. *ISPRS Journal of Photogrammetry and Remote Sensing*, 84:85--99.

dos Santos, J., Penatti, O., Gosselin, P., Falcao, A., Philipp-Foliguet, S., and Torres, R. (2014). Efficient and effective hierarchical feature propagation. *Selected Topics in Applied Earth Observations and Remote Sensing*, IEEE Journal of, PP(99):1-12.

dos Santos, J. A., Gosselin, P., Philipp-Foliguet, S., da S. Torres, R., and Falcão, A. X. (2012). Multiscale classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 50:3764-3775.

Dos Santos, J. A., Gosselin, P.-H., Philipp-Foliguet, S., Torres, R. d. S., and Falcao, A. X. (2013). Interactive multiscale classification of high-resolution remote sensing images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 6(4):2020--2034.

dos Santos, J. A., Penatti, O. A. B., and da S. Torres, R. (2010). Evaluating the potential of texture and color descriptors for remote sensing image retrieval and classification. In *International Conference on Computer Vision Theory and Applications*, pages 203--208, Angers, France.

Faria, F., Pedronette, D., dos Santos, J., Rocha, A., and Torres, R. (2014). Rank aggregation for pattern classifier selection in remote sensing images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(4):1103-1115.

Faria, F. A., dos Santos, J. A., Torres, R. d. S., Rocha, A., and Falcao, A. X. (2012). Automatic fusion of region-based classifiers for coffee crop recognition. In *Geoscience and Remote Sensing Symposium (IGARSS), 2012 IEEE International*, pages 2221--2224. IEEE.

Ferreira, E., de Albuquerque Araújo, A., and dos Santos, J. A. (2016). A boosting-based approach for remote sensing multimodal image classification. In *Graphics, Patterns and Images (SIBGRAPI), 2016 29th SIBGRAPI Conference on*, pages 416--423. IEEE.

Firat, O., Can, G., and Yarman Vural, F. (2014). Representation learning for contextual object and region detection in remote sensing. In *International Conference on Pattern Recognition*, pages 3708-3713.

Glorot, X. and Bengio, Y. (2010). Understanding the difficulty of training deep feed-



- forward neural networks. In Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, pages 249--256.
- Guigues, L. and Le Men, H. (2003). Scale-sets image analysis. In Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on, volume 2, pages II--45. IEEE.
- Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., and Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18--28.
- Hung, C., Xu, Z., and Sukkarieh, S. (2014). Feature learning based approach for weed classification using high resolution aerial images from a digital camera mounted on a uav. *Remote Sensing*, 6(12):12037--12054.
- Ippoliti-Ramilo, G. A., Epiphonio, J. C. N., Shimabukuro, Y. E., and Formaggio, A. R. (1999). Sensoriamento remoto orbital como meio auxiliar na previsão de safras. *Agricultura em São Paulo*, 46:89--101.
- Jollie, I. (2002). Principal component analysis. Wiley Online Library.
- Laerty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data.
- Landis, J. R. and Koch, G. G. (1977). The measurement of observer agreement for categorical data. *Biometrics*, 33(1):159--174.
- LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278--2324.
- Lotufo, R. and Falcao, A. (2002). The ordered queue and the optimality of the watershed approaches. *Mathematical Morphology and its Applications to Image and Signal Processing*, pages 341--350.
- Luz, E. and Menotti, D. (2015). Denoising autoencoder for iris recognition in noncooperative environments. In *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pages 200--207. Springer.
- Makantasis, K., Karantzas, K., Doulamis, A., and Doulamis, N. (2015). Deep supervised learning for hyperspectral data classification through convolutional neural networks. In *Geoscience and Remote Sensing Symposium (IGARSS), 2015 IEEE International*, pages 4959--4962. IEEE.
- Meneses, P. R. and Almeida, T. d. (2012). *Introdução ao processamento de imagens de sensoriamento remoto*. Embrapa Cerrados-Livros técnicos (INFOTECA-E).
- Menotti, D., Chiachia, G., Falcao, A. X., and Oliveira Neto, V. (2014). Vehicle license

plate recognition with random convolutional networks. In Graphics, Patterns and Images (SIBGRAPI), 2014 27th SIBGRAPI Conference on, pages 298--303. IEEE.

Midhun, M. E., Nair, S. R., Prabhakar, V. T. N., and Kumar, S. S. (2014). Deep model for classification of hyperspectral image using restricted boltzmann machine. In International Conference on Interdisciplinary Advances in Applied Computing, pages 35:1--35:7.

Ministério da agricultura, p. e. a. (2017). Café no brasil. <http://www.agricultura.gov.br/assuntos/politica-agricola/cafe/cafeicultura-brasileira>. Accessed: 2017-07-20.

Nogueira, K., Dalla Mura, M., Chanussot, J., Schwartz, W. R., and dos Santos, J. A. (2016). Learning to semantically segment high-resolution remote sensing images. In Pattern Recognition (ICPR), 2016 23rd International Conference on, pages 3566--3571. IEEE.

Nogueira, K., Penatti, O. A., and dos Santos, J. A. (2017). Towards better exploiting convolutional neural networks for remote sensing scene classification. *Pattern Recognition*, 61:539--556.

Nogueira, K., Schwartz, W. R., and dos Santos, J. A. (2015). Coee crop recognition using multi-scale convolutional neural networks. In Iberoamerican Congress on Pattern Recognition, pages 67--74. Springer.

Penatti, O. A. B., Nogueira, K., and dos Santos, J. A. (2015). Do deep features generalize from everyday objects to remote sensing and aerial scenes domains? In Computer Vision and Pattern Recognition Workshops (CVPRW), 2015 IEEE Conference on, pages 44--51.

Romero, A., Gatta, C., and Camps-Valls, G. (2014). Unsupervised feature extraction of hyperspectral images. In International Conference on Pattern Recognition.

Rozanov, Y. A. (1982). Markov random elds. In *Markov Random Fields*, pages 55--102. Springer.

Ruder, S. (2016). An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747.

Santos, J., Faria, F., Calumby, R., Torres, R. d. S., and Lamparelli, R. A. (2010). A genetic programming approach for coee crop recognition. In Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International, pages 3418--3421. IEEE.

Schapire, R. E. (1999). A brief introduction to boosting. In *Ijcai*, volume 99, pages 1401--1406.

Silva, P., Luz, E., Baeta, R., Menotti, D., Pedrini, H., and Falcao, A. X. (2015). An

- approach to iris contact lens detection based on deep image representations. In Graphics, Patterns and Images (SIBGRAPI), 2015 28th SIBGRAPI Conference on, pages 157--164. IEEE.
- Slavkovikj, V., Verstockt, S., De Neve, W., Van Hoecke, S., and Van de Walle, R. (2015). Hyperspectral image classification with convolutional neural networks. In Proceedings of the 23rd ACM international conference on Multimedia, pages 1159--1162. ACM.
- Souza, C. G., Carvalho, L., Aguiar, P., and Arantes, T. B. (2016). Algoritmos de aprendizagem de máquina e variáveis de sensoriamento remoto para o mapeamento da cafeicultura. *Boletim de Ciências Geodésicas*, 22(4):751--773.
- Tokarczyk, P., Wegner, J., Walk, S., and Schindler, K. (2015). Features, color spaces, and boosting: New insights on semantic classification of remote sensing images. *IEEE Transactions on Geoscience and Remote Sensing*, 53(1):280-295.
- Tsai, D.-M. and Chen, W.-L. (2017). Coffee plantation area recognition in satellite images using fourier transform. *Computers and Electronics in Agriculture*, 135:115-127.
- Tuia, D., Flamary, R., and Courty, N. (2015). Multiclass feature learning for hyperspectral image classification: Sparse and hierarchical solutions. *{ISPRS} Journal of Photogrammetry and Remote Sensing*, (0):.
- Wang, L., Zhang, J., Liu, P., Choo, K.-K. R., and Huang, F. (2017). Spectralspatial multi-feature-based deep learning for hyperspectral remote sensing image classification. *Soft Computing*, 21(1):213--221.
- Xie, H., Wang, S., Liu, K., Lin, S., and Hou, B. (2014). Multilayer feature learning for polarimetric synthetic radar data classification. In *IEEE International Geoscience & Remote Sensing Symposium*, pages 2818-2821.
- Yang, Y. and Newsam, S. (2008). Comparing sift descriptors and gabor texture features for classification of remote sensed imagery. In *International Conference on Image Processing*, pages 1852--1855.
- Yao, X., Han, J., Cheng, G., Qian, X., and Guo, L. (2016). Semantic annotation of high-resolution satellite images via weakly supervised learning.
- Yokoya, N., Nakazawa, S., Matsuki, T., and Iwasaki, A. (2014). Fusion of hyperspectral and lidar data for landscape visual quality assessment. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 7(6):2419--2425.
- Zhang, F., Du, B., and Zhang, L. (2015). Saliency-guided unsupervised feature learn-

ing for scene classification. IEEE Transactions on Geoscience and Remote Sensing, 53(4):2175-2184.

Zhou, W., Huang, G., Troy, A., and Cadenasso, M. (2009). Object-based land cover classification of shaded areas in high spatial resolution imagery of urban areas: A comparison study. Remote Sensing of Environment, 113(8):1769--1777.

Zhou, Y. and Wei, Y. (2016). Learning hierarchical spectralspatial features for hyperspectral image classification. IEEE transactions on cybernetics, 46(7):1667--1678.