UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

CÉSAR SALGADO VIEIRA DE SOUZA

Classify-Normalize-Classify: A Novel Data-driven Framework for Classifying Forest Pixels in Remote Sensing Images

Thesis presented in partial fulfillment
of the requirements for the degree of
Master of Computer Science

Advisor: Prof. Dr. Jacob Scharcanski

Porto Alegre
2017

CIP — CATALOGING-IN-PUBLICATION

Souza, César Salgado Vieira de

Classify-Normalize-Classify: A Novel Data-driven Framework for Classifying Forest Pixels in Remote Sensing Images / César Salgado Vieira de Souza. – Porto Alegre: PPGC da UFRGS,

.

77 f.: il.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS,

. Advisor: Jacob Scharcanski.

1. Image normalization. 2. Radiometric correction. 3. Pixel classification. 4. Forest Segmentation. 5. Deforestation detection. 6. Supervised learning. I. Scharcanski, Jacob. II. Título.

“More things should not be used than are necessary.”

— WILLIAM OF OCKHAM

AGRADECIMENTOS

Primeiramente, agradeço meu orientador, Professor Dr. Jacob Scharcanski, pela paciência, orientações e liberdade que me deu durante minha pesquisa.

Ao meu amigo Maciel Zortea que me ajudou muito durante a pesquisa e informalmente teve um papel de co-orientador.

Aos meus pais, especialmente minha mãe, pela educação, valores e apoio que foram fundamentais na minha vida.

À minha noiva, Daniela, que sempre me deu amor, carinho e manteve minha alegria mesmo nos dias mais difíceis.

Aos meus irmãos, avós, tios e primos que influenciaram minha vida de uma forma muito positiva.

Ao meu sogro e minha sogra que me receberam como um filho em sua casa.

Aos colegas do laboratório onde todos eram muito legais, inteligentes e sempre prestativos quando eu precisava de ajuda. Obrigado por todas as conversas e momentos de descontração que tivemos.

Agradeço também ao PPGC e a UFRGS por proporcionar um mestrado de alta qualidade.

ABSTRACT

Monitoring natural environments and their changes over time requires the analysis of a large amount of image data, often collected by orbital remote sensing platforms. However, variations in the observed signals due to changing atmospheric conditions often result in a data distribution shift for different dates and locations making it difficult to discriminate between various classes in a dataset built from several images. This work introduces a novel supervised classification framework, called Classify-Normalize-Classify (CNC), to alleviate this data shift issue. The proposed scheme uses a two classifier approach. The first classifier is trained on non-normalized top-of-the-atmosphere reflectance samples to discriminate between pixels belonging to a class of interest (COI) and pixels from other categories (e.g. forest vs. non-forest). At test time, the estimated COI's multivariate median signal, derived from the first classifier segmentation, is subtracted from the image and thus anchoring the data distribution from different images to the same reference. Then, a second classifier, pre-trained to minimize the classification error on COI median centered samples, is applied to the median-normalized test image to produce the final binary segmentation.

The proposed methodology was tested to detect deforestation using bitemporal Landsat 8 OLI images over the Amazon rainforest. Experiments using top-of-the-atmosphere multispectral reflectance images showed that the deforestation was mapped by the CNC framework more accurately as compared to running a single classifier on surface reflectance images provided by the United States Geological Survey (USGS). Accuracies from the proposed framework also compared favorably with the benchmark masks of the PRODES program.

Keywords: Image normalization. Radiometric correction. Pixel classification. Forest Segmentation. Deforestation detection. Supervised learning.

Classifica-Normaliza-Classifica: Uma Nova Abordagem para Classificar Pixels de Floresta em Imagens de Sensoriamento Remoto

RESUMO

O monitoramento do meio ambiente e suas mudanças requer a análise de uma grade quantidade de imagens muitas vezes coletadas por satélites. No entanto, variações nos sinais devido a mudanças nas condições atmosféricas frequentemente resultam num deslocamento da distribuição dos dados para diferentes locais e datas. Isso torna difícil a distinção dentre as várias classes de uma base de dados construída a partir de várias imagens. Neste trabalho introduzimos uma nova abordagem de classificação supervisionada, chamada Classifica-Normaliza-Classifica (CNC), para amenizar o problema de deslocamento dos dados. A proposta é implementada usando dois classificadores. O primeiro é treinado em imagens não normalizadas de refletância de topo de atmosfera para distinguir dentre pixels de uma classe de interesse (CDI) e pixels de outras categorias (e.g. floresta versus não-floresta). Dada uma nova imagem de teste, o primeiro classificador gera uma segmentação das regiões da CDI e então um vetor mediano é calculado para os valores espectrais dessas áreas. Então, esse vetor é subtraído de cada pixel da imagem e portanto fixa a distribuição de dados de diferentes imagens num mesmo referencial. Finalmente, o segundo classificador, que é treinado para minimizar o erro de classificação em imagens já centralizadas pela mediana, é aplicado na imagem de teste normalizada no segundo passo para produzir a segmentação binária final.

A metodologia proposta foi testada para detectar desflorestamento em pares de imagens co-registradas da Landsat 8 OLI sobre a floresta Amazônica. Experimentos usando imagens multiespectrais de refletância de topo de atmosfera mostraram que a CNC obteve maior acurácia na detecção de desflorestamento do que classificadores aplicados em imagens de refletância de superfície fornecidas pelo *United States Geological Survey*. As acurácias do método proposto também se mostraram superiores às obtidas pelas máscaras de desflorestamento do programa PRODES.

Palavras-chave: Normalização de imagens, Correção radiométrica, Classificação de pixels, Segmentação de floresta, Detecção de desflorestamento, Aprendizado supervisionado.

LIST OF ABBREVIATIONS AND ACRONYMS

COI	Class of Interest
PIFs	Pseudo-Invariant Features
CNC	Classify-Normalize-Classify
GLCM	Gray-Level Co-occurrence Matrix
MAD	Multivariate Alteration Detection
MRF	Markov Random Field
PCC	Post-Classification Comparison
RBF	Radial Basis Function
SGD	Stochastic Gradient Descent
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
LDA	Linear Discriminant Analysis
KNN	K-Nearest Neighbors
MLP	Multi-layer Perceptron
QDA	Quadratic Discriminant Analysis
RF	Random Forest
SVM	Support Vector Machine
LaSRC	Landsat Surface Reflectance Code (atmospheric correction algorithm for the Landsat 8 OLI)
MODIS	Moderate-Resolution Imaging Spectroradiometer
OLI	Operational Land Imager
PRODES	Project for Monitoring Deforestation in the Legal Amazon by Satellite
SR	Surface Reflectance
TOA	Top-of-Atmosphere
USGS	United States Geological Survey

LIST OF SYMBOLS

f_1	CNC first classifier
f_2	CNC second classifier
f_{cnc}	Function representing the whole CNC pipeline.
$\mathbb{X}_d$	Unnormalized dataset of pixel-label pairs of domain d
$\mathbb{X}$	The union of all domains unnormalized datasets
$\boldsymbol{x}_d^{(i)}$	Spectral vector for the ith pixel of domain d
$y_d^{(i)}$	Ground-truth label for the ith pixel of domain d
P_d^1	Multivariate distribution for class with label 1 of domain d
P_d^0	Multivariate distribution for class with label 0 of domain d
P_d	Multivariate distribution of domain d of a sample belonging to any class
$\boldsymbol{m}_d^1$	Multivariate median of the distribution P_d^1
$\mathbb{X}_d^*$	Normalized dataset containing median centered pixels for domain d
$\mathbb{X}^*$	The union of all domains normalized datasets
$\hat{y}_d^{(i)}$	Predicted label for the ith unnormalized pixel of domain d
$\hat{y}_d^{*(i)}$	Predicted label for the ith normalized pixel of domain d
$\mathbf{T}_u^i$	Multidimensional array representing the unnormalized tile i
$\mathbf{T}_n^i$	Multidimensional array representing the normalized tile i
$\boldsymbol{F}^i$	Predicted forest mask for tile i with value 1 in the forest pixels and 0 in the rest
$\hat{\boldsymbol{m}}_f$	Estimated forest median vector
$\hat{\boldsymbol{D}}$	Predicted deforestation mask with value 1 in the loss regions and 0 in the rest

LIST OF FIGURES

LIST OF TABLES

CONTENTS

1 INTRODUCTION

Designing robust classification methods when the underlying statistics of the classes of interest vary in space and time is a challenging task. In remote sensing applications, this issue may arise, for instance, due to local variations in atmospheric conditions and seasonal changes in illumination. Often, these shifts negatively affect classification accuracies, especially when the user wants to learn a model in a specific training set and apply it to new scenes acquired at different locations and dates. We propose to handle this issue with the proposed Classify-Normalize-Classify (CNC) framework.

We aim to create a framework that alleviates the data distribution shift between images within the training and test set, and also across them. As a working hypothesis, we assume that by compensating for a spectral shift within the training and test images, the data will become more easily separable, and the learned model will generalize better to the test set.

A partial solution to the above-mentioned data shift problem could be preprocessing the input images using atmospheric correction algorithms such as the MODTRAN (BERK; BERNSTEIN; ROBERTSON, 1987; TARDY et al., 2016) or the 6S codes (VERMOTE et al., 1997) that in principle can remove undesired effects in the observed radiance signals captured by orbital sensors in the visible and infrared electromagnetic spectrum, therefore making a pretrained classification model applicable to other atmospherically corrected images. However, detailed information about the atmosphere conditions is needed as input to those algorithms at each pixel location, but often they are unavailable or are not known precisely (SONG et al., 2001). Another possible solution is the dark object subtraction approach (JR, 1988), which may be less demanding regarding the input parameters. However, this method is not as accurate as more complex methods such as the 6S codes (NAZEER; NICHOL; YUNG, 2014).

Relative normalization of overlapping images based on the selection of fixed landmark points in multitemporal acquisitions is an alternative to aligning spectral observations to a common reference, facilitating classification and change detection in remote sensing (SCHOTT; SALVAGGIO; VOLCHOK, 1988; BAO et al., 2012; CANTY; NIELSEN, 2008). For instance, the multivariate alteration detection (MAD) transformation (CANTY; NIELSEN, 2008) can automatically find normalization targets, also known as pseudo-invariant features (PIFs), but these can only be applied to a set of images that intersect geographically. A significant challenge remains when the labeled training set

and unlabeled test images do not overlap spatially.

Recently, transfer learning methods (PAN; YANG, 2010) are emerging as an alternative to tackle the problem of training a supervised model in a given data and testing it in other data cubes from different acquisitions. Manifold alignment (HAM; LEE; SAUL, 2005) is a promising strategy currently being investigated in the remote sensing literature. Despite promising at dealing with the shifting classes distribution problem, often these methods need at least a few labeled samples in the target domain, thus making them not fully automatic at test time (TUIA et al., 2014; TUIA; CAMPS-VALLS, 2016). There are also techniques which do not need labeled examples in the target domain, but they do not accept multiple sources domains to be simultaneously aligned (TUIA et al., 2013; MATASCI et al., 2015).

Differently from current relative normalization methods, here we propose the CNC framework which can handle multiple domains and for each one it translates the median vector of a class of interest (COI) to the origin and therefore it is independent of a reference image. It is also fully automatic at test time and allows several possibly non-overlapping images to be put in a common spectral reference using a data-driven approach. The CNC requires labeled samples only at the initial training phase.

The CNC has three steps. The first one consists in applying to the input image a binary classifier f_1 trained on unnormalized images to discriminate between a class of interest (COI) and all the remaining pixels of the image (non-COI). This produces an initial rough COI segmentation. Then, the multivariate median spectral signal of the COI segmented region is computed and subtracted from all the input image pixels values. Now that the input image is normalized (shifted), a second classifier f_2, previously trained on COI median centered images, is applied to the input pixels to produce the final refined segmentation. The median is used instead of the mean because the former is more robust to outliers than the later (see Section 4.1 for more information).

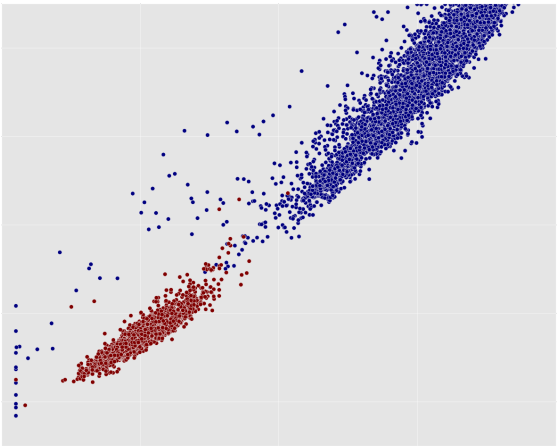
The first two steps (Classify-Normalize) of the CNC framework can be seen as an approximate relative radiometric normalization. It is relative because the values do not correspond to absolute reflectance values, and is an approximation because the proposed model only accounts for a possible rigid translation and does not take into consideration an eventual deformation of the classes distributions.

We evaluate the framework in the context of automatic pixel-wise deforestation detection as this task can be reduced to classifying forest/non-forest pixels in two co-registered images, and then finding the difference between their resulting binary masks.

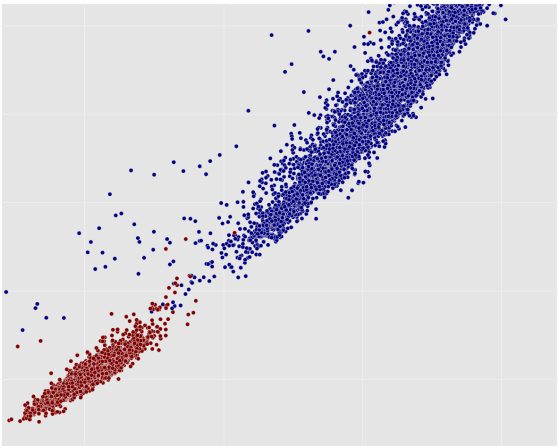
This technique is known in the literature as post-classification comparison (SINGH, 1989). We focus on Landsat 8 OLI images (ROY et al., 2014) acquired over the Amazon rainforest. As our experiments will show, segmenting an unnormalized image, with a single classifier, into forest/non-forest categories is not as effective as when classifying with the complete CNC pipeline. Although a single classifier may only provide a rough forest segmentation, it is sufficient to estimate the forest median multispectral signal accurately. Figure 1.1 illustrates the problem of shifting distributions in the context of forest/non-forest classification. As can be seen from the figure, the plots regarding unnormalized pixels have a large shift of the classes clusters from one image to the other, and this causes points from different categories to mix together when combining the points from different images. On the other hand, the normalized pixels points in the right-column plots in Figure 1.1 lead to a more modest distribution shift from one image to the other, and therefore the combined plots in the last row can be more easily divided between forest and non-forest categories.

It is important to note that the aim of this work is not to present a complete chain for deforestation detection as in Hansen et al. (2013), Hansen et al. (2016). Instead, we evaluate in which circumstances the CNC framework performs well and how it compares to other methods in similar settings. This means that, for instance, we do not include an automatic method for cloud screening, and we use pairs of images for detecting changes through the post-classification comparison strategy, instead of classifying the stacked images directly, using object-based classification (BLASCHKE, 2010; WANG; JENSEN; IM, 2010) or using a time-series approach. The proposed scheme can be combined with other methods (e.g. in ensembles or with additional preprocessing procedures) to produce a final change detection product.

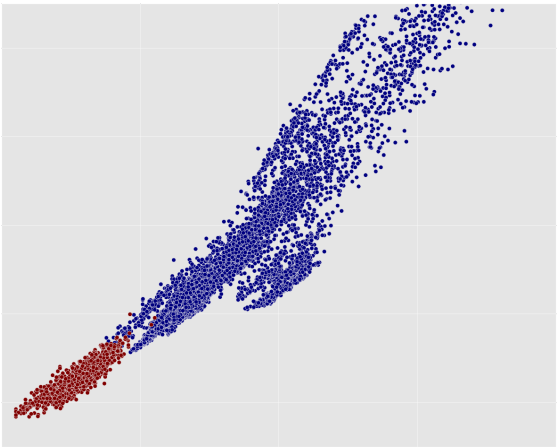
Figure 1.1: The left column sub-figures places the forest (in red) and non-forest (in blue) pixels of two real Landsat 8 OLI images in the spectral space, where the horizontal axis represents the top-of-atmosphere reflectance for the band SWIR 1 and the vertical axis the reflectance in band SWIR 2. In the right column the pixels of each image were normalized to have its forest median translated to the origin. The plots in the first row regard image A, the plots in the second row come from image B, and the bottom row show plots that are the fusion of the two plots above it.



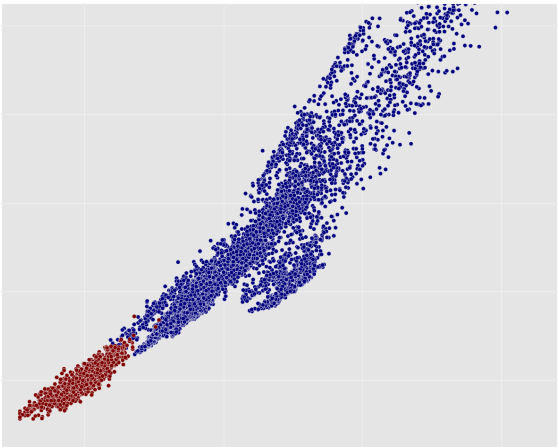
(a) Plot of **unnormalized** pixels of image A.



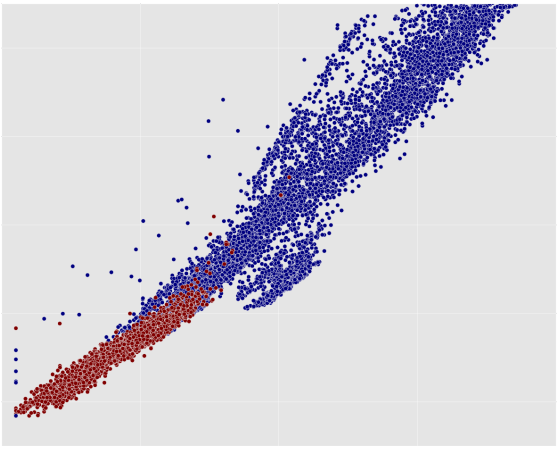
(b) Plot of **normalized** pixels of image A.



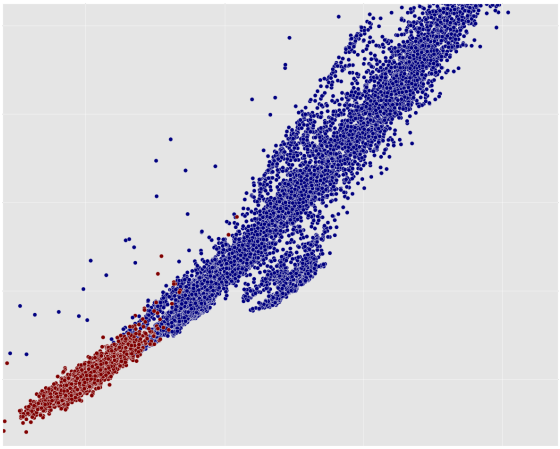
(c) Plot of **unnormalized** pixels of image B.



(d) Plot of **normalized** pixels of image B.



(e) Plot of **unnormalized** coming from both images.



(f) Plot of **normalized** coming from both images.

1.1 Goals of this Thesis

The main goals of this work are:

- To develop a fully automatic deforestation detection method that can generalize well to new images not seen in the training stage and possibly acquired at different atmospheric conditions.
- To describe the proposed CNC framework and evaluate which of its various possible configurations works well in detecting tropical deforestation using Landsat 8 OLI images.
- To compare the CNC framework with other competitive methods on the task of deforestation detection.

1.2 Comparative Review of the State of the Art

To evaluate the usefulness of the proposed CNC framework, we will compare it with other techniques on the task of deforestation detection. As the CNC does not specify which classifiers it should use in the first and third step, in Section 5.3.1 we will search for a suitable combination of classifiers f_1 and f_2 for the deforestation detection task and select three CNC configurations to be compared with the other methods. The following classifiers were tried as candidates for f_1 and f_2: CNN, LDA, QDA, KNN, MLP, Random Forest, and SVM. Each of those methods can receive as an input just the pixel being classified or also a context window centered on it. For the contextual methods we optionally also extract GLCM features (HARALICK; SHANMUGAM; DINSTEIN, 1973).

We will then compare the three selected CNC instances applied to TOA reflectance images against classifiers applied to images transformed to surface reflectance with the Landsat 8 OLI atmospheric correction algorithm (LaSRC) (VERMOTE et al., 2016). This is a natural comparison as one of the main goals of the CNC is to work well in new images with possibly different atmospheric conditions and the surface reflectance images provided by USGS is currently the standard option for trying to achieve this goal while using Landsat 8 images. Also, the first two steps of the CNC (Classify-Normalize) can be considered an approximate relative radiometric normalization as mentioned earlier. So indirectly, we will be comparing the normalization of the first two phases of CNC against

the correction performed by the LaSRC algorithm. All classifiers, context sizes, and features that were tried as f_1 and f_2 of the CNC will also be applied to surface reflectance images so that the chosen CNC instances can be compared with the classifiers that obtained best results on images corrected with the LaSRC [1]. Also, as a weaker baseline, we will compare the best CNC instances against classifiers applied to non-atmospheric-corrected TOA reflectance images.

We will also compare the deforestation maps generated by the CNC instances with those of the PRODES program (SHIMABUKURO et al., 2012). PRODES is a project from the Brazilian government for monitoring deforestation in the Amazon forest. Every year they produce deforestation maps using a semi-automatic method that involves automatic segmentation and manual annotations by a human photo interpreter.

1.3 Contributions of this Thesis

The main contributions of this thesis are:

- The development of the novel supervised classification framework CNC which is fully automatic at test time and generalize well for unseen images acquired at different atmospheric conditions.
- Evaluation of what configurations of the CNC framework works well in the deforestation detection task.
- Assessment of how well standard classifiers performs on the deforestation detection task when directly applied to surface or TOA reflectance images.
- Creation of a new forest classification and deforestation detection dataset with ground truth annotations.
- Determination of the PRODES masks accuracies for three investigated regions by using the exhaustive deforestation ground truth annotations created by the author.

[1] In this thesis, when we say "surface reflectance images" we always mean images that were corrected with the Landsat 8 atmospheric correction algorithm (LaSRC).

1.4 Summary of the Main Conclusions

We anticipate here the main findings and insights obtained after performing the experiments:

- Non-contextual classifiers perform better than the contextual classifiers on the deforestation detection task.
- The best CNC instances applied to non-atmospheric-corrected images maps deforestation with higher accuracy than the best classifiers applied to images atmospherically corrected with the LaSRC algorithm.
- The deforestation maps generated by the best CNC instances are more accurate than the ones produced by the PRODES program.
- Standard classifiers applied to surface reflectance images perform better than standard classifiers applied to TOA reflectance images.

1.5 Thesis Structure

The remaining of this work is organized as follows. Chapter 2 review some fundamental concepts in supervised learning and remote sensing that will be useful in later chapters. Chapter 3 review relevant literature related to deforestation detection, and image normalization/correction. Chapter 4 describes the proposed CNC framework in detail. The dataset, experimental setup, and results are presented in Chapter 5. Chapter 6 presents a general discussion, future work, and conclusion. In Appendix A, we show more detailed evaluation metrics for every method, input normalization, and tested region. In Appendix B, preliminary experiments using alternative normalization options are shown.

2 FUNDAMENTALS REVIEW

In this chapter, we will review some fundamental concepts in supervised learning and remote sensing that will be useful in later chapters.

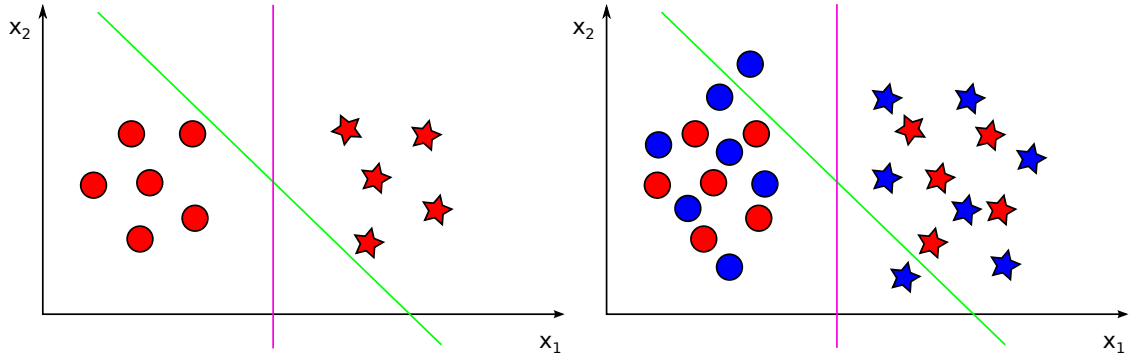
2.1 Supervised Learning

Supervised Learning is a subfield of Machine Learning where we are given a training set of example-label pairs $\{\boldsymbol{x}^{(i)}, y^{(i)} \mid i = 1 \dots n\}$ and then we try to learn a function f that aims to map the examples $\boldsymbol{x}^{(i)}$ to their correct labels $y^{(i)}$ even for new examples that are not present in the training set. If the labels $y^{(i)}$ are continuous values, we call that a regression problem, and if the labels represent discrete classes it is known as a classification problem, and in this context, f is called a classifier. In this work, we will only be concerned with the later, although in the future work section we mention how the framework proposed in this thesis could be extended to handle regression problems.

The example $\boldsymbol{x}^{(i)}$ is represented by a vector of real values that correspond to features of the object being classified. Sometimes it may be useful to transform the vector $\boldsymbol{x}^{(i)}$ to a new feature space where the transformed vectors $\boldsymbol{z}^{(i)}$ of examples from different classes may be more easily distinguishable than they would be in its original representation. These transformations sometimes can be hand-engineered by researchers, as is the case for the GLCM descriptors (HARALICK; SHANMUGAM; DINSTEIN, 1973), and sometimes they can be learned automatically from the data itself, as it happens inside the neural networks based methods. To better understand what a classifier is trying to accomplish Figure 2.1a illustrates a set of training examples where the vectors $\boldsymbol{x}^{(i)}$ have two dimensions. We also show in the figure a line representing the decision boundary found by two hypothetical linear classifiers. Classifier *A* finds a tilted line as a solution and classifier *B* finds a vertical line. Figure 2.1b, in addition to the training points, also shows the test points. While both classifiers solutions make zero mistakes in the training set, the classifier *A* makes two mistakes in the test set, while classifier *B* still makes no error. In this case, we say the classifier *B* generalized better than the classifier *A*.

In the following subsections, we will describe briefly each classifier that will be used in this thesis. As in this thesis we just approach binary classification problems, we will describe all classifiers supposing there are just to classes of interest.

Figure 2.1: Supervised classification illustration: (a) Training set of 2D samples for two classes (circles and stars). The tilted line represents the decision boundary of an hypothetical linear classifier *A* and the vertical line correspond to another classifier *B*; (b) The training samples (in red) and test samples (in blue) are combined in the same image.



(a) Training samples (b) Training and test samples

2.1.1 Linear Discriminant Analysis (LDA)

The LDA classifier assumes that the training data is generated by a Gaussian process where a sample $\boldsymbol{x}^{(i)}$ from class $y^{(i)}$ is sampled with probability:

$$p(\boldsymbol{x}^{(i)}, y^{(i)}; \pi, \mu_0, \mu_1, \Sigma_0, \Sigma_1) = \mathcal{N}(\mu_0, \Sigma_0)(1-\pi) + \mathcal{N}(\mu_1, \Sigma_1)\pi, \tag{2.1}$$

where π is the probability of selecting the Gaussian for class 1, and $\mathcal{N}(\mu_k, \Sigma_k)$ is the Gaussian for class k with mean μ_k and covariance matrix Σ_k. In LDA we also make an additional assumption that both covariances matrix are equal. We then try to find the parameters $(\pi, \mu_0, \mu_1, \Sigma)$ that will maximize the probability that the training data was generated by the described Gaussian process. Once we find the parameters we can classify a new example $\boldsymbol{x}^{(j)}$ by checking whether the following ratio is greater or less than 1:

$$\frac{p(y=1|\boldsymbol{x}^{(j)})}{p(y=0|\boldsymbol{x}^{(j)})} = \frac{p(\boldsymbol{x}^{(j)}|y=1)\pi}{p(\boldsymbol{x}^{(j)}|y=0)(1-\pi)} = \frac{\mathcal{N}(\mu_1, \Sigma)\pi}{\mathcal{N}(\mu_0, \Sigma)(1-\pi)} \tag{2.2}$$

The decision boundary for the above rule will be linear on the vector $\boldsymbol{x}^{(j)}$.

2.1.2 Quadratic Discriminant Analysis (QDA)

The QDA classifier is similar to the LDA, but it does not make the assumption that the covariances matrices of each class are equal. The decision boundary dividing the two classes of interest, in this case, will be described by a quadratic curve.

2.1.3 Support Vector Machine (SVM)

There are many variants of SVM classifiers. We will start by explaining the simplest one which is the hard-margin LinearSVM (VAPNIK; LERNER, 1963). In this case, the method will try to find a separating hyperplane that is as far as possible from the nearest example of either class. This is equivalent to first finding two parallel hyperplanes that separate the two classes and are as far as possible from each other, and then outputting a third parallel hyperplane that is halfway from the other two. The examples that touch the first two hyperplanes are called support vectors.

In the hard-margin case, the two classes need to be linearly separable, and if it is not, the optimization method will fail. The soft-margin SVM (CORTES; VAPNIK, 1995) was created to work with data that cannot be separated by a hyperplane by introducing a hyperparameter C that controls the trade off between the width of the margin and amount of margin's violation in the optimization problem. In this thesis, we always use the soft-margin version of SVM.

In addition to the LinearSVM, there also exist versions that can find non-linear boundaries. This can be done efficiently due to the kernel trick (BOSER; GUYON; VAPNIK, 1992) where the dot product performed inside the SVM can be replaced by a kernel function $K(.,.)$ that receives two vectors as input. Depending on the choice of the function it can implicitly represent the dot product of two vectors in a higher or infinite dimensional space. In this thesis, we use the Radial Basis Function (RBF) kernel which has the following formula:

$$K(\boldsymbol{x}, \boldsymbol{x'}) = exp(-\frac{\|\boldsymbol{x} - \boldsymbol{x'}\|}{2\sigma^2}), \tag{2.3}$$

where σ is a hyperparameter.

The most common algorithm to find the SVM solution for a given training set is the sequential minimal optimization (PLATT, 1998), but this algorithm does not scale well to many samples. For big datasets, one can instead use Stochastic Gradient Descent (SGD) to minimize the hinge loss what will lead to the same solution as the other algorithm, but it only works for the LinearSVM.

2.1.4 *K*-Nearest Neighbors (KNN)

At test time, the KNN classifier will simply assign the most common label among the K training samples that are nearest to the test example $\boldsymbol{x}^{(i)}$ being classified. The naive implementation of KNN does not need training phase, but the method at test time can be made faster by building a data structure at training time that offers a fast way to find the nearest neighbors, such as kd-tree or ball-tree.

2.1.5 Random Forest (RF)

Random Forest is an ensemble learning method where several decision trees are trained, and then a new test example is classified by assigning to it the most common prediction of the decision trees. Each tree is trained on a random sample (with replacement) of the training set. This ensembling procedure decreases the chance of overfitting and increases the model's generalizability. Also, in each tree node only a random subset of the features are considered for finding the best split, and also the trees are grown to largest extent possible (BREIMAN, 2001).

2.1.6 Deep Neural Network (DNN)

DNNs are Artificial Neural Networks (ANN) which have multiple hidden layers of non-linear functions. In this thesis, we use two types of DNNs: Multi-layer Perceptron (MLP), and Convolutional Neural Network (CNN). The MLP utilized in this thesis can also be considered a DNN because it has four hidden layers. All neural networks need to minimize a loss function, and in this thesis, we will use the cross-entropy loss.

The MLP is the most common type of feedforward neural network where all the neurons of a given layer is fully-connected to all neurons of the following layer. It will try to find a mapping $\boldsymbol{y} = f(\boldsymbol{x}; \mathbb{W})$ that is parameterized by the set of weight matrices and bias vectors $\mathbb{W} = \{(\mathbf{W}^l, \mathbf{b}^l) | 1 \leq l \leq L\}$, where L is the number of layers. For every layer l in $\{1 \leq l \leq L-1\}$ its output will be $\mathbf{h}^l = \sigma(\mathbf{W}^{l\top}\mathbf{h}^{l-1} + \mathbf{b}^l)$, where σ is a element-wise non-linear differentiable activation function, $\mathbf{W}^l_{i,j}$ is the weight of the edge connecting the neuron i from layer $l-1$ to neuron j from layer l, and $\mathbf{h}^0$ equals the input $\boldsymbol{x}$ of the MLP. For the final layer $l = L$ the output will be $\boldsymbol{y} = \mathbf{h}^L = softmax(\mathbf{W}^{L\top}\mathbf{h}^{L-1} + \mathbf{b}^L)$,

where $softmax(\mathbf{z})_j = e^{z_j}/(\sum_{k=1}^{m} e^{z_k})$ for j in $1 \ldots m$, where m is the size of vector $\mathbf{z}$. The MLP will learn by changing its weights in the direction that decreases most the cross entropy loss. This is generally accomplished by using stochastic gradient descent (SGD) and by calculating the gradients using the backpropagation algorithm (LINNAINMAA, 1976; WERBOS, 1981; RUMELHART; HINTON; WILLIAMS, 1988).

The success of Convolutional Neural Networks (CNNs) in recent years is one of the main reasons the term Deep Learning became popular. Despite being created in 1998 (LECUN et al., 1998) it only became popular in 2012 when it won the Imagenet image classification competition (KRIZHEVSKY; SUTSKEVER; HINTON, 2012). As in the MLP, a typical CNN will also use a softmax function in its last layer and the cross-entropy loss. For now on we will use the terms layer and block interchangeably. We will also limit the description of the CNN for the case of 3-dimensional input (e.g., an image). The first layers of the CNNs consist of convolutional and pooling blocks. The last layers may consist of fully-connected blocks. The convolutional block of layer l has a set of weight filters $\{\mathbf{F}_c^l \mid 1 \leq c \leq n_f^l\}$, where $\mathbf{F}_c^l$ is a tensor (multidimensional array), with dimensions ($h_f^l \times w_f^l \times d^{l-1}$), that is convolved with the layer's input $\mathbf{H}^{l-1}$, with dimensions ($h_i^{l-1} \times w_i^{l-1} \times d^{l-1}$). Notice that both the depth of the filter and input tensor are the same as the convolution is only performed along the spatial dimensions. The output of the convolutional layer will be another tensor where its channel c will be the matrix $\mathbf{H}_{:,:,c}^l = \mathbf{F}_c^l * \mathbf{H}^{l-1}$, where the symbol $*$ is the convolution operator. The number of channels of the output tensor is equal to the number of filters of the given layer. The convolutions may also be strided what means that the filter center is placed after every s pixels during the convolution (i.e., there is a gap of $s-1$ pixels between two convolution window centers). After the convolutional layer, an element-wise non-linear activation function is applied to the tensor. Typically a rectified linear unit (NAIR; HINTON, 2010) activation function is used.

The pooling layer objective is to reduce the spatial size of the tensors being processed by the layers[1]. For instance, a pooling layer with a window size of 2×2 and stride $s = 2$ will produce a new tensor with half of its previous size. This is done by taking the maximum (or the average depending on the pooling type) value of every 2×2 non-overlapping window[2]. The pooling operation is done independently for each channel of the input tensor.

[1] Another way of decreasing the spatial size of the tensors is to use convolutions with strides greater than one.

[2] It could be overlapping if window size was 3×3 and the stride was equal to 2, for instance.

A typical CNN architecture starts with a convolutional layer, followed by an activation function, and these two are repeated until we decide to place the first pooling layer. Then we repeat the previous pattern. When the spatial size of the tensors become small enough, we then flatten it in a vector and then proceed with 0 or more fully-connected layers, before the final softmax layer. A possible CNN architecture might be: INPUT->CONV->ACT->CONV->ACT->POOL->CONV->ACT->CONV->ACT->POOL->FC->FC->SOFTMAX->OUTPUT, where ACT stands for activation function, FC for Fully Connected layer, and the other names are self-explanatory. As all the blocks used in the CNN is differentiable, then the backpropagation algorithm can be applied to it.

2.2 Remote Sensing

One of the definitions of remote sensing is "the measurement of object properties on the earth's surface using data acquired from aircraft and satellites" (SCHOWENGERDT, 2006). In this thesis, the object property we are interested is whether it belongs to the tropical forest class or not. We will also use Landsat images which are acquired by sensors on board of satellites.

The Landsat project is a joint initiative between the United States Geological Survey (USGS) and NASA that collects space-based moderate-resolution land multispectral images. Landsat 8 is the name of the latest satellite launched by the project, and it carries two sensors: the Operational Land Imager (OLI), and the Thermal Infrared Sensor (TIRS). In this thesis, we will use just the data obtained by the OLI sensor, which focuses on collecting radiation reflected on the ground that originates from the sun[3]. This sensor has eight 30 m resolution bands where each one captures energy at a different interval from the electromagnetic spectrum. It also has one broad band at 15 m resolution that captures energy within a larger interval in the spectrum. We will only use the 30 m resolution bands in this thesis[4].

In the next paragraphs, we want to define two physical quantities that will useful in this thesis, namely the **surface reflectance**, and the **top-of-atmosphere reflectance**. However, first, we will need to understand what is spectral radiance and how the sensor

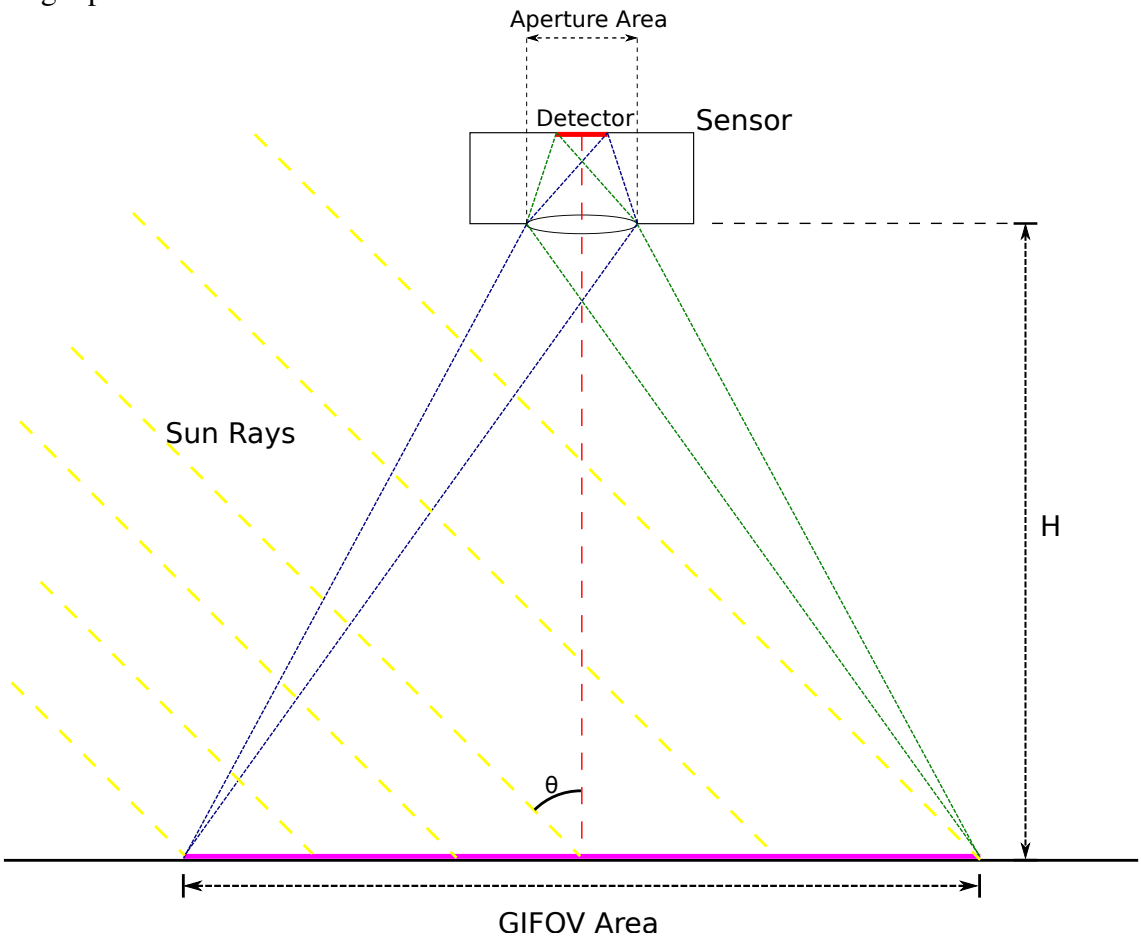
[3]Other possibilities could be thermal radiation emitted from the earth's surface or a laser beam originating from the satellite that is reflected on the earth's surface.

[4]Even though we use the 30 m resolution bands, we evaluate the produced deforestation maps at the resolution of 60 m by ignoring deforestation blobs smaller than 60×60 meters. More information can be seen in Section 5.2.7.

can capture that quantity. In Figure 2.2 is shown an illustration of a simple hypothetical remote sensor that produces an image with just one pixel and has only one band capturing energy between the wavelengths λ_a and λ_b ($\lambda_a < \lambda_b$). We also suppose the detector plane is parallel to the earth's surface, and that the Ground-projected Instantaneous Field of View (GIFOV) terrain is horizontally flat. The physical quantity that the detector will directly measure is the amount of electromagnetic energy in the specified wavelength range that hit the detector area during the interval of time τ that the sensor's shutter remained opened. Supposing for the moment that we do not have an atmosphere, then the amount of energy captured by the sensor is also equal to the energy of solar radiation that hit the GIFOV region and was reflected within the specified spectral range in the direction of the sensor's aperture. Notice that a ray from the sun being reflected outside of the GIFOV area and then hitting the lens will be mapped to a region outside of the detector and thus its energy will not be accumulated.

Although the sensor measures electromagnetic energy, it is useful to convert it to spectral radiance. The spectral radiance $L(\mathbf{x}, \boldsymbol{\omega}, \lambda)$ for position $\mathbf{x}$, direction $\boldsymbol{\omega}$, and wavelength λ is the amount of electromagnetic energy per unit time that goes through an infinitesimal area dA, centered at $\mathbf{x}$ and perpendicular to the direction $\boldsymbol{\omega}$, in the directions contained in an infinitesimal solid angle $d\Omega$ around the direction $\boldsymbol{\omega}$, and with wavelength from a infinitesimal interval $d\lambda$ centered at λ, divided by dA, $d\Omega$, and $d\lambda$. The spectral radiance units are watt per steradian per square meter per micrometer $(W/(sr \cdot m^2 \cdot \mu m))$. It is useful to convert energy to spectral radiance because we then can compare measurements made by different sensors on board different satellites when everything else is held constant. On our hypothetical satellite we can convert the accumulated energy during time τ to spectral radiance by dividing it by τ, $(\lambda_b - \lambda_a)$, detector area (A_d), and the solid angle Ω_{gv} at the sensor aperture subtended by the GIFOV area. This approximation can be considered a proper derivative to the extent that the energy density does not change much within each interval. For instance, supposing the surface material on the GIFOV region is homogeneous, as the perpendicular distance H from the sensor to earth's surface is much greater than the GIFOV size, then the energy density coming from different directions within Ω_{gv} will be very similar since the distance from the sensor to any point in the GIFOV won't change much. Also, any point in GIFOV reflects the sunlight uniformly in all directions as we assume a Lambertian surface. If the GIFOV contains different materials, then the fraction of sunlight reflected may change with position, so the energy density received in each direction within Ω_{gv} may also vary. In this case, the previous spectral

Figure 2.2: Sensor receiving light from the sun that is reflected on the earth's surface. The sensor, which is hypothetical, produces images with only one pixel and has only one band. θ is the angle of incidence. H is the height of the sensor from the earth's surface. The yellow lines represent the sun rays. GIFOV area is a square region corresponding to the Ground-projected Instantaneousness Field of View. The detector is a square red region inside the sensor that captures light coming from the GIFOV, and it represents a single pixel.



radiance can be considered an average among all direction in Ω_{gv}. If the sensor's band wavelength range $[\lambda_a, \lambda_b]$ is not small enough, then we can also say that we measure the average spectral radiance over this interval.

Now let us try to relate the incoming average spectral radiance at the sensor with the spectral radiance leaving the earth's surface still supposing no atmosphere. The total energy K received at the sensor in time τ is:

$$\begin{aligned} K &= \int_{A_a} \int_{\Omega_{gv}} (\lambda_b - \lambda_a)\tau L_a(\mathbf{x}_a, \boldsymbol{\omega}, \lambda_m)(\boldsymbol{\omega} \cdot \mathbf{n}) d\Omega dA \\ &= (\lambda_b - \lambda_a)\tau \int_{A_a} \int_{\Omega_{gv}} L_a(\mathbf{x}_a, \boldsymbol{\omega}, \lambda_m)(\boldsymbol{\omega} \cdot \mathbf{n}_a) d\Omega dA, \end{aligned}$$

where A_a is the aperture area, Ω_{gv} are the directions pointing to the GIFOV, $\mathbf{n}_a$ is a normal vector to the aperture's plane, $(\boldsymbol{\omega} \cdot \mathbf{n}_a)$ equals the cosine of the angle between these two vectors, and $L_a(\mathbf{x}_a, \boldsymbol{\omega}, \lambda_m)$ is the spectral radiance at the aperture's position $\mathbf{x}_a$, direction $\boldsymbol{\omega}$, and wavelength $\lambda_m = (\lambda_a + \lambda_b)/2$. As the distance from the sensor to earth is much greater than the aperture size then we can say that the spectral radiance won't change much for different aperture positions $\mathbf{x}_a$, so we can pull L_a out of the area integration, and also because the solid angle Ω_{gv} is very small, then $(\boldsymbol{\omega} \cdot \mathbf{n}_a)$ can be replace by one:

$$\begin{aligned} K = (\lambda_b - \lambda_a)\tau (\int_{A_a} dA) \int_{\Omega_{gv}} L_a(\mathbf{x}_a, \boldsymbol{\omega}, \lambda_m) d\Omega \\ K = (\lambda_b - \lambda_a)\tau A_a \int_{\Omega_{gv}} L_a(\boldsymbol{\omega}, \lambda_m) d\Omega \end{aligned} \tag{2.4}$$

Now it can be shown that by the spectral radiance invariance property that $L_a(\boldsymbol{\omega}, \lambda_m) = L_{es}(\mathbf{x}_{es}(\boldsymbol{\omega}), -\boldsymbol{\omega}, \lambda_m)(-\boldsymbol{\omega} \cdot \mathbf{n}_{es})$, where L_{es} is the spectral radiance leaving the earth's surface at point $\mathbf{x}_{es}(\boldsymbol{\omega})$, which is the intersection of the line leaving the sensor's aperture at direction $\boldsymbol{\omega}$ with the earth's surface plane, and $\mathbf{n}_{es}$ is a vector normal to the earth's surface. Again, $(-\boldsymbol{\omega} \cdot \mathbf{n}_{es})$ will always be very near one, so we can say that $L_a(\boldsymbol{\omega}, \lambda_m) = L_{es}(\mathbf{x}_{es}(\boldsymbol{\omega}), -\boldsymbol{\omega}, \lambda_m)$. As $\mathbf{x}_{es}$ is a function of $\boldsymbol{\omega}$, we simplify the formula to $L_a(\boldsymbol{\omega}, \lambda_m) = L_{es}(-\boldsymbol{\omega}, \lambda_m)$. By substituting L_a by L_{es} in Equation 2.4, dividing both sides by the solid angle, and moving the multiplication constants to the left part of the equation, we get:

$$\bar{L}_a = \frac{K}{(\lambda_b - \lambda_a)\tau A_a \Omega_{gv}} = \frac{\int_{\Omega_{gv}} L_{es}(-\boldsymbol{\omega}, \lambda_m) d\Omega}{\Omega_{gv}} = \bar{L}_{es} \tag{2.5}$$

So we can say that the sensor is indirectly measuring the average surface spectral radiance of the GIFOV area.

Now let us write the surface spectral radiance in terms of surface reflectance. The **surface reflectance** is the fraction of the electromagnetic power that is reflected at the earth's surface. This quantity is a property of the material at the surface, and thus it is a useful measure in remote sensing applications where we want to classify each pixel. More precisely the **surface reflectance** $\rho(x, y, \lambda)$ at position (x, y) and wavelength λ is defined as $\rho(x, y, \lambda) = L_{es}(x, y, \lambda)\pi/E(x, y, \lambda)$, where $E(x, y, \lambda)$ is the spectral solar irradiance at the earth's surface position (x, y). Spectral irradiance has the same units of spectral flux density $(W/(m^2 \cdot \mu m))$. Supposing $E(x, y, \lambda)$ is constant over the GIFOV (what is a good approximation), and if we know $E(x, y, \lambda)$ value then the satellite sensor can also measure the average surface reflectance indirectly in the GIFOV region, and therefore the "average" material of this region, supposing no atmosphere.

Let us now add the atmosphere in our model and try to derive the total at aperture spectral radiance L_a^t in terms of other basic physical quantities. For simplicity let us suppose that the surface reflectance is constant in the whole GIFOV, and when we refer to a quantity in the earth's surface, we always mean a position inside the GIFOV. In the following derivations, we avoid the λ notation also for simplicity, but the reader should have in mind that all quantities depended on the wavelength. The total at aperture spectral radiance L_a^t can be decomposed into three terms: $L_a^t = L_a^{usr} + L_a^{dsr} + L_a^{up}$, where L_a^{usr} correspond to the unscattered/unabsorbed surface-reflected radiation, L_a^{dsr} is the down-scattered surface-reflected skylight, and L_a^{up} is the up-scattered path radiance (SCHOWENGERDT, 2006).

Let us begin deriving a formula to the unscattered/unabsorbed surface-reflected spectral radiance L_a^{usr}. This radiance at the aperture is the remaining radiance of the upwelling radiance at earth's surface, derived from unscattered/unabsorbed reflected sun light, L_{es}^{usr} that was not absorbed or scattered by the atmosphere in the path from the earth to the sensor (view path). The ratio L_a^{usr}/L_{es}^{usr} is a characteristic of the atmosphere along the view path and we give to it the name of view path transmittance $\tau_v(\lambda)$. Then, $L_a^{usr} = \tau_v L_{es}^{usr}$. L_{es}^{usr} can be written in terms of the unscattered/unabsorbed downwelling solar irradiance at earth's surface E_{es}^u, so the previous formula becomes $L_a^{usr} = \tau_v(\rho/\pi)E_{es}^u$, where ρ is the surface reflectance. The E_{es}^u can be written in terms of the solar spectral irradiance E_{toa} on the top of the earth's atmosphere as $E_{es}^u = \tau_s E_{toa} cos(\theta)$, where τ_s is the atmosphere transmittance along the solar path, and θ is the incident angle (i.e., angle between the earth's surface normal vector and a vector pointing to the sun). E_{toa} is a well known quantity and varies with distance from the earth to the sun during the year. The

final formula for L_a^{usr} is then:

$$L_a^{usr} = \tau_v \frac{\rho}{\pi} \tau_s E_{toa} cos(\theta) \tag{2.6}$$

The formula for the down-scattered surface-reflected skylight L_a^{dsr} is:

$$L_a^{dsr} = \tau_v \frac{\rho}{\pi} E_{es}^{sky}, \tag{2.7}$$

where E_{es}^{sky} is the irradiance at the surface due to skylight, which can be measured with ground-based instruments (SCHOWENGERDT, 2006). Skylight are the sun rays that are scattered on the atmosphere particles in the direction of the earth's surface.

The up-scattered path radiance L_a^{up} is the radiation that gets scattered upward in the direction of the sensor. This quantity can vary with position in the earth's surface and with view angle. For the field of a view of a single detector (or pixel), it can be assumed constant.

The final formula for the total at aperture spectral radiance L_a^t can be written as:

$$\begin{aligned} L_a^t &= L_a^{usr} + L_a^{dsr} + L_a^{up} \\ L_a^t &= \tau_v \frac{\rho}{\pi} \tau_s E_{toa} cos(\theta) + \tau_v \frac{\rho}{\pi} E_{es}^{sky} + L_a^{up} \\ L_a^t &= \frac{\rho}{\pi} \tau_v (\tau_s E_{toa} cos(\theta) + E_{es}^{sky}) + L_a^{up} \end{aligned} \tag{2.8}$$

The Equation 2.8 shows that the total at sensor radiance can be transformed to surface reflectance by subtracting a bias component, and by dividing by a multiplicative component. It is important to note that even if we know precisely quantities necessary for calculating the surface reflectance, it can still be challenging to classify pixels to their correct classes because of the natural variability of some materials, and due to a possible coarse spectral quantization performed by sensor's hardware (SCHOWENGERDT, 2006).

The **top-of-atmosphere (TOA) reflectance** ρ_{toa} is the ratio of the total at-sensor radiance L_a^t to the solar spectral irradiance E_{toa} on the top of the earth's atmosphere corrected by the angle of solar light incidence, and multiplied by π. The formula is given below:

$$\rho_{toa} = \frac{L_a^t \pi}{E_{toa} cos(\theta)} \tag{2.9}$$

The TOA reflectance for the same position and atmospheric conditions is invariant to the time of the year, and sun angle. Surface reflectance is better than TOA reflectance

because the earlier is additionally invariant to atmospheric conditions, but the later is easier to obtain.

3 RELATED WORK

This chapter will review the relevant literature to this thesis. As this work is mainly about image normalization/correction, and deforestation detection, we will divide this chapter into two section covering each of these topics.

3.1 Image Correction and Normalization

One of the main challenges in achieving a high classification accuracy in remote sensing images is anchoring images from different dates and locations to the same reference. What we mean by that is that if two materials are identical and are placed in different places on the earth's surface, and images are taken from it from satellites for different dates, then their pixels for the two different images corresponding to the same material should have identical spectral values. If we can devise a normalization that achieves that, posterior classification would be significantly easier. In this section, we review correction/normalization methods that can be considered as alternatives to the first two stages (Classify-Normalize) of the CNC framework. We divide this section into two subsections. In subsection 3.1.1, we review atmospheric corrections procedures that try to estimate the absolute surface reflectance values for all pixels (see Section 2.2 for the definition of surface reflectance). Subsection 3.1.2, review methods that normalize the images to make it easy to compare data from different dates and locations, but does not output values in surface reflectance units.

3.1.1 Absolute Radiometric Correction

The most traditional methods for obtaining the surface reflectance from the at-sensor radiance are based on rigorous radiation transfer models and require auxiliary data, often achieved by in-situ measurements, to acquire several atmospheric parameters for the model, such as average atmospheric pressure, integrated water vapor and ozone content, and aerosol characteristics. Examples of these models are the 6S code (Second Simulation of the Satellite Signal in the Solar Spectrum) (VERMOTE et al., 1997), and the MODTRAN code (MODerate resolution atmospheric TRANsmission) (BERK; BERNSTEIN; ROBERTSON, 1987). Other simpler methods such as the Dark Object Subtraction (DOS)

normalization requires just information derived from the image itself to obtain the absolute surface reflectance values (JR, 1988; JR, 1989; JR, 1996).

The United States Geological Survey (USGS) provides surface reflectance Landsat 8 OLI images which are generated by using the algorithm LaSRC[1] from Vermote et al. (2016), which is based on a refined version of the 6SV code that takes advantage of the narrow OLI spectral bands. The 6SV code (KOTCHENOVA et al., 2006) is a vectorial version of the 6S code. In the experiments chapter, we compare the utility of the LaSRC correction versus the normalization provided by the first two stages of the CNC framework in the deforestation detection context.

Dark Object Subtraction (DOS) is a popular option for atmospherically correcting the images when the atmospheric parameters needed in the more complex models (e.g., 6S, MODTRAN) are not available. DOS is a strictly image-based atmospheric correction technique that supposes that if some dark objects in the scene have pixel values greater than zero, then this is due to the up-scattered path radiance L_a^{up}. So the simplest form of DOS just subtracts the minimum value of a given band of a scene from all the pixels. There exist several updated DOS versions that improve over the basic technique (JR, 1988; JR, 1989; JR, 1996; SONG et al., 2001). They differ from each other on the simplifying assumptions they make on Equation 2.8 from Section 2.2. If we isolate the surface reflectance ρ from Equation 2.8 we get:

$$\rho = \frac{\pi(L_a^t - L_a^{up})}{\tau_v(\tau_s E_{toa} cos(\theta) + E_{es}^{sky})}, \tag{3.1}$$

where the parameters that are estimated differently in each DOS version are the view path transmittance τ_v, the solar path transmittance τ_s, and the irradiance at the earth's surface due to skylight E_{es}^{sky}. The DOS1 assumes both transmittance to be unit, and E_{es}^{sky} to be zero (JR, 1989). DOS2 approximate τ_s by $cos(\theta)$ for the bands 1-4 of Landsat TM sensor, and unity for bands 5 and 7 (JR, 1996; SONG et al., 2001). For more complex versions of DOS we refer the reader to Song et al. (2001). Although the DOS method can be useful when atmospheric informations are not available, if instead these parameters can be obtained, then the 6S code is preferred as it produce more precise results (NAZEER; NICHOL; YUNG, 2014).

[1] In this thesis, we refer to the Landsat 8 surface reflectance algorithm as LaSRC (Landsat Surface Reflectance Code) because it is the name used in the online product guide: <https://landsat.usgs.gov/sites/default/files/documents/provisional_lasrc_product_guide.pdf>.

3.1.2 Relative Radiometric Normalization

In this section, we cover relative radiometric normalization (RRN) techniques that try to put two or more images in a common spectral reference but do not aim necessarily to obtain the absolute surface reflectance values as output. Most RRN methods require a reference image to which the other images will be normalized. On the other hand, the proposed CNC framework does not require choosing a reference image, as the median forest vector of all images will be translated to zero.

Many popular RRN methods are based on pseudo-invariant features (PIFs) which correspond to objects which are known to not change its surface reflectance throughout the year (SCHOTT; SALVAGGIO; VOLCHOK, 1988). If the same PIF objects can be found in two images, then one of the images can be normalized to the other's spectral reference by linear regression, for instance. The problem now is finding the PIFs objects in an automatic way. The iteratively re-weighted multivariate alteration detection (IR-MAD) transformation can find PIF regions automatically in the difference between two co-registered images (CANTY; NIELSEN, 2008). However, the requirement for the images to overlap geographically is problematic when we want, for instance, to train a classifier in a set of spatially disjoint images, and then apply the trained model to another set of non-overlapping images.

RRN methods can also be viewed as a subcategory of domain adaptation methods where the data distributions are adapted instead of the models themselves. Domain adaptation is a subfield of machine learning that tries to tackle the problem of learning from a source domain and then adapting the learned model to the target domain or aligning the target domain data to the source's data. Tuia, Persello and Bruzzone (2016) made a table listing several domain data adaptation methods and their characteristic which we reproduce in Table 3.1 with the addition of the CNC framework. The column "Labels in S" indicates methods which can leverage the information given by the labels in the source domain during adaptation to the target domain. The "No Labels in T" column refers to semi-supervised or unsupervised methods which do not need labels in the target domain. Multi-source methods allow the alignment of more than two domains simultaneously. The "Unpaired" column indicates if the method requires images of different domains to overlap geographically. The ability of a method adapting between domains from spaces with different dimensionality is indicated on column "Multi-D". Finally, the column "Nonlinear" refers to the methods that perform a nonlinear transformation during

the data adaptation.

As can be seen from the table only the CNC framework has the four first columns checked. It is important for a method to have the multi-source property, since aligning multiple source domains may allow the classifier to find a more discriminative decision rule for the training set, and also for the target domains in the test set. Excluding the CNC, there are only four multi-source methods in Table 3.1, but they either need labels on the target domain, or they require overlapping source and target images. The description of each of the methods in Table 3.1 is beyond the scope of this thesis, and the interested readers are referred to Tuia, Persello and Bruzzone (2016) for a complete review.

Table 3.1: Domain data adaptation methods characteristics comparison. This table is a slightly modified version from the one in Tuia, Persello and Bruzzone (2016). For instance, we correct the characteristics of the MA method (YANG; CRAWFORD, 2016) which in Tuia, Persello and Bruzzone (2016) wrongly stated that it was multi-source and that it was linear.

Method	Labels in S	No Labels in T	Multi-Source	Unpaired	Multi-D	Nonlinear
PCA	×	✓	×	✓	×	×
KPCA (NIELSEN; CANTY, 2009)	×	✓	×	✓	×	✓
(SS)TCA (MATASCI et al., 2015)	×✓	✓	×	✓	×	✓
MAD (CANTY; NIELSEN, 2008)	×✓	×✓	✓	×	✓	×
KCCA (VOLPI; CAMPS-VALLS; TUIA, 2015)	×✓	×✓	✓	×	✓	✓
MA (YANG; CRAWFORD, 2016)	✓	✓	×	✓	✓	✓
KMA (TUIA; CAMPS-VALLS, 2016)	✓	×	✓	✓	✓	✓
SSMA (TUIA et al., 2014)	✓	×	✓	✓	✓	×
GM (TUIA et al., 2013)	×	✓	×	✓	×	×
Proposed CNC framework	✓	✓	✓	✓	×	×

Source: Modified from Tuia, Persello and Bruzzone (2016) to include the CNC framework, and with correction for wrong markings in the MA method.

It is also worth mentioning that the normalization performed in the first two steps (Classify-Normalize) of the CNC is similar to the one in Hansen et al. (2008) where the median spectral signal of regions of interest (forest in this context) was also used to normalize the images before further analysis, but it relied on an auxiliary preexisting MODIS-based land cover product to determine forest regions. On the other hand, our framework does not depend on auxiliary data, as it can automatically find forest regions, and this is important as not always those might be available. Although subtracting the forest median vector from all pixels was previously already known to be a good normalization strategy, we are the first to propose and show, to the best of our knowledge, that the forest median vector can be estimated with enough precision (to produce a proper normalization) by classifiers trained and tested on unnormalized images, and thus making the procedure fully automatic at inference time.

3.2 Deforestation Detection

The most famous works involving deforestation detection in the literature use a methodology that is not fully automatic. They also focus more on the quantification of deforestation in a given region than the method itself. Examples of works that focus on the quantification of deforestation in a particular area and period are Potapov et al. (2012), Hansen et al. (2013), Grinand et al. (2013), and Kim et al. (2014). It is also very rare to compare methods in the deforestation detection literature making it challenging to know which method is the best in some sense.

One of the most famous deforestation detection work is that of Hansen et al. (2013). It mapped forest cover change in the whole world for each year between 2000 and 2012 at 30 m resolution. Forest was defined as any vegetation taller than 5 m in height. Its results can be interactively visualized online[2]. In that work, they use a bagged decision tree (LOH, 2011) to try to relate percent tree cover, forest loss, and forest gain with custom-made spectral features derived from a time-series. The method requires observations from over three dates to classify a single pixel. This contrasts with the methodology we adopt in this thesis where we require just two images to verify the changes between them. Bitemporal change detection is useful for alert systems which cannot afford to wait to collect enough samples to increase its certainty. Also, Hansen et al. (2013) uses auxiliary data from MODIS to perform the atmospheric correction, while our framework require just the input Landsat images at inference time. It is hard to reproduce the work of Hansen et al. (2013) since they did not make publicly available their training set and ground truth labels[3]. Its global loss between 2000 and 2012 producer's and user's accuracy measured from 1500 samples were 87.8% and 87%, respectively. It is important to note that since the period is long, the deforestation polygons will be larger and easier to detect than those from this thesis experiments which use periods of about one year.

More recently, Hansen et al. (2016) focused on the problem of detecting deforestation in a shorter period. The aim was to develop a system for forest disturbance alerts. As in Hansen et al. (2013), this work also uses MODIS as auxiliary data for normalizing the Landsat images. Its user's and producer's accuracy measured from 1294 samples were, respectively, 86.5% and 67%. It is important to note that this system produced deforestation maps at a resolution of 30 m, while our proposed method results were evaluated at

[2]<https://earthenginepartners.appspot.com/science-2013-global-forest>

[3]Nonetheless, they provide loss, gain, forest cover percentage images predicted by their trained model.

60 m resolution, despite the input images having 30 m resolution (see Section 5.2.7).

The Project for Monitoring Deforestation in the Legal Amazon by Satellite (PRODES) is a program of Brazil's National Institute for Space Research (INPE). The program quantifies and maps the deforestation in the Brazilian Amazon annually. To that end, it uses a semi-automatic approach that involves visual interpretation of Landsat images with the help of algorithms such as linear mixture model, segmentation, and unsupervised clustering (SHIMABUKURO et al., 1998; HANSEN; LOVELAND, 2012). Although the resolution of the deforestation polygons created by PRODES are of 60 m, the final product only considers polygons with area greater than 5.76 ha (57,600 m^2) (MALDONADO et al., 2007)[4]. Deforestation maps produced by the method proposed in this thesis will be compared with those of PRODES for the same start and end image acquisition dates.

It is also worth mentioning the adaptive semi-supervised method of Zanotta et al. (2015) that was also applied to detecting deforestation on the Amazon rain forest, and does not need auxiliary data, and neither visual interpretation by an analyst. The method was shown to be more accurate than PRODES on the evaluated regions. Their method requires ground truth annotations of changes of interest just in the first pair of images in a time-series. Then it adapts the decision rule, learned on the source pair, based on 3D change vector analysis (CVA) to the upcoming images target pairs from the time-series. In this way, the method can, for instance, avoid an explicit radiometric correction as a preprocessing step, since the adaptation phase can change the decision rule to also ignore changes due to different atmospheric conditions in the new pair of images in relation to the previous. Although promising, this approach differs from the objectives in this thesis since we are interested in bitemporal change detection, and we do not expect to have a time-series available. Although time-series can give additional information and confidence in a prediction, it is worth limiting our research to just the bitemporal setting, since in this way we can investigate what are its limits and because we can always include or adapt a bitemporal method to a time-series procedure. This can be done through ensembling for instance.

[4]In practice this means that the polygons contours will be precise within the 60 m range and that all polygons with area greater than 5.76 ha will be deleted from the final deforestation map.

4 PROPOSED CLASSIFY-NORMALIZE-CLASSIFY FRAMEWORK

In this chapter, we explain in detail the CNC framework. Section 4.1 will formulate the problem, and describe the CNC in mathematical terms. The algorithmic description in the context of forest classification will be made in Section 4.2.

4.1 Problem Formulation

Let $\mathbb{X}_d$ be a dataset of pixel-label pairs $(\boldsymbol{x}_d^{(i)}, y_d^{(i)})$, where $\boldsymbol{x}_d^{(i)}$ is a vector with a value for each channel of pixel i and $\boldsymbol{y}_d^{(i)}$ is a label that equals 1 when the pixel belongs to the class of interest (COI) and 0 otherwise (let us call the class with label 0 as NCOI). The subscript d indicates from which domain the data is sampled. Let P_d^1 be the COI distribution for domain d and P_d^0 be the NCOI distribution, and P_d be the two combined distribution ($\boldsymbol{x}_d^{(i)} \sim P_d$). From one domain to another the classes distribution may suffer a shift. Now let $\mathbb{X}$ be the union of all domains datasets ($\mathbb{X} = \bigcup_{d=1}^{n_d} \mathbb{X}_d$), where n_d is the number of different domains. Even if within the same domain the two classes were well separated, now in the union of the domains the distribution of different classes might become mixed, because of the data shift between domains, and thus making it difficult to discriminate between the classes.

The first two phases of the CNC framework (Classify-Normalize) will try to make a rigid translation of the distribution P_d such that the median vector $\boldsymbol{m}_d^1$ of P_d^1 is moved to the origin, for every domain d. The median vector is obtained by calculating the median value of $\{\boldsymbol{x}_d^{(i)}(c) \mid y_d^{(i)} = 1\}$ for each channel c. The translation can then be accomplished by subtracting, for every domain d, the median vector $\boldsymbol{m}_d^1$ from all pixels $\boldsymbol{x}_d^{(i)}$ of $\mathbb{X}_d$, and thus obtaining a new dataset $\mathbb{X}_d^*$ containing median centered pixels. Let us denote the union of these shifted datasets as $\mathbb{X}^* = \bigcup_{d=1}^{n_d} \mathbb{X}_d^*$. Our assumption is that the discrimination of the two classes of interest in the new dataset $\mathbb{X}^*$ is easier than in $\mathbb{X}$.

Now suppose we have a training set $\mathbb{X}_{train} = \bigcup_{d=1}^{n_{train}} \mathbb{X}_d$ for which we know the true labels $y_d^{(i)}$ and an unlabeled test set $\mathbb{X}_{test} = \bigcup_{d=n_{train}+1}^{n_d} \mathbb{X}_d$ that needs to be classified. The CNC framework proposes the following in the training phase. First we train a supervised classifier f_1 on $\mathbb{X}_{train}$. Then we build a COI median centered dataset $\mathbb{X}_{train}^*$ by using the true labels. Finally, we train a supervised classifier f_2 on $\mathbb{X}_{train}^*$.

On the testing phase, we first apply the trained f_1 on the $\mathbb{X}_{test}$ pixels to obtain the estimated labels $\hat{y}_d^{(i)}$. Then for each domain d belonging to the test set, we calculate the

estimated median $\hat{\boldsymbol{m}}_d^1(c)$ on the set $\{\boldsymbol{x}_d^{(i)}(c) \mid \hat{y}_d^{(i)} = 1\}$ for each channel c. Next, for each test domain d we subtract $\hat{\boldsymbol{m}}_d^1$ from all pixels in $\mathbb{X}_d$ and then make the union of all of those to obtain the normalized test set $\mathbb{X}_{test}^*$. Finally, we apply the trained f_2 classifier to the normalized pixels in $\mathbb{X}_{test}^*$ to obtain the final improved estimated labels $\hat{y}_d^{*(i)}$.

We decided to use just the median, and not the mean and standard deviation, for instance, to normalize the images because we conjecture that these other statistics will be more sensitive to errors in the first classifier segmentation. Furthermore, preliminary experiments shown on Appendix B where we normalized the images by subtracting the estimated COI mean and optionally also dividing by the estimated COI standard deviation, showed no improvement over just using the median.

4.2 CNC in the Context of Forest Classification

This section presents the CNC framework in the context of forest classification where each tile of an image represents a new domain d.

In the training stage, the first classifier is trained on top-of-atmosphere (TOA) reflectance image tiles and the second classifier is trained on tiles normalized with forest medians obtained from manually annotated forest regions. These forest annotations consist of sparse rectangles demarcated inside forest regions. Note that the manual annotation is only needed in the training phase, as in the testing phase, the forest regions are estimated with the first classifier. This makes the method fully automatic at test time after the initial offline training using a fixed set of tiles. Also note that differently than other solutions in the literature (SCHOTT; SALVAGGIO; VOLCHOK, 1988; BAO et al., 2012; CANTY; NIELSEN, 2008), the training tiles do not need to overlap with the test tiles geographically. The classifiers can be either non-contextual or contextual. In the later case, a window centered at each annotated pixel will be used as a training sample. The training phase is summarized in the Algorithm 1, and illustrated in the flowchart of Figure 4.1.

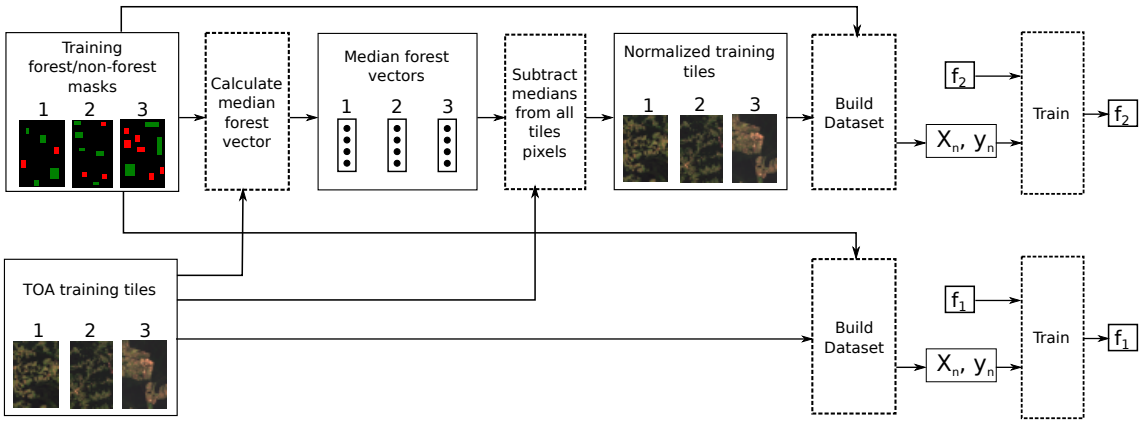
At testing time, the proposed CNC framework has three steps. First, a forest classifier f_1 is applied to every pixel (or to every overlapping window with a size corresponding to the classifier's context) of the input TOA reflectance tile to obtain an initial forest segmentation. Optionally, the forest mask produced by f_1 may also be eroded to reduce false positives at the expense of increasing false negative forest predictions, which is not very problematic because those are less likely to affect the estimated median[1]. Then, we com-

[1] Supposing that the erosion process of removing predicted forest pixels from the initial mask is unbiased.

pute the median multispectral vector for the estimated forest areas and subtract this vector from all pixels in the tile to translate the new forest median spectral vector to the origin. Finally, a second forest classifier f_2 is applied to every pixel (or overlapping window) of the normalized tile to produce the final forest segmentation mask that, as our experiments will show, is more precise than the first one. It is important to observe that the erosion operation may only be applied to the f_1 predicted mask if the number of remaining forest pixels is not less than a given threshold ($th = 1000$ pixels in our experiments) to prevent an inaccurate median estimate[2]. The test phase is also summarized in the Algorithm 2, and illustrated in the flowchart of Figure 4.2. The procedure to generate deforestation maps with the CNC framework trained for forest classification will be detailed in Section 5.2.6.

[2]As our test tiles had some large forest regions, the threshold was never violated in the experiments.

Figure 4.1: Flowchart for the training phase of the proposed CNC framework in the context of forest classification.



Algorithm 1 CNC Training.

Require: Unnormalized TOA reflectance training tiles $\mathbf{T}_u^i$, with i varying between 1 and n_t; ground truth images for each of the n_t tiles with the value 1 for forest pixels, 0 for non-forest pixels, and -1 for non-annotated pixels; and untrained classifiers f_1 and f_2.

1: Extract all pixels (or overlapping windows) with label different than -1 from all tiles $\mathbf{T}_u^i$ and build the unnormalized dataset $\mathbb{X}_{train}$ of pixel-label (or window-label) pairs $(\boldsymbol{x}^{(k)}, y^{(k)})$.

2: Train f_1 on $\mathbb{X}_{train}$.

3: **for** i from 1 to n_t **do** ▷ Normalize each tile.

4: Take the median of all the pixels spectral vector for which $y_i^{(k)} = 1$ to obtain the forest median vector $\hat{\boldsymbol{m}}_f^i$. ▷ medians are calculated independently for each channel.

5: Subtract $\hat{\boldsymbol{m}}_f^i$ from each pixel (or window) $\boldsymbol{x}_i^{(k)}$ to obtain the normalized samples $\boldsymbol{x}_i^{*(k)}$.

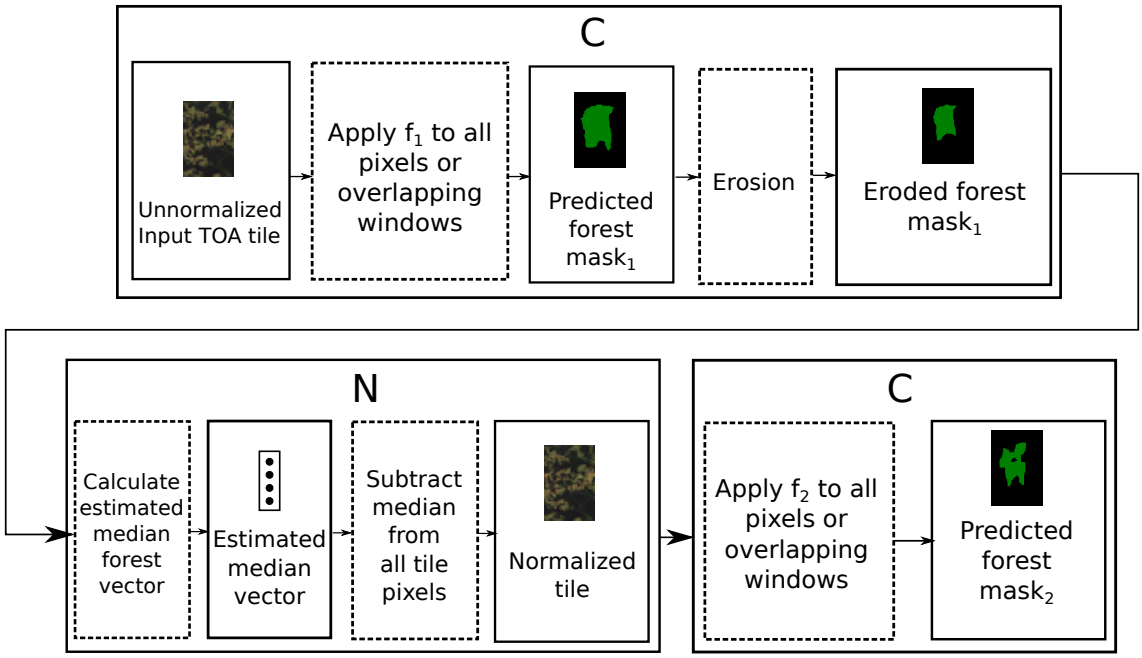
6: **end for**

7: Build a new dataset $\mathbb{X}_{train}^*$ containing all normalized samples $\boldsymbol{x}_i^{*(k)}$ from all tiles.

8: Train f_2 on $\mathbb{X}_{train}^*$.

9: **return** f_1 and f_2.

Figure 4.2: Flowchart for the inference phase of the proposed CNC framework in the context of forest classification.



Algorithm 2 CNC Inference.

Require: An unnormalized TOA reflectance tile image $\mathbf{T}_u$; trained classifiers f_1 and f_2; and size k_e of the erosion kernel.

1: Apply f_1 for every pixel (or overlapping window) of $\mathbf{T}_u$ to obtain a first forest segmentation mask $\boldsymbol{F}^1$.
2: **if** $k_e > 0$ **then**
3: $\quad \boldsymbol{F}^1 \leftarrow$ APPLYEROSION($\boldsymbol{F}^1$, k_e).
4: **end if**
5: Calculate the median forest vector $\hat{\boldsymbol{m}}_f$ by using all pixel spectral vectors of $\mathbf{T}_u$ belonging to forest regions as indicated by $\boldsymbol{F}^1$.
6: Subtract $\hat{\boldsymbol{m}}_f$ from all pixel spectral vectors of $\mathbf{T}_u$ to obtain an normalized image $\mathbf{T}_n$.
7: Apply f_2 for every pixel (or overlapping window) of $\mathbf{T}_n$ to obtain the final forest segmentation mask $\boldsymbol{F}^2$.
8: **return** $\boldsymbol{F}^2$.

5 EXPERIMENTS

This chapter will cover the experiments which evaluate the performance of selected CNC instances on four regions over the Amazon forest. It also will compare its results with those of other baselines. Information about the tested regions and the dataset creation is covered in Section 5.1. Details about the experiment's methodology such as classifiers setup and the chosen evaluation metrics are shown in Section 5.2. Finally, Section 5.3 show the results of each experiment.

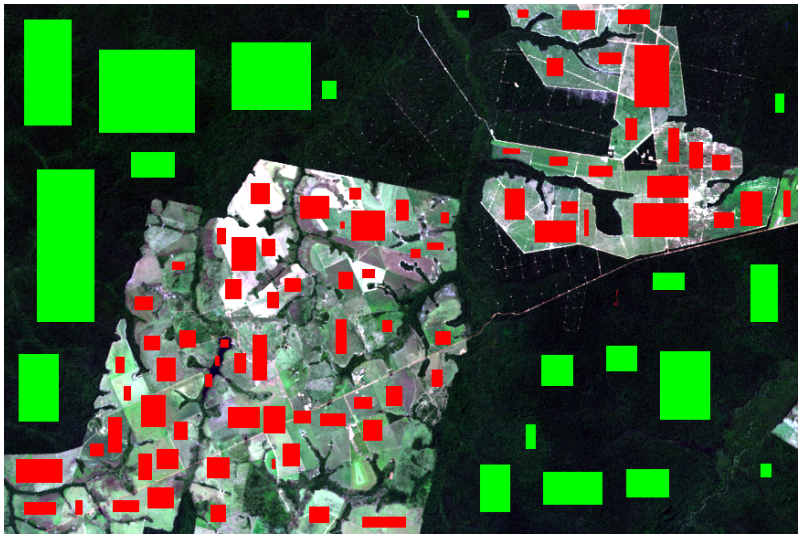
5.1 Study Area and Data

The data used in the experiments were 30 m resolution[1] top-of-atmosphere (TOA) and surface reflectance Landsat 8 OLI images (ROY et al., 2014) collected over the Brazilian Amazon Forest. The images were split into three disjoint sets, used for training, validation, and test as detailed in Table 5.1. Each set consists of four pairs of co-registered tiles extracted from distinct Landsat scenes (i.e. different Landsat path-rows). The location and spatial size of the test tiles were constrained to areas where existed very high-resolution images on Google Earth to facilitate ground truth annotation. Three of those four regions were also selected for dates that coincided with those of the Landsat images used in the PRODES product. All tiles were annotated non-exhaustively with forest/non-forest labels as in the example shown in Fig 5.1. The test tiles, in addition to having non-exhaustive forest/non-forest annotations, were also annotated exhaustively with deforestation/non-deforestation labels. An example is shown in Figure 5.4c, where white pixels correspond to deforested regions, and black pixels correspond to non-deforested areas. Therefore, the classifiers were trained and validated on the task of forest/non-forest classification and were tested both on the forest/non-forest discrimination task and on the deforestation detection objective through the post-classification comparison strategy. Eventual areas affected by clouds and its shadows were manually annotated and ignored during evaluation. The framework is tested using only one tile per Landsat scene due to the difficulty of annotating the deforestation ground truth exhaustively for the entire Landsat[2].

[1] Despite the input images having 30 m resolution, all generated forest loss masks were evaluated at 60 m resolution by ignoring deforestation blobs smaller than 60×60 meters (see Section 5.2.7).

[2] All tiles used in our experiments are smaller than 1000×1000 pixels. For comparison, a typical Landsat image has dimensions of about 7750×7750 pixels.

Figure 5.1: An example of a training tile with forest/non-forest annotations. The green rectangles (brighter in greyscale) are forest regions, the red ones (darker in greyscale) are non-forest areas, and the rest is not annotated and its pixels are not included in the forest/non-forest classification datasets.



There are two training datasets: a large and a small one. The large one consists of the annotated pixels from all training tiles, and the small one is a subset of the former with a subsampling factor of 16 (382,212 and 23,892 samples, respectively). There is only one validation dataset, which has size 68,948, which is a subset of all annotated pixels in the validation tiles with a subsampling factor of 8. The test dataset for the forest/non-forest classification task consists of all the 155,452 annotated pixels for this objective. On the other hand, the test deforestation detection dataset consists of all co-registered pairs of pixels from all test tiles. In an attempt to remove extreme values, all pixels were saturated in the 1st (p^b_{1st}) and 99th (p^b_{99th}) percentiles for each band b values from the training set. This clipping was done for all datasets using the same threshold values found in the training set.

The Landsat imagery was downloaded from the USGS repository both in radiance units (and then converted to TOA reflectance), and in surface reflectance[3]. Following Hansen et al. (2016) strategy, we used four bands from the Landsat 8 OLI sensor: red ($0.64 - 0.67$ μm), near infrared ($0.85 - 0.88$ μm), shortwave infrared 1 ($1.57 - 1.65$ μm), and shortwave infrared 2 ($2.11 - 2.29$ μm).

In addition to the spectral datasets (where each example consists of only the spectral information of a single pixel), we also created additional ones for spectral-spatial

[3]The App version of the surface reflectance images was "LaSRC_0.8.0". In addition to Vermote et al. (2016), more information about the surface reflectance Landsat 8 OLI product can be found in <https://landsat.usgs.gov/sites/default/files/documents/provisional_lasrc_product_guide_ee.pdf>.

analysis, where window patches of size 3×3, 5×5, and 9×9 centered in each of the selected pixels were extracted and then flattened in a vector.

Table 5.1: Training, validation, and test tiles details. The cities are the nearest ones to the tiles centers. Tile centers are given by the row and column in the parent Landsat image. Tile sizes are given in number of pixels. Tiles from different dates of the same scene correspond to identical regions on the ground, but their centers in pixel coordinates differ from each other because their parent Landsats are misaligned.

Phase	Path-Row	City	Date	Center	Size
Train	225-67	Santa Cruz do Xingu	2014-05-09	4663, 1168	872×583
Train	225-67	Santa Cruz do Xingu	2015-07-15	4663, 1238	872×583
Train	227-67	Alta Floresta	2013-05-20	4202, 1577	867×577
Train	227-67	Alta Floresta	2013-07-07	4422, 1507	867×577
Train	228-63	Itaituba	2013-06-12	4773, 4847	873×548
Train	228-63	Itaituba	2014-08-18	4993, 4867	873×548
Train	229-67	Nova Bandeirantes	2015-06-09	4357, 2721	865×585
Train	229-67	Nova Bandeirantes	2015-08-12	4367, 2651	865×585
Val	2-67	Rio Branco	2015-06-19	5722, 2918	876×584
Val	2-67	Rio Branco	2015-08-06	5722, 2848	876×584
Val	226-65	Novo Progresso	2013-06-14	1117, 4954	396×1198
Val	226-65	Novo Progresso	2015-07-22	1347, 5014	396×1198
Val	227-69	Lucas do Rio Verde	2013-05-04	1856, 4810	873×583
Val	227-69	Lucas do Rio Verde	2014-08-11	2076, 4920	873×583
Val	231-67	Theobroma	2014-10-10	3900, 1275	873×580
Val	231-67	Theobroma	2015-07-25	3900, 1295	873×580
Test	1-67	Placido de Castro	2013-07-08	4304, 2442	317×196
Test	1-67	Placido de Castro	2014-08-12	4304, 2462	317×196
Test	2-66	Porto Acre	2013-07-15	7349, 5260	274×655
Test	2-66	Porto Acre	2014-08-19	7349, 5290	274×655
Test	5-65	Cruzeiro do Sul	2013-09-22	5592, 1977	908×228
Test	5-65	Cruzeiro do Sul	2014-06-21	5592, 1987	908×228
Test	231-68	Seringueiras	2013-08-04	5107, 987	837×727
Test	231-68	Seringueiras	2015-06-23	5107, 987	837×727

5.2 Experimental Setup

5.2.1 Classification Methods

The following classifiers were tested: Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), LinearSVM, LinearSVM-SGD, RBF-SVM, K-Nearest Neighbors (KNN), Random Forest (RF), Multi-layer Perceptron (MLP), and Convolutional Neural Networks (CNN). Most classifiers were trained exclusively either on the small or the large training set. Only LDA and QDA were evaluated for both training set sizes. We will hereafter call the LDA classifier trained on the small dataset as LDAsmall and the big one as LDAbig, and likewise for QDA. We made two training sets because some classifiers take too long to train on the large dataset, such as LinearSVM, RBF-SVM, Random Forest, and KNN. LinearSVM-SGD is trained with Stochastic Gradient Descent (instead of using sequential minimal optimization (PLATT, 1998) to solve the dual prob-

lem), so it can be trained on the big training set very quickly. As the MLP and CNN have deep architectures, they require more data to learn successfully. Therefore, both are only trained on the large dataset. Details about the architecture of the neural network methods are provided separately in Section 5.2.5. All experiments were coded in Python 2.7 (ROSSUM, 1995), the NumPy array package (WALT; COLBERT; VAROQUAUX, 2011) and the scikit-image library (WALT et al., 2014) was used to manipulate the data, the Keras library (CHOLLET, 2015) with TensorFlow backend (ABADI et al., 2015) was used to define and run the neural networks methods, and the scikit-learn library (PEDREGOSA et al., 2011) was used to run the other classifiers.

All classifiers, except KNN and CNN, when using the TOA reflectance values as input, are evaluated for the window context sizes 1×1, 3×3, 5×5, and 9×9. The KNN run on input sizes of 1×1 and 3×3 because bigger contexts cause it to be very slow on inference time. The CNN is run just for input sizes 5×5 and 9×9 because its filter size is 3×3, so it only makes sense to be convolved on input sizes larger than that. Hereafter, we will refer to the method $ClassifierName$ with context size S×S as $ClassifierNameS$.

Every method trained on the small dataset (LinearSVM, RBF-SVM, Random Forest, KNN, LDAsmall, and QDAsmall) are alternatively also trained with gray-level co-occurrence matrix (GLCM) based features (HARALICK; SHANMUGAM; DINSTEIN, 1973) and a few additional statistics described in Section 5.2.2. Results for these methods will have a "-GLCM" appended to the name of the classifiers.

5.2.2 GLCM-Based Texture Features

The gray-level co-occurrence matrix was calculated for a single displacement vector (d, θ) ($d = 1$ and $\theta = 0$ in our experiments). Similar to the procedure in Dube and Mutanga (2015), the following features are extracted from the GLCM matrix: angular second moment, contrast, correlation, dissimilarity, entropy, homogeneity, standard deviation, and mean. Additionally, the mean, standard deviation, and data range (max-min) from the raw input window values are also calculated and appended to the GLCM descriptors. All features were extracted independently for each image band and then concatenated together. Before being fed to the GLCM algorithm, the sample values are first quantized into 64 levels between the global minimum and maximum in the training set. The GLCM matrix was calculated using the corresponding function from the scikit-image library (WALT et al., 2014).

5.2.3 Feature Standardization

The last preprocessing step for all classifiers (using GLCM descriptors or not) is to standardize the input features by subtracting it from the training mean and dividing by the training standard deviation. That means the data is mean centered and scaled to have unit variance[4]. This standardization is always done, even if the input images had already been normalized or radiometrically corrected because it facilitates the optimization of some of the used learning algorithms.

5.2.4 Hyperparameter Settings

Hyperparameters were set by optimizing the accuracy scores on the validation set. The values tested were:

- SVMs: the kernel width in the radial basis function (RBF) kernel was set to $1/n_c$, where n_c is the number of input features, and the cost parameter C was found by choosing among $\{10^i \mid -5 \leq i \leq 12\}$, the best parameter 10^{i^*} and then further refining it in a smaller range $\{j \times 10^{i^*} \mid 1 \leq j \leq 9\} \cup \{j \times 10^{i^*-1} \mid 2 \leq j \leq 9\}$.
- Random Forest: we used 500 estimators and the number of features (n_s) to be considered in each tree node split was searched in the neighborhood of $n_{mid} = \lfloor \sqrt{n_{in}} \rfloor$, where n_{in} is the input data dimensionality. In addition to n_{mid}, we also tested 10 equally spaced integers between 1 and $n_{mid} - 1$, and 5 equally spaced integers between $n_{mid} + 1$ and $\lfloor 1.5 \times n_{mid} \rfloor$.
- KNN: The number of neighbors K varied between 1 and 15.
- MLP and CNN: These were run for 15 epochs and the model from the epoch that achieved the best validation score was chosen.

The LDA and QDA do not have hyperparameters.

[4]Do not confuse this normalization with the forest median centering normalization used by the CNC framework. The first uses statistics from all pixels from all tiles in the training set. The later uses only statistics from the forest pixels independently for each tile.

5.2.5 Neural-Network Based Methods

The MLP architecture consisted of 4 fully-connected (also known as dense) layers with 400 neurons and a two-neuron dense layer with a softmax on top as output. The first 3 dense layers were followed by a Batch Normalization layer (IOFFE; SZEGEDY, 2015), a Rectified Linear Unit (NAIR; HINTON, 2010) activation function, and a Dropout layer (SRIVASTAVA et al., 2014) with $p = 0.5$. The forth dense layer was followed by the same ones as before, except the Dropout. We used mini-batch gradient descent with momentum to minimize the cross-entropy loss. The following training parameters were also used: mini-batch size (128), initial learning rate (0.01), and momentum (0.9). The learning rate was decayed by 10^{-6} after each update. The normalized initialization of Glorot and Bengio (2010) was employed.

The CNN architecture for input size 9×9 has three Conv blocks with eight filters each, followed by a max pooling layer with pool size 2×2, and then more three Conv blocks with 16 filters each followed by another 2×2 max pooling layer. The CNN for input size 5×5 is similar to the previous one, but it just goes up to the first max pooling layer. The Conv blocks consist of a convolutional layer with kernel size 3×3 (and padding to preserve spatial dimensions) followed by a Rectified Linear Unit (NAIR; HINTON, 2010) activation function. After the last max pooling layer, the activations are fully connected to two output neurons with a Softmax layer on top. We used mini-batch gradient descent with Nesterov momentum to minimize the cross-entropy loss, which was regularized with L_2 weight decay equal to 10^{-3}. The rest of the parameters are identical to the MLP ones.

5.2.6 Deforestation Detection With Post-Classification Comparison

Let us call the CNC pipeline, trained with classifiers f_1 and f_2, as f_{cnc}. Then, the deforestation detection with the post-classification comparison (PCC) strategy proceeds as follows. Given two co-registered unnormalized tiles $\mathbf{T}_u^1$ and $\mathbf{T}_u^2$ from two Landsat scenes acquired at consecutive times t_1 and t_2, we aim to estimate the final binary image $\hat{\boldsymbol{D}}$ indicating whether a forest loss happened in each pixel location. First we apply f_{cnc} to the two tiles and obtain the forest mask $\boldsymbol{F}^i = f_{cnc}(\mathbf{T}_u^i)$ for each tile $\mathbf{T}_u^i$. Then we build the final binary deforestation mask $\hat{\boldsymbol{D}}$ by setting to 1 all pixels that were marked as forest in $\boldsymbol{F}^1$ and as non-forest in $\boldsymbol{F}^2$, and 0 otherwise.

5.2.7 Accuracy Metrics

The validation and test forest/non-forest classification score is the harmonic mean between the $sensitivity = T_p/(T_p + F_n)$ and $specificity = T_n/(T_n + F_p)$, where T_p refers to the number of true positives, T_n the number of true negatives, F_p the number of false positives, and F_n the number of false negatives. This is computed for each tile, and then the minimum harmonic mean among all tiles is selected. In other words, we aim to maximize the harmonic mean for the worst performing tile.

Before computing the deforestation test scores, loss areas smaller the 2×2 (60m×60m) are removed from both the predicted and ground truth loss masks by applying a binary opening morphological operation with a structuring element of size 2×2. The objective is to reduce small false positive loss areas caused by slight misregistration between the two images (TOWNSHEND et al., 1992).

The overall deforestation test score is calculated by computing the geometric mean between the deforestation F1-Score ($2 \times (precision \times sensitivity)/(precision + sensitivity)$) of each tile, where $precision = TP/(TP+FP)$, $sensitivity = TP/(TP+FN)$, TP are the number of true positives, FP false positives, and FN the false negatives[5]. That is, the overall F1-Score is equal to $(\prod_{t=1}^{n_t} f1(t))^{\frac{1}{n_t}}$, where $f1(t)$ is the F1-Score for tile t and n_t is the number of test tiles. Two types of deforestation F1-Scores are calculated: one is **area** based, and the other is **alert** based. In the **area** based score, the $precision$ is the fraction of the number of predicted deforested **pixels** that were correctly classified and the $sensitivity$ is the fraction of the number of true deforestation **pixels** that were correctly classified. On the other hand, in the **alert** based score, the $precision$ is the fraction of the number of predicted deforested **polygons** which have at least 10% overlap with the true deforested areas and the $sensitivity$ is the fraction of the number of true deforested **polygons** overlapping at least 10% with the areas predicted by the method under evaluation. The alert based score is of interest for assessing deforestation alerts. In this case, detecting if a region suffered deforestation is prioritized over having the exact extent of the loss.

[5]Precision is also known in the remote sensing literature as the user's accuracy for the positive class and sensitivity is also know as the producer's accuracy for the positive class.

5.3 Results

5.3.1 Best CNC Instances for Deforestation Detection

The CNC framework does not specify which classifiers f_1 and f_2 one should use. In this section, we perform several experiments to find good choices for both classifiers concerning deforestation detection accuracy.

We begin by finding the best classifier f_2 among those listed in Section 5.2.1. As f_2 needs to be trained and tested on forest median centered TOA reflectance images, we first isolate the effect of f_1 by normalizing all tile images from the training, validation, and test set with the ground truth forest median. Then we train all classifiers on the training set, fine-tune its hyperparameters on the validation set, and evaluate the fine-tuned model's accuracy on the test set. The forest/non-forest classification scores for this experiment are shown in Table 5.2 and the deforestation detection results are presented in Table 5.3. By comparing Tables 5.2 and 5.3 we can observe that the bigger context helps in the forest/non-forest classification task but harms the deforestation detection results. The reason for this is discussed in Section 5.3.2.

The results shown in Table 5.3 indicate that the best classifier f_2 regarding the overall area and alert deforestation F1-Score was the LinearSVM1, with 68.02% and 74.75%, respectively. The first column of Table 5.4 shows the LinearSVM1 results for each tested region. On the other hand, the best f_2 classifiers, regarding forest/non-forest classification score, are the RBF-SVM and the MLP both with context size of 9 and with score of 99.91%, as shown in Table 5.2.

In the sequence, we search for the best classifier f_1 among the same candidates used previously for f_2, but now we train and test the classifiers on unnormalized TOA reflectance images. Here we aim to find the classifier that leads to the estimated forest medians closest to the ground truth ones. Experimentally, we noticed that by morphologically eroding the f_1 predicted forest masks we almost always obtained estimated medians closer to the true ones (Figure 5.2). The erosion kernels tested had a square shape with sides varying among 0 (meaning no erosion) and odd integers between 3 and 17 pixels wide. The Euclidean distance was used to measure the dissimilarity between the ground truth median and the estimated median. For each combination of classifiers and erosion kernels, we calculated the distance for each validation tile and selected the combination that minimized the distance for the tile that had the worst result.

Table 5.2: Overall test tiles forest/non-forest classification scores (defined in Section 5.2.7) for classifiers trained and tested on TOA reflectance images that were normalized using the ground truth forest medians.

Method/Context Size	1×1	3×3	5×5	9×9
LDAsmall	93.71%	93.84%	94.28%	94.2%
LDAbig	93.72%	93.78%	94.24%	94.69%
QDAsmall	95.41%	95.77%	97.64%	98.94%
QDAbig	95.42%	95.9%	97.6%	98.97%
LinearSVM	99.58%	99.79%	99.84%	99.82%
LinearSGD-SVM	99.48%	99.58%	99.56%	99.75%
RBF-SVM	99.28%	99.48%	99.73%	**99.91%**
RandomForest	98.49%	99.18%	99.24%	99.39%
KNN	98.77%	99.5%	-	-
MLP	99.63%	99.73%	99.78%	**99.91%**
CNN	-	-	99.69%	99.76%
LDAsmall-GLCM	-	97.64%	98.99%	98.46%
QDAsmall-GLCM	-	89.8%	65.04%	14.87%
LinearSVM-GLCM	-	98.82%	99.61%	99.43%
RBF-SVM-GLCM	-	99.0%	99.35%	99.41%
RandomForest-GLCM	-	97.96%	98.07%	96.42%
KNN-GLCM	-	98.39%	-	-

Figure 5.2 shows that the erosion operation improved the similarity between the predicted and true medians for almost all methods. The best combination was the RBF-SVM-GLCM with context size equal to 9 and erosion kernel size equal to 7. The worst tile for this combination had a distance of 0.003314. The best combination without GLCM features was the Random Forest with context size equal to 9 and erosion kernel size equal to 5, yielding a distance of 0.005461. The LDAsmall3 comes as the second best method without GLCM features with a distance of 0.007210.

Now let us define three CNC instances where all will have the best deforestation detection f_2 found previously (LinearSVM1), but the first classifier f_1 will vary between the three methods mentioned in the last paragraph. We will denote them by $f_{cnc}^{rbf\text{-}glcm9}$, f_{cnc}^{rf9}, and $f_{cnc}^{lda\text{-}small3}$, where the superscript refers to f_1's classifier name and its context size.

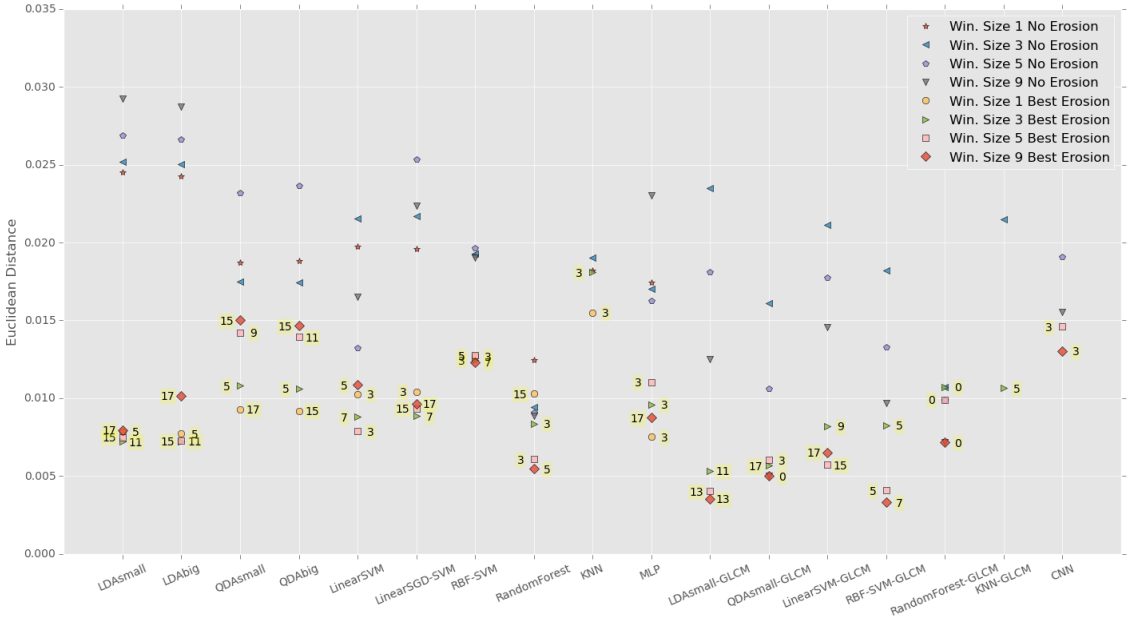
By running these three CNC instances on the TOA reflectance unnormalized datasets, we obtain the accuracies shown in columns 2-4 of Table 5.4, for both the forest/non-forest classification task and for the deforestation detection objective, which are just slightly lower than the respective ones obtained by centering the data to the true forest medians (see the first column of Table 5.4). This indicates that the classifier f_1 can in practice be used to estimate forest medians in unseen images. Individual results for each tested region and classifier are shown in the appendix.

The deforestation masks produced by the $f_{cnc}^{rbf\text{-}glcm9}$ can be seen on Figs. 5.4f, 5.5c, 5.6b,

Table 5.3: Overall test tiles deforestation detection F1-Scores for classifiers trained and tested on TOA reflectance images that were normalized using the ground truth forest medians. Each table cell shows F1-Scores in the format a/b, where 'a' is the area based score and 'b' is the alert based score as defined in Section 5.2.7. The best area and alert F1-Scores are highlighted in bold.

Method/Context Size	1×1	3×3	5×5	9×9
LDAsmall	27.68%/30.33%	23.44%/28.21%	16.82%/21.77%	9.72%/13.44%
LDAbig	27.51%/30.39%	23.68%/30.0%	17.38%/21.65%	10.57%/14.98%
QDAsmall	47.22%/49.49%	31.38%/29.99%	23.49%/26.72%	13.82%/16.72%
QDAbig	47.19%/49.65%	31.48%/30.34%	23.47%/26.05%	13.98%/15.97%
LinearSVM	**68.02%/74.75%**	53.73%/61.25%	41.14%/46.4%	28.09%/31.16%
LinearSGD-SVM	65.45%/72.95%	49.73%/58.4%	42.54%/49.88%	29.31%/33.22%
RBF-SVM	55.73%/61.38%	46.35%/50.63%	37.65%/44.29%	24.93%/27.64%
RandomForest	57.43%/60.87%	46.7%/50.54%	37.78%/42.57%	25.9%/29.65%
KNN	54.5%/58.91%	45.7%/47.53%	-	-
MLP	59.56%/63.45%	46.47%/55.59%	37.51%/44.1%	27.12%/32.64%
CNN	-	-	39.97%/48.75%	28.04%/32.62%
LDAsmall-GLCM	-	35.03%/33.48%	34.05%/33.13%	23.84%/22.1%
QDAsmall-GLCM	-	24.13%/22.27%	9.63%/11.25%	2.8%/4.87%
LinearSVM-GLCM	-	45.37%/48.05%	34.56%/37.6%	19.89%/22.12%
RBF-SVM-GLCM	-	44.92%/52.59%	30.42%/38.61%	20.13%/21.19%
RandomForest-GLCM	-	42.32%/47.03%	31.89%/35.77%	18.2%/20.81%
KNN-GLCM	-	42.51%/47.14%	-	-

Figure 5.2: Euclidean distance from the predicted forest median to the actual median for each classifier and window context size. Markers are shown for both the distance for the optimal erosion kernel size and for no erosion. Next to the best erosion markers the kernel size is also shown.



and 5.7b. To appreciate visually the main CNC steps, Figure 5.3 shows the forest segmentation masks for the Placido de Castro region using $f_{cnc}^{lda\text{-}small3}$ as an example.

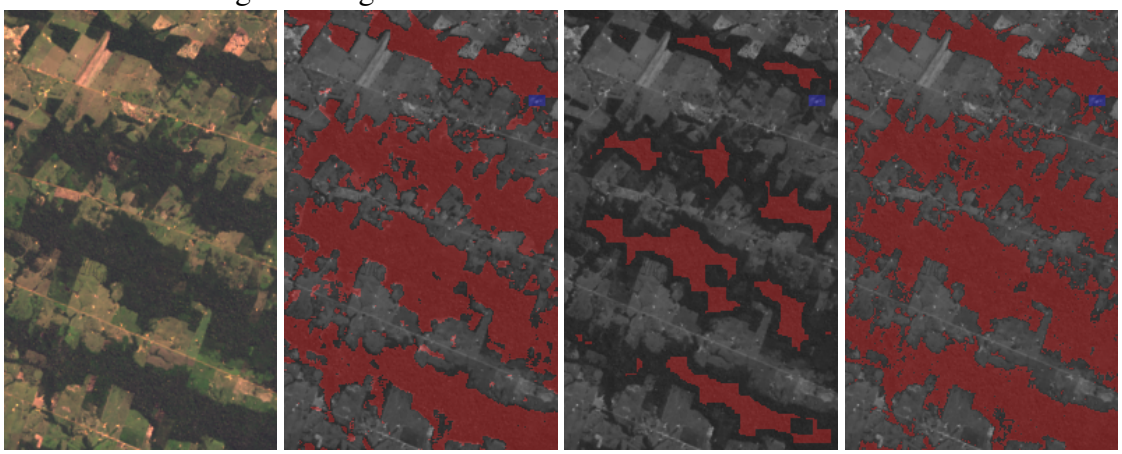
Table 5.4: Detailed results for selected methods. The first row (Overall Forest Classification Score) shows the overall scores for the forest/non-forest classification task as defined in Section 5.2.7. The remaining rows show deforestation detection scores in the format a/b, where 'a' is the area based score and 'b' is the alert based score. The last row (Overall Deforestation F1-Score) shows the geometric mean of the four tile's deforestation F1-Score. All values are percentages. Column 1: Show results for the classifier with best deforestation F1-Score (LinearSVM1) that was trained and tested on TOA reflectance images that were normalized using the ground truth forest medians. Columns 2-4: Show results for the three CNC instances defined in Section 5.3.1 applied to TOA reflectance images. Column 5: Show results for the classifier with best deforestation F1-Score (RBF-SVM1) that was trained and tested on TOA reflectance images (See Section 5.3.2). Column 6: Show results for the classifier with best deforestation F1-Score (LinearSGD-SVM1) that was trained and tested on surface reflectance images (See Section 5.3.3). The number following the classifier's name indicates the size of the used context.

Method		LinearSVM1	$f_{cnc}^{rbf\text{-}glcm9}$	f_{cnc}^{rf9}	$f_{cnc}^{lda\text{-}small3}$	RBF-SVM1	LinearSGD-SVM1
Input Normalization/Correction		GT Median	TOA	TOA	TOA	TOA	Surf. Reflec.
Overall Forest Class. Score		99.58	99.62	99.44	99.57	98.02	96.66
Cruzeiro do Sul	User's	76.76/75.61	76.04/73.81	75.95/73.81	75.49/72.09	67.27/91.30	72.07/96.15
	Producer's	66.49/83.78	66.89/83.78	67.43/83.78	67.43/83.78	35.00/54.05	47.43/67.57
	F1-Score	71.25/79.49	71.17/78.48	71.44/78.48	71.23/77.50	46.04/67.91	57.21/79.37
Placido de Castro	User's	76.16/73.13	75.16/72.46	74.08/69.01	75.22/72.06	41.23/30.11	58.42/49.21
	Producer's	59.13/79.31	59.71/79.31	59.32/79.31	59.32/79.31	64.13/81.03	59.32/84.48
	F1-Score	66.57/76.10	66.55/75.73	65.88/73.80	66.33/75.51	50.19/43.91	58.87/62.19
Porto Acre	User's	59.22/56.86	58.11/54.27	55.34/50.87	57.56/53.29	20.23/14.65	21.69/15.45
	Producer's	65.41/77.57	66.73/80.37	69.46/82.24	67.91/80.37	73.32/86.92	77.22/85.98
	F1-Score	62.16/65.62	62.12/64.79	61.60/62.86	62.31/64.09	31.72/25.07	33.87/26.20
Seringueiras	User's	89.71/92.58	89.39/91.56	89.64/92.47	89.69/92.80	91.74/95.39	92.47/95.19
	Producer's	60.94/68.40	61.18/69.70	61.22/70.56	60.92/70.13	49.57/60.17	50.76/58.44
	F1-Score	72.58/78.67	72.65/79.15	72.75/80.04	72.56/79.89	64.37/73.80	65.54/72.42
Overall Deforest. F1-Score		68.02/74.75	68.00/74.30	67.77/73.47	67.98/73.98	46.60/48.46	52.29/55.32

5.3.2 Single Step Methods on Unnormalized TOA Reflectance Images

Tables 5.5 and 5.6 show results for classifiers trained and tested on unnormalized TOA reflectance images. Note that using spatial context helps on the forest classification task, however, it harms deforestation detection accuracies. This happens because the forest/non-forest classification dataset pixels were taken from rectangular regions (see Figure 5.1 for an example) that often were not drawn very close to the forest boundaries (forest to non-forest transition contours). On the other hand, in the deforestation detection dataset, all pixels from the tiles were used, including those in or near the forest boundaries. We conjecture that contextual methods are not reliable when not all of its input context pixels belong to the same class and thus they do not work well near boundaries. As

Figure 5.3: Forest segmentation for each of the $f_{cnc}^{lda\text{-}small3}$ steps: (a) Placido de Castro region acquired 2014-08-12; (b–d) The regions in red represent the forest segmentation, and the blue rectangle is an ignored area due to a cloud.



(a) Tile RGB (b) f_1^* prediction (c) f_1^* 11×11 erosion (d) f_2^* prediction

the forest/non-forest dataset has few boundary pixels, the contextual methods performed better than the non-contextual ones, because the former can take into account contextual or textural cues. On the deforestation dataset, the non-contextual classifiers performed better because their precision on the boundaries overcompensates the lack of contextual information.

The classifier in Table 5.6 that obtained the best overall area deforestation F1-Score (46.6%) was the RBF-SVM1. We show more detailed evaluation metrics for this method in column 5 of Table 5.4. As can be seen in this table, any of the best CNC instances ($f_{cnc}^{rbf\text{-}glcm9}$, f_{cnc}^{rf9}, and $f_{cnc}^{lda\text{-}small3}$) outperform the RBF-SVM1 applied to unnormalized TOA reflectance images on all regions and metrics except in some of the deforestation user's and producer's accuracies[6]. However, it is important to note that it is easy to achieve a very high producer's accuracy in detriment of a very low user's accuracy and vice-versa. The real challenge is to keep both measures high, and this is what the F1-Score captures. It is also interesting to mention that even though the f_2 classifier from the best CNC instances was chosen to maximize the deforestation detection score, its test forest/non-forest classification accuracies were also higher than those of all the others methods in Table 5.5. Thus, these results support our claim in Chapter 1 that the final segmentation of the CNC framework is more accurate than the one of a single classifier applied to unnormalized images.

For illustrative purposes, we show the deforestation masks produced by the RBF-SVM1 in Figs. 5.4d, 5.6g, and 5.7g. Individual results for each tested region and classifier

[6]These terms are also know as precision and sensitivity.

applied to unnormalized TOA reflectance images are shown in the appendix.

Table 5.5: Overall test tiles forest/non-forest classification scores (defined in Section 5.2.7) for classifiers trained and tested on non-atmospherically-corrected TOA reflectance images.

Method/Context Size	1×1	3×3	5×5	9×9
LDAsmall	91.27%	93.25%	93.34%	90.82%
LDAbig	91.36%	93.37%	93.56%	91.55%
QDAsmall	89.12%	92.68%	96.26%	98.19%
QDAbig	89.37%	92.68%	96.26%	97.98%
LinearSVM	96.11%	95.94%	96.02%	98.96%
LinearSGD-SVM	96.09%	95.77%	91.84%	96.37%
RBF-SVM	98.02%	99.01%	99.10%	98.85%
RandomForest	92.43%	93.61%	94.19%	94.91%
KNN	96.72%	98.76%	-	-
MLP	96.87%	97.82%	98.26%	98.44%
CNN	-	-	99.17%	98.60%
LDAsmall-GLCM	-	94.63%	97.55%	**99.28%**
QDAsmall-GLCM	-	94.35%	92.47%	90.57%
LinearSVM-GLCM	-	95.90%	98.02%	98.93%
RBF-SVM-GLCM	-	94.16%	96.63%	98.55%
RandomForest-GLCM	-	94.92%	95.69%	95.38%
KNN-GLCM	-	95.38%	-	-

Table 5.6: Overall test tiles deforestation detection F1-Scores for classifiers trained and tested on non-atmospherically-corrected TOA reflectance images. Each table cell shows F1-Scores in the format a/b, where 'a' is the area based score and 'b' is the alert based score as defined in Section 5.2.7. The best area and alert F1-Scores are highlighted in bold.

Method/Context Size	1×1	3×3	5×5	9×9
LDAsmall	25.09%/32.19%	20.68%/28.71%	13.96%/19.8%	8.11%/9.4%
LDAbig	24.58%/31.67%	20.1%/27.86%	14.27%/21.16%	8.23%/12.28%
QDAsmall	34.69%/34.85%	26.25%/25.54%	21.13%/24.5%	13.35%/16.15%
QDAbig	34.8%/35.35%	26.34%/25.7%	20.99%/24.08%	13.55%/15.94%
LinearSVM	44.32%/48.72%	35.7%/41.67%	20.66%/21.01%	14.07%/17.01%
LinearSGD-SVM	44.39%/**48.89%**	34.8%/40.62%	22.11%/32.19%	18.16%/22.6%
RBF-SVM	**46.6%**/48.46%	40.27%/44.52%	32.66%/40.59%	20.09%/23.62%
RandomForest	37.96%/36.7%	28.69%/30.47%	22.9%/25.74%	15.35%/18.46%
KNN	45.62%/44.96%	36.34%/39.08%	-	-
MLP	43.75%/44.97%	34.42%/38.15%	26.55%/28.64%	18.69%/21.98%
CNN	-	-	30.69%/34.97%	14.21%/14.33%
LDAsmall-GLCM	-	30.31%/33.16%	32.0%/32.9%	23.78%/22.62%
QDAsmall-GLCM	-	25.42%/24.39%	15.22%/16.85%	6.32%/5.73%
LinearSVM-GLCM	-	37.37%/39.75%	30.47%/34.08%	18.95%/22.51%
RBF-SVM-GLCM	-	29.81%/29.73%	22.7%/26.41%	16.01%/19.0%
RandomForest-GLCM	-	30.95%/33.46%	24.91%/27.41%	13.39%/14.81%
KNN-GLCM	-	36.97%/40.93%	-	-

5.3.3 Single Step Methods on Surface Reflectance Images

In this section, we train and test all the classifiers configurations described in Section 5.2.1 on the surface reflectance images provided by USGS (VERMOTE et al., 2016). The results for all classifiers are shown in Tables 5.7 and 5.8. The same comments about contextual versus non-contextual methods said in the previous section are also true for this section's results. As can be seen by comparing the Tables 5.4, 5.7, and 5.8, any of the best CNC instances ($f_{cnc}^{rbf\text{-}glcm9}$, f_{cnc}^{rf9}, and $f_{cnc}^{lda\text{-}small3}$) outperforms all the other classifiers on forest/non-forest classification scores and deforestation F1-Scores. The only exception was in the Cruzeiro do Sul region where the **alert** F1-Score for the LinearSGD-SVM1 classifier (79.37%) was higher than the one from the best CNC instance (78.48%). However, in the same region, the CNC instance with worst **area** based F1-Score (71.17%) was better than the LinearSGD-SVM1 with 57.21% **area** F1-Score. The LinearSGD-SVM1 accuracies for each region can be seen in the last column of Table 5.4. Individual results for each tested region and classifier applied to surface reflectance images are shown in the appendix.

The deforestation masks produced by the best classifier in Table 5.8 (LinearSGD-SVM1) applied to surface reflectance test tiles can be seen on Figs. 5.4e, 5.5g, 5.6h, and 5.7h. We can see on Figure 5.7h that the surface reflectance tiles in the Porto Acre region, induced an erroneous elongated change region which is not present on the masks from the methods that were applied to TOA reflectance images. This mistake was due to an artifact introduced by the surface reflectance Landsat 8 algorithm in one of the tiles.

5.3.4 CNC versus PRODES

Here we compare the deforestation masks produced by the $f_{cnc}^{rbf\text{-}glcm9}$ CNC instance with those of the PRODES product (SHIMABUKURO et al., 2012). Differently from the experiments done in previous sections, the Seringueiras region is excluded from the comparisons because no PRODES masks correspond to its acquisition dates. Furthermore, here we ignore deforestation segments (polygons) with an area smaller than 64 pixels (5.76 ha), both on the ground truth and CNC predicted masks, to be consistent with the PRODES product which follows this procedure (MALDONADO et al., 2007).

Sometimes excluding a segment may be unfair with one or both of the methods being compared since, for instance, a polygon with 70 pixels from the PRODES mask

Table 5.7: Overall test tiles forest/non-forest classification scores (defined in Section 5.2.7) for classifiers trained and tested on the Landsat 8 OLI surface reflectance images provided by USGS.

Method/Context Size	1×1	3×3	5×5	9×9
LDAsmall	90.58%	92.53%	92.71%	90.25%
LDAbig	90.56%	92.55%	92.97%	91.13%
QDAsmall	84.26%	91.60%	96.06%	98.84%
QDAbig	84.64%	91.57%	96.06%	98.87%
LinearSVM	96.52%	98.21%	98.89%	97.33%
LinearSGD-SVM	96.66%	98.74%	99.06%	99.19%
RBF-SVM	97.50%	99.08%	**99.25%**	99.22%
RandomForest	90.09%	92.09%	91.86%	92.25%
KNN	96.31%	99.19%	-	-
MLP	96.90%	98.90%	99.18%	99.18%
CNN	-	-	98.76%	98.47%
LDAsmall-GLCM	-	95.30%	97.92%	99.17%
QDAsmall-GLCM	-	92.58%	89.38%	83.33%
LinearSVM-GLCM	-	98.08%	98.79%	99.05%
RBF-SVM-GLCM	-	96.99%	98.76%	98.79%
RandomForest-GLCM	-	94.62%	95.58%	95.58%
KNN-GLCM	-	96.89%	-	-

might overlap with two polygons of 35 pixels on the ground truth mask. So the last two segments would be excluded, and the PRODES mask accuracy would decrease. To avoid those situations, we give an advantage to the PRODES product by excluding all polygons on the ground truth smaller than 64 pixels, except those where its area overlaps at least 10% with one or more polygons on PRODES. We also do not exclude segments, from the CNC predicted mask, with at least 10% of its area overlapped with the previously excepted ground truth polygons. There could still be segments, smaller than 64 pixels, which were excluded from the CNC mask and overlapped with bigger polygons on the ground truth that were not excluded. This would also unfairly negatively affect the accuracy of the CNC method. So we also excluded objects from the ground truth that had an overlap of at least 10% with others on the CNC mask before performing the exclusions but no longer had after it with the exception of segments that would negatively affect the PRODES results if excluded. The inverse procedure is also done, that is, we excluded segments from the CNC mask that had an overlap of at least 10% with others on the ground truth mask before the exclusions, but no longer overlapped after it.

The results for the experiments are shown in Table 5.9 and the resulting masks are shown in the rightmost column of Figs. 5.5, 5.6, and 5.7, where the ground truth and CNC predicted masks with polygons smaller than 64 pixels removed are referred as “filtered”, and as “original” otherwise. As can be seen on the figures, many small segments were excluded in order to make a fair comparison with PRODES. Table 5.9 shows that the

Table 5.8: Overall test tiles deforestation detection F1-Scores for classifiers trained and tested on the Landsat 8 OLI surface reflectance images provided by USGS. Each table cell shows F1-Scores in the format a/b, where 'a' is the area based score and 'b' is the alert based score as defined in Section 5.2.7. The best area and alert F1-Scores are highlighted in bold.

Method/Context Size	1×1	3×3	5×5	9×9
LDAsmall	24.52%/32.89%	20.42%/28.32%	13.79%/21.33%	7.96%/8.81%
LDAbig	24.43%/32.37%	19.72%/27.15%	13.84%/21.96%	7.82%/10.8%
QDAsmall	35.01%/35.29%	26.36%/26.6%	20.99%/25.35%	13.09%/16.03%
QDAbig	35.02%/35.0%	26.44%/26.5%	21.08%/24.68%	13.24%/16.28%
LinearSVM	51.27%/54.75%	41.19%/44.99%	30.63%/35.19%	16.12%/15.97%
LinearSGD-SVM	**52.29%**/55.32%	42.63%/46.5%	32.03%/37.4%	18.52%/20.25%
RBF-SVM	51.18%/55.29%	44.29%/47.62%	35.52%/42.33%	21.74%/26.83%
RandomForest	41.9%/40.97%	32.42%/33.83%	26.76%/30.13%	17.79%/17.75%
KNN	49.64%/53.11%	40.86%/43.96%	-	-
MLP	51.17%/**55.81%**	42.0%/45.32%	30.71%/35.12%	19.13%/20.85%
CNN	-	-	30.07%/34.28%	17.11%/18.93%
LDAsmall-GLCM	-	30.36%/31.69%	29.88%/30.29%	23.24%/24.32%
QDAsmall-GLCM	-	25.39%/24.84%	13.93%/15.7%	5.39%/5.48%
LinearSVM-GLCM	-	41.14%/45.43%	31.42%/35.55%	18.63%/22.06%
RBF-SVM-GLCM	-	34.87%/36.27%	26.72%/29.72%	16.09%/18.72%
RandomForest-GLCM	-	32.73%/37.03%	25.12%/26.8%	14.02%/16.43%
KNN-GLCM	-	38.47%/43.29%	-	-

CNC method is more accurate, reaching a higher overall F1-Score, both in the area based (90.14% vs 33.49%) and alert based versions (94.10% vs 76.31%, see Section 5.2.7 for the scores definitions).

In the Cruzeiro do Sul region, both methods achieved a 100% alert score because both approaches detected the only deforestation spot in the ground truth mask (Figure 5.5) and no false positives. Regarding the area based score, the CNC method outperformed PRODES, achieving a F1-Score of 95.12% versus 49.52% for the PRODES product (Table 5.9).

In the Placido de Castro tile, there were two deforestation polygons in the ground truth mask, a small and a large one (Figure 5.6). The PRODES product detected the smaller one, while the CNC method detected both of them.

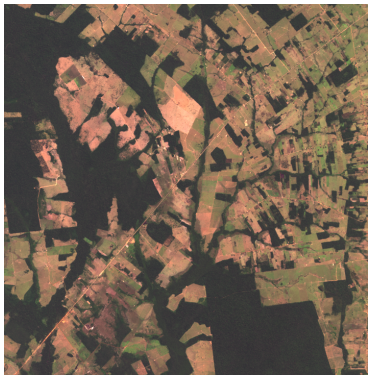
Figure 5.7 shows that both methods produced masks that were visually similar to the ground truth in the Porto Acre region. The PRODES product was better than the CNC regarding alert producer's accuracy but worse concerning alert user's accuracy. The CNC method was once again more accurate regarding the area based scores (Table 5.9).

We also ran this section's experiments for the f_{cnc}^{rf9} and $f_{cnc}^{lda\text{-}small3}$ CNC instances and the overall **alert** F1-Score for those were identical to the $f_{cnc}^{rbf\text{-}glcm9}$ one. On the other hand, the **area** F1-Scores for f_{cnc}^{rf9} (90.22%) and $f_{cnc}^{lda\text{-}small3}$ (90.18%) were slightly higher than the one of $f_{cnc}^{rbf\text{-}glcm9}$ (90.14%).

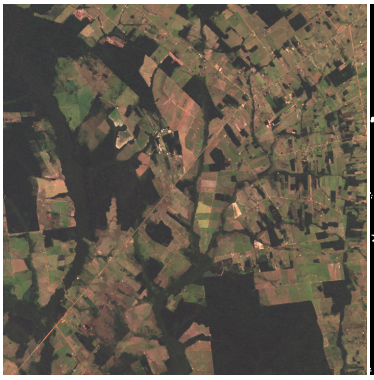
Table 5.9: Comparisons between the deforestation masks produced by $f_{cnc}^{rbf\text{-}glcm9}$ and PRODES. For the results in this table, polygons smaller than 64 pixels were ignored on the CNC masks and the ground truth masks, as described in Section 5.3.4. The deforested area is shown in number of pixels. The user's accuracy, producer's accuracy, and F1-Score results are shown in the format a/b, where 'a' is the area based score and 'b' is the alert based score as defined in Section 5.2.7.

Method		Ground Truth	$f_{cnc}^{rbf\text{-}glcm9}$	PRODES
Cruzeiro do Sul	Defo. Area	41	41	64
	# Polygons	1	1	1
	User's	-	95.12%/100.00%	40.62%/100.00%
	Producer's	-	95.12%/100.00%	63.41%/100.00%
	F1-Score	-	95.12%/100.00%	49.52%/100.00%
Placido de Castro	Defo. Area	263	286	32
	# Polygons	2	2	1
	User's	-	87.06%/100.00%	65.62%/100.00%
	Producer's	-	94.68%/100.00%	7.98%/50.00%
	F1-Score	-	90.71%/100.00%	14.24%/66.67%
Porto Acre	Defo. Area	650	503	985
	# Polygons	8	5	8
	User's	-	90.46%/100.00%	48.02%/62.50%
	Producer's	-	70.00%/62.50%	72.77%/100.00%
	F1-Score	-	78.92%/76.92%	57.86%/76.92%
Total Defo.	Area	954	830	1081
	# Polygons	11	8	10
Geometric Mean of F1-Scores		-	87.98%/91.63%	34.42%/80.04%

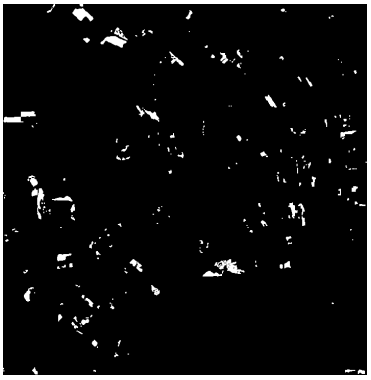
Figure 5.4: Deforestation masks for the Seringueiras region. The deforestation F1-Score, user's accuracy, and producer's accuracy are shown for each predicted mask. The scores on the left of '/' are area based and on the right are alert based (see Section 5.2.7 for the scores definitions). Here we do not show masks with polygons smaller than 64 pixels removed because the Seringueiras region is not compared with PRODES.



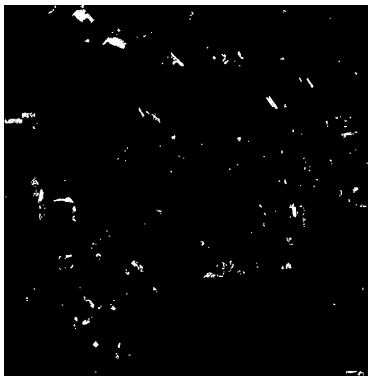
(a) RGB (2013-08-04)



(b) RGB (2015-06-23)



(c) Ground Truth Original

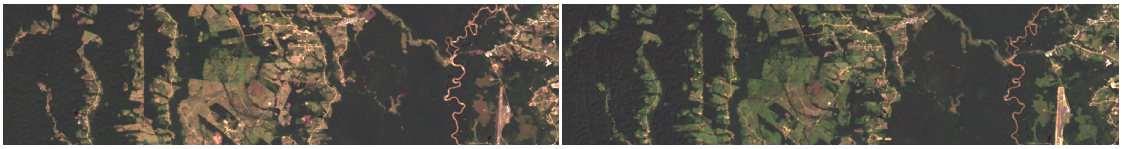


(d) TOA RBF-SVM1
F1-Score: 64.37/73.80
User: 91.74/95.39 | Prod.: 49.57/60.17

(e) Surface Ref. SGD-SVM1
F1-Score: 65.54/72.42
User: 92.47/95.19 | Prod.: 50.76/58.44

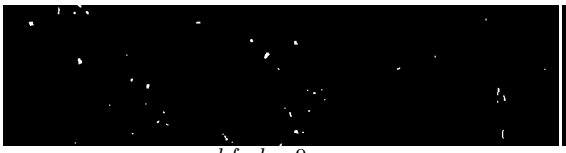
(f) $f_{cnc}^{rbf\text{-}glcm9}$ Original
F1-Score: 72.65/79.15
User: 89.39/91.56 | Prod.: 61.18/69.70

Figure 5.5: Deforestation masks for the Cruzeiro do Sul region. Masks for the PRODES comparisons are shown in the right column, where the polygons smaller than 64 pixels are removed. The left column shows masks without removing those small regions. The deforestation F1-Score, user's accuracy, and producer's accuracy are presented for each predicted mask. The scores on the left of '/' are area based and on the right are alert based (see Section 5.2.7 for the scores definitions).



(a) RGB (2013-09-22)

(b) RGB (2014-06-21)



(c) $f_{cnc}^{rbf\text{-}glcm9}$ Original
F1-Score: 71.17/78.48
User: 76.04/73.81 | Prod.: 66.89/83.78

(d) $f_{cnc}^{rbf\text{-}glcm9}$ Filtered
F1-Score: 95.12/100.00
User: 95.12/100.00 | Prod.: 95.12/100.00

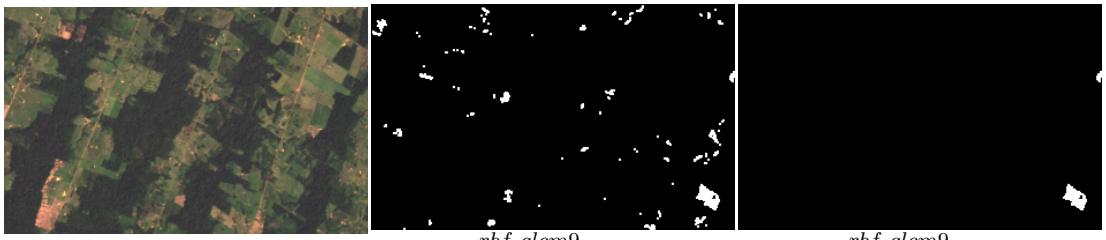
(e) Ground Truth Original

(f) Ground Truth Filtered

(g) Surface Ref. SGD-SVM1
F1-Score: 57.21/79.37
User: 72.07/96.15 | Prod.: 47.43/67.57

(h) PRODES
F1-Score: 49.52/100.00
User: 40.62/100.00 | Prod.: 63.41/100.00

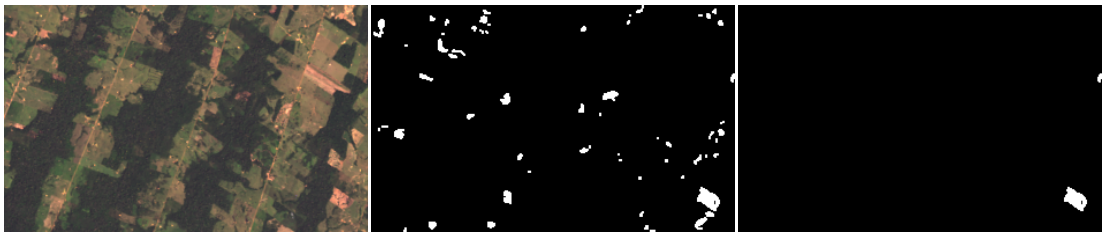
Figure 5.6: Deforestation masks for the Placido de Castro region. Masks for the PRODES comparisons are shown in the rightmost column, where the polygons smaller than 64 pixels are removed. The center and left columns show masks without removing those small regions. The deforestation F1-Score, user's accuracy, and producer's accuracy are presented for each predicted mask. The scores on the left of '/' are area based and on the right are alert based (see Section 5.2.7 for the scores definitions).



(a) RGB (2013-07-08)

(b) $f_{cnc}^{rbf\text{-}glcm9}$ Original
F1-Score: 66.55/75.73
User: 75.16/72.46 | Prod.: 59.71/79.31

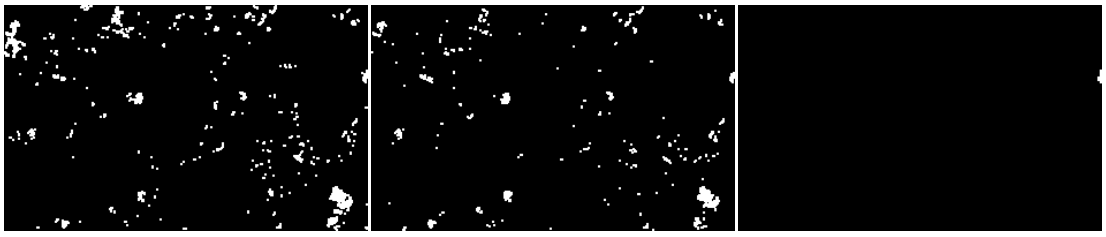
(c) $f_{cnc}^{rbf\text{-}glcm9}$ Filtered
F1-Score: 90.71/100.00
User: 87.06/100.00 | Prod.: 94.68/100.00



(d) RGB (2014-08-12)

(e) Ground Truth Original

(f) Ground Truth Filtered

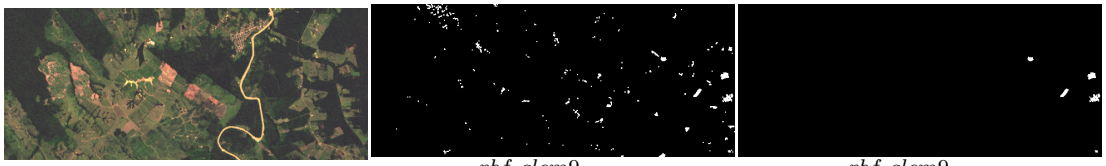


(g) TOA RBF-SVM1
F1-Score: 50.19/43.91
User: 41.23/30.11 | Prod.: 64.13/81.03

(h) Surface Ref. SGD-SVM1
F1-Score: 58.87/62.19
User: 58.42/49.21 | Prod.: 59.32/84.48

(i) PRODES
F1-Score: 14.24/66.67
User: 65.62/100.00 | Prod.: 7.98/50.00

Figure 5.7: Deforestation masks for the Porto Acre region. Masks for the PRODES comparisons are shown in the rightmost column, where the polygons smaller than 64 pixels are removed. The center and left columns show masks without removing those small regions. The deforestation F1-Score, user's accuracy, and producer's accuracy are presented for each predicted mask. The scores on the left of '/' are area based and on the right are alert based (see Section 5.2.7 for the scores definitions).



(a) RGB (2013-07-15)

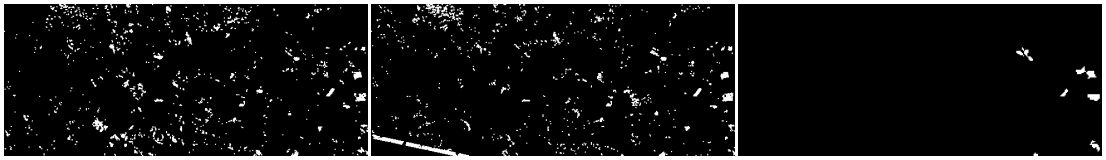
(b) $f_{cnc}^{rbf\text{-}glcm9}$ Original
F1-Score: 62.12/64.79
User: 58.11/54.27 | Prod.: 66.73/80.37

(c) $f_{cnc}^{rbf\text{-}glcm9}$ Filtered
F1-Score: 78.92/76.92
User: 90.46/100.00 | Prod.: 70.00/62.50

(d) RGB (2014-08-19)

(e) Ground Truth Original

(f) Ground Truth Filtered



(g) TOA RBF-SVM1
F1-Score: 31.72/25.07
User: 20.23/14.65 | Prod.: 73.32/86.92

(h) Surface Ref. SGD-SVM1
F1-Score: 33.87/26.20
User: 21.69/15.45 | Prod.: 77.22/85.98

(i) PRODES
F1-Score: 57.86/76.92
User: 48.02/62.50 | Prod.: 72.77/100.00

6 DISCUSSION AND CONCLUSION

6.1 Discussion of the Advantages and Limitations of the Proposed Approach

Our experiments suggest that the CNC framework applied to Landsat 8 TOA reflectance images is capable of:

1. Producing better tropical rainforest segmentation and deforestation maps than any of the tested classifiers applied to TOA reflectance images.
2. Generating superior tropical rainforest segmentation and deforestation maps than any of tested classifiers applied to surface reflectance images.
3. Yielding better deforestation detection maps than PRODES in the three studied regions.

Evidence for the superior CNC tropical rainforest segmentation quality is demonstrated directly through its forest/non-forest classification results and indirectly by its deforestation detection scores (see Table 5.4). For purposes of tropical rainforest classification, the normalization provided by the first two CNC steps (Classify-Normalize) proved to be a competitive alternative to the surface reflectance (SR) Landsat 8 OLI product (VERMOTE et al., 2016) provided by USGS.

It is also important to mention that even though the methods applied to Landsat 8 SR obtained inferior results compared to the CNC framework, the earlier, nonetheless, achieved higher overall deforestation F1-Score than the methods applied to unnormalized TOA reflectance images (see Table 5.6 and 5.8). This shows that the Landsat 8 SR algorithm also performs an image normalization that is useful for machine learning methods.

The experiments presented in Section 5.3.4, in addition to indicating the CNC prominence in deforestation detection, also showed that by ignoring polygons smaller than 64 pixels the CNC framework accuracy increased in all three analyzed regions, but the ordering of the scores between the regions remained the same. That is, the $f_{cnc}^{rbf\text{-}glcm9}$ CNC instance area/alert F1-Score went from 62.12%/64.79% (including small polygons) to 78.92%/76.92% (excluding small polygons) in the Porto Acre region, from 66.55%/75.73% to 90.71%/100.00% in the Placido de Castro tile, and from 71.17%/78.48% to 95.12%/100.00% in the Cruzeiro do Sul site. The regions were listed in ascending order of scores.

This work also showed that the erosion trick improves the similarity between the estimated forest median and the actual median for all classifiers, except the RandomForest-GLCM, as indicated by Fig 5.2. We defined and evaluated three instances of the CNC framework: $f_{cnc}^{rbf\text{-}glcm9}$, f_{cnc}^{rf9}, and $f_{cnc}^{lda\text{-}small3}$. We can observe in Table 5.4 that the scores are very similar for the three instances and that all of them perform better than the competing approaches, except in the Cruzeiro do Sul region where the LinearSGD-SVM1, trained and tested on surface reflectance images, has a slightly higher alert F1-Score than the best CNC instance (but lower area F1-Score). The fact of the three instances having similar results is important since the GLCM features used on the $f_{cnc}^{rbf\text{-}glcm9}$ CNC instance take very long to calculate compared to the running time of the other instances, which do not need to compute any descriptors. So we recommend using either the f_{cnc}^{rf9} or the $f_{cnc}^{lda\text{-}small3}$ instances since they are much faster than the $f_{cnc}^{rbf\text{-}glcm9}$ and also achieves good accuracies.

6.1.1 Analyzing Errors on the CNC Deforestation Detection Masks

Some of the errors on the forest loss masks are because we apply the opening operation to the predicted masks and the ground truth masks so that it deletes regions smaller than 2×2. So some small polygon may be removed from the ground truth mask, but its corresponding polygon in the predicted mask is not deleted because it was slightly bigger. The opposite situation can also happen, that is, a small polygon may be removed from the predicted mask, but its ground truth equivalent is not deleted. Even though applying the opening operation may sometimes cause those problems, we still employ it because otherwise the amount of false positive polygons, smaller than 2×2, would be very high.

Another source of errors we could see in the predicted masks are some false negatives that are produced when the forest is not completely removed from one date to another (clear-cut). This may happen due to selectively logging or wildfire. When this happens, the classifier will label the regions as forest for both dates, and thus the post-classification comparison approach will output that no change occurred in the area. A possible solution to remedy this type of errors is to use a regressor, instead of a classifier, as f_2 to try to estimate the forest coverage percentage for a given pixel for both dates. Then the procedure will yield a forest loss when the difference of forest coverage between the two dates is above a certain threshold. A second possible solution would be to stack both images

and use loss/non-loss labels directly as classification targets. The drawback of the first approach is that the ground truth regression real values will be more costly to obtain than hard labels. The downside of the second solution is that it will require a larger training dataset to keep the data density in the feature space constant since the dimensionality of the stacked training samples will double in size.

6.1.2 Processing Time

In this subsection, we discuss the processing time of the $f_{cnc}^{lda\text{-}small3}$ model which shows a good speed versus accuracy trade-off. The size of the trained model is 74.2 kB and it took 5 minutes to classify all pixels[1] into forest/non-forest labels of a Landsat of size 7741×7581 pixels[2] on an Intel Core i5-3337U 1.80GHz×4 processor laptop (using only one core) with 6 GB of RAM and running Ubuntu 14.04 LTS. All code was written in Python 2.7 (ROSSUM, 1995). By extrapolating the previous running time, it would take approximately 16 hours to segment forest regions in the whole Amazon Basin, which has an area of 6,200,000 km^2 (MARENGO et al., 2012), and about 16 days to segment the whole earth's land area (148,326,000 km^2) using the specified Laptop. It would take twice as much time to produce deforestation maps, as the model would need to be applied for two different periods.

6.2 Conclusion

In this work, we presented the Classify-Normalize-Classify framework for classifying images pixel-wise even when the distribution of the target classes shifts from one image to another. We showed experimentally that although a class of interest (COI) segmentation obtained by a classifier using unnormalized images can be unsatisfactory as a final product, the segmentation results can be useful to calculate a COI median signal with enough accuracy to perform an image normalization that facilitates the discrimination between the COI and non-COI classes.

The CNC method was evaluated on the task of deforestation detection using Landsat 8 OLI images over the Brazilian Amazon rainforest. To accomplish this task we used

[1] All pixels in the Landsat image were processed including those located inside the no-data regions (black pixels). We could make it faster by avoiding processing those regions.

[2] The valid region of the image corresponds to an area of 185 km×180 km.

the CNC framework (with the COI as forest) to segment a pair of co-registered images for two distinct dates, and then used the post-classification comparison strategy to obtain the predicted deforestation mask. The obtained results were compared with those achieved by classifiers trained and tested on the Landsat 8 surface reflectance images downloaded from USGS Earth Explorer, where the CNC approach performed better than other comparative schemes applied to surface reflectance images on all tested regions, regarding area F1-Score. The proposed framework generated deforestation masks were also compared with those of PRODES, and again the CNC results were more accurate in all evaluated regions.

6.3 Future Work

In domains where the input images can be very large such as in remote sensing, the classes distributions may shift within different parts of the same image. Therefore, dividing the input images into several tiles is recommended to reduce the internal dispersion of the classes, and then the CNC framework needs to be applied independently to each one. In future work, we intend to explore possible strategies to classify a whole Landsat. The simplest implementation of this would be to partition the entire image into equally sized non-overlapping tiles. More complex approaches could also be devised such as using half-overlapping tiles and then averaging the result of the first two steps (Classify-Normalize) in the intersected regions, or we could interpolate the estimated COI median signals from the center of each tile to the other positions. There is also several options for dealing with the scenario where the first classifier of the framework does not find any COI pixels (or find very few pixels such that the estimated median would likely be a bad estimate of the true median) in a given tile. One approach to solving that problem would be to interpolate between the median values of the nearest tiles with sufficient predicted COI pixels.

Another idea we would like to explore in the future is to use a regressor in the place of the second classifier in the CNC framework, and then try to estimate forest coverage fraction for each pixel instead of the binary categories forest/non-forest. We then would subtract the fraction for two different dates of a given pixel, and in this way, we could estimate how much coverage was lost in a given pixel between two dates. However, this exploration would require datasets with forest cover fraction ground truths.

REFERENCES

ABADI, M. et al. **TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems**. 2015. Available from Internet: <http://tensorflow.org/>. Accessed in: January 25, 2017.

BAO, N. et al. Comparison of relative radiometric normalization methods using pseudo-invariant features for change detection studies in rural and urban landscapes. **Journal of Applied Remote Sensing**, v. 6, n. 1, p. 063578, 2012.

BERK, A.; BERNSTEIN, L. S.; ROBERTSON, D. C. **MODTRAN: A moderate resolution model for LOWTRAN**. [S.l.], 1987. Available from Internet: <www.dtic.mil/docs/citations/ADA214337>. Accessed in: January 15, 2017.

BLASCHKE, T. Object based image analysis for remote sensing. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 65, n. 1, p. 2–16, 2010.

BOSER, B. E.; GUYON, I. M.; VAPNIK, V. N. A training algorithm for optimal margin classifiers. In: **Proceedings of the Fifth Annual Workshop on Computational Learning Theory**. New York, NY, USA: ACM, 1992. (COLT '92), p. 144–152.

BREIMAN, L. Random forests. **Machine Learning**, v. 45, n. 1, p. 5–32, 2001.

CANTY, M. J.; NIELSEN, A. A. Automatic radiometric normalization of multitemporal satellite imagery with the iteratively re-weighted mad transformation. **Remote Sensing of Environment**, v. 112, n. 3, p. 1025–1036, 2008.

CHOLLET, F. **Keras**. GitHub, 2015. Available from Internet: <https://github.com/fchollet/keras>. Accessed in: January 25, 2017.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine Learning**, v. 20, n. 3, p. 273–297, 1995.

DUBE, T.; MUTANGA, O. Investigating the robustness of the new landsat-8 operational land imager derived texture metrics in estimating plantation forest aboveground biomass in resource constrained areas. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 108, p. 12–32, 2015.

GLOROT, X.; BENGIO, Y. Understanding the difficulty of training deep feedforward neural networks. In: **AISTATS**. [S.l.: s.n.], 2010. v. 9, p. 249–256.

GRINAND, C. et al. Estimating deforestation in tropical humid and dry forests in madagascar from 2000 to 2010 using multi-date landsat satellite images and the random forests classifier. **Remote Sensing of Environment**, v. 139, p. 68–80, 2013.

HAM, J.; LEE, D. D.; SAUL, L. K. Semisupervised alignment of manifolds. In: **AISTATS**. [S.l.: s.n.], 2005. p. 120–127.

HANSEN, M. C. et al. Humid tropical forest disturbance alerts using Landsat data. **Environmental Research Letters**, v. 11, n. 3, p. 034008, 2016.

HANSEN, M. C.; LOVELAND, T. R. A review of large area monitoring of land cover change using landsat data. **Remote Sensing of Environment**, v. 122, p. 66–74, 2012. Landsat Legacy Special Issue.

HANSEN, M. C. et al. High-resolution global maps of 21st-century forest cover change. **Science**, v. 342, n. 2013, p. 850–3, 2013.

HANSEN, M. C. et al. A method for integrating MODIS and Landsat data for systematic monitoring of forest cover and change in the Congo Basin. **Remote Sensing of Environment**, v. 112, n. 5, p. 2495–2513, 2008.

HARALICK, R. M.; SHANMUGAM, K.; DINSTEIN, I. Textural features for image classification. **IEEE Transactions on Systems, Man, and Cybernetics**, SMC-3, n. 6, p. 610–621, 1973.

IOFFE, S.; SZEGEDY, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: **International Conference on Machine Learning**. [S.l.: s.n.], 2015. p. 448–456.

JR, P. S. C. An improved dark-object subtraction technique for atmospheric scattering correction of multispectral data. **Remote Sensing of Environment**, v. 24, n. 3, p. 459–479, 1988.

JR, P. S. C. Radiometric calibration of landsat thematic mapper multispectral images. **Photogrammetric Engineering & Remote Sensing**, v. 55, n. 9, p. 1285–1294, 1989.

JR, P. S. C. Image-based atmospheric corrections-revisited and improved. **Photogrammetric Engineering & Remote Sensing**, v. 62, n. 9, p. 1025–1036, 1996.

KIM, D.-H. et al. Global, landsat-based forest-cover change from 1990 to 2000. **Remote Sensing of Environment**, v. 155, p. 178–193, 2014.

KOTCHENOVA, S. Y. et al. Validation of a vector version of the 6s radiative transfer code for atmospheric correction of satellite data. part i: Path radiance. **Applied Optics**, v. 45, n. 26, p. 6762–6774, Sep 2006.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2012. p. 1097–1105.

LECUN, Y. et al. Gradient-based learning applied to document recognition. **Proceedings of the IEEE**, v. 86, n. 11, p. 2278–2324, Nov 1998.

LINNAINMAA, S. Taylor expansion of the accumulated rounding error. **BIT Numerical Mathematics**, v. 16, n. 2, p. 146–160, 1976.

LOH, W.-Y. Classification and regression trees. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery**, v. 1, n. 1, p. 14–23, 2011.

MALDONADO, M. J. D. L. R. et al. Modificações no método do prodes para estimar a mudança da cobertura florestal na bacia trinacional do rio acre na região de fronteira entre bolívia, brasil e peru na amazonia sul-ocidental. In: **Anais XIII Simpósio Brasileiro de Sensoriamento Remoto**. [S.l.: s.n.], 2007. p. 5903–5910.

MARENGO, J. A. et al. Extreme climatic events in the amazon basin. **Theoretical and Applied Climatology**, v. 107, n. 1, p. 73–85, 2012.

MATASCI, G. et al. Semisupervised transfer component analysis for domain adaptation in remote sensing image classification. **IEEE Transactions on Geoscience and Remote Sensing**, v. 53, n. 7, p. 3550–3564, 2015.

NAIR, V.; HINTON, G. E. Rectified linear units improve restricted boltzmann machines. In: **International Conference on Machine Learning**. [S.l.: s.n.], 2010. p. 807–814.

NAZEER, M.; NICHOL, J. E.; YUNG, Y.-K. Evaluation of atmospheric correction models and landsat surface reflectance product in an urban coastal environment. **International Journal of Remote Sensing**, v. 35, n. 16, p. 6271–6291, 2014.

NIELSEN, A. A.; CANTY, M. J. Kernel principal component and maximum autocorrelation factor analyses for change detection. In: **SPIE Europe Remote Sensing Conference**. Berlin, Germany: SPIE Europe, 2009. v. 7477.

PAN, S. J.; YANG, Q. A survey on transfer learning. **IEEE Transactions on Knowledge and Data Engineering**, v. 22, n. 10, p. 1345–1359, 2010.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

PLATT, J. **Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines**. [S.l.], 1998. 21 p. Available from Internet: <https://www.microsoft.com/en-us/research/publication/sequential-minimal-optimization-a-fast-algorithm-for-training-support-vector-machines/>. Accessed in: November 22, 2016.

POTAPOV, P. V. et al. Quantifying forest cover loss in Democratic Republic of the Congo, 2000-2010, with Landsat ETM+ data. **Remote Sensing of Environment**, v. 122, p. 106–116, 2012.

ROSSUM, G. **Python Reference Manual**. Amsterdam, The Netherlands, 1995. Available from Internet: <https://docs.python.org/2/reference/>. Accessed in: January 25, 2017.

ROY, D. P. et al. Landsat-8: Science and product vision for terrestrial global change research. **Remote Sensing of Environment**, v. 145, p. 154–172, 2014.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. In: ANDERSON, J. A.; ROSENFELD, E. (Ed.). **Neurocomputing: Foundations of Research**. Cambridge, MA, USA: MIT Press, 1988. p. 696–699.

SCHOTT, J. R.; SALVAGGIO, C.; VOLCHOK, W. J. Radiometric scene normalization using pseudoinvariant features. **Remote Sensing of Environment**, v. 26, n. 1, p. 1–16, 1988.

SCHOWENGERDT, R. A. **Remote sensing: models and methods for image processing**. Orlando, FL, USA: Academic press, 2006.

SHIMABUKURO, Y. E. et al. Using shade fraction image segmentation to evaluate deforestation in landsat thematic mapper images of the amazon region. **International Journal of Remote Sensing**, v. 19, n. 3, p. 535–541, 1998.

SHIMABUKURO, Y. E. et al. The brazilian amazon monitoring program: Prodes and deter projects. In: RUDORFF, B. F. T.; HANSEN, M. C. (Ed.). **Global forest monitoring from earth observation**. Boca Raton, FL: CRC Press, 2012. chp. 9, p. 153–170.

SINGH, A. Digital change detection techniques using remotely-sensed data. **International Journal of Remote Sensing**, v. 10, n. 6, p. 989–1003, 1989.

SONG, C. et al. Classification and change detection using landsat TM data: When and how to correct atmospheric effects? **Remote Sensing of Environment**, v. 75, n. 2, p. 230–244, 2001.

SRIVASTAVA, N. et al. Dropout: A simple way to prevent neural networks from overfitting. **Journal of Machine Learning Research**, v. 15, p. 1929–1958, 2014.

TARDY, B. et al. A software tool for atmospheric correction and surface temperature estimation of landsat infrared thermal data. **Remote Sensing**, v. 8, n. 9, p. 696, 2016.

TOWNSHEND, J. R. G. et al. The impact of misregistration on change detection. **IEEE Transactions on Geoscience and Remote Sensing**, v. 30, n. 5, p. 1054–1060, 1992.

TUIA, D.; CAMPS-VALLS, G. Kernel manifold alignment for domain adaptation. **PloS one**, v. 11, n. 2, p. 1–25, 2016.

TUIA, D. et al. Graph matching for adaptation in remote sensing. **IEEE Transactions on Geoscience and Remote Sensing**, v. 51, n. 1, p. 329–341, 2013.

TUIA, D.; PERSELLO, C.; BRUZZONE, L. Domain adaptation for the classification of remote sensing data: An overview of recent advances. **IEEE Transactions on Geoscience and Remote Sensing**, v. 4, n. 2, p. 41–57, 2016.

TUIA, D. et al. Semisupervised manifold alignment of multimodal remote sensing images. **IEEE Transactions on Geoscience and Remote Sensing**, v. 52, n. 12, p. 7708–7720, 2014.

VAPNIK, V.; LERNER, A. Pattern recognition using generalized portrait method. **Automation and Remote Control**, v. 24, p. 774–780, 1963.

VERMOTE, E. et al. Preliminary analysis of the performance of the landsat 8/oli land surface reflectance product. **Remote Sensing of Environment**, v. 185, p. 46–56, 2016.

VERMOTE, E. F. et al. Second simulation of the satellite signal in the solar spectrum, 6s: An overview. **IEEE Transactions on Geoscience and Remote Sensing**, v. 35, n. 3, p. 675–686, 1997.

VOLPI, M.; CAMPS-VALLS, G.; TUIA, D. Spectral alignment of multi-temporal cross-sensor images with automated kernel canonical correlation analysis. **ISPRS Journal of Photogrammetry and Remote Sensing**, v. 107, p. 50–63, 2015.

WALT, S. van der; COLBERT, S. C.; VAROQUAUX, G. The numpy array: A structure for efficient numerical computation. **Computing in Science & Engineering**, v. 13, n. 2, p. 22–30, 2011.

WALT, S. van der et al. scikit-image: image processing in Python. **PeerJ**, v. 2, p. e453, 2014.

WANG, Z.; JENSEN, J. R.; IM, J. An automatic region-based image segmentation algorithm for remote sensing applications. **Environmental Modelling & Software**, v. 25, n. 10, p. 1149–1165, 2010.

WERBOS, P. J. Applications of advances in nonlinear sensitivity analysis. In: **Proceedings of the 10th IFIP Conference**. New York, NY, USA: Springer Berlin Heidelberg, 1981. p. 762–770.

YANG, H. L.; CRAWFORD, M. M. Domain adaptation with preservation of manifold geometry for hyperspectral image classification. **IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing**, v. 9, n. 2, p. 543–555, 2016.

ZANOTTA, D. C. et al. An adaptive semisupervised approach to the detection of user-defined recurrent changes in image time series. **IEEE Transactions on Geoscience and Remote Sensing**, v. 53, n. 7, p. 3707–3719, 2015.

APPENDIX A — TABLES WITH DETAILED DEFORESTATION SCORES

In this appendix we show three tables that indicate the deforestation detection F1-Score, user's accuracy, and producer's accuracy for each of the four tested regions for classifiers applied to each of the following three types of image data: top-of-atmosphere (TOA) reflectance Landsat 8 OLI images normalized with the ground truth forest median vector (Table A.1), unnormalized TOA reflectance images (Table A.2), and surface reflectance images (Table A.3). Previously, we showed detailed scores for each region separately just for a few selected methods in Table 5.4, but here we will show for every tested classifier and input normalization/correction type.

Table A.1: Deforestation detection results of non-contextual classifiers trained and tested on TOA reflectance images that were normalized using the ground truth forest medians. For each column, the best score is highlighted in bold. The columns show scores in the format a/b, where 'a' is the area based score and 'b' is the alert based score as defined in Section 5.2.7. The first column (Overall F1-Score) shows the geometric mean of the four tile's deforestation F1-Score. All values are percentages.

Metric	Overall	Cruzeiro do Sul Defo. Metrics			Placido de Castro Defo. Metrics			Porto Acre Defo. Metrics			Seringueiras Defo. Metrics		
	F1-Score	User's	Producer's	F1-Score	User's	Producer's	F1-Score	User's	Producer's	F1-Score	User's	Producer's	F1-Score
LDAsmall1	27.68/30.33	9.59/9.23	55.14/75.68	16.34/16.45	43.44/29.59	38.63/51.72	40.90/37.65	15.56/16.96	17.65/31.78	16.54/22.11	63.61/58.31	45.60/65.80	53.12/61.83
LDAbig1	27.51/30.39	9.58/8.76	54.19/75.68	16.28/15.70	43.93/30.30	38.95/53.45	41.29/38.68	15.21/17.24	16.99/32.71	16.05/22.58	63.77/58.54	45.42/66.23	53.05/62.15
QDAsmall1	47.22/49.49	31.47/26.13	47.84/70.27	37.96/38.09	62.76/65.62	28.51/48.28	39.21/55.63	46.84/35.75	53.04/52.34	49.75/42.48	84.54/83.57	55.68/55.41	67.14/66.64
QDAbig1	47.19/49.65	31.18/26.61	47.70/70.27	37.71/38.60	62.94/66.67	28.51/48.28	39.24/56.00	47.17/35.59	52.85/51.40	49.84/42.06	84.75/84.21	55.71/55.41	67.23/66.84
LinearSVM1	**68.02/74.75**	**76.76/75.61**	**66.49/83.78**	**71.25/79.49**	**76.16**/73.13	59.13/**79.31**	**66.57**/76.10	**59.22/56.86**	65.41/77.57	**62.16/65.62**	89.71/92.58	**60.94**/68.40	**72.58**/78.67
LinearSGD-SVM1	65.45/72.95	64.63/69.77	64.46/81.08	64.55/75.00	76.09/**73.24**	57.08/**79.31**	65.23/**76.15**	59.02/55.77	61.13/74.77	60.06/63.89	90.03/92.07	60.78/67.10	72.57/77.63
RBF-SVM1	55.73/61.38	29.38/29.91	62.97/**83.78**	40.07/44.08	73.45/69.14	50.16/75.86	59.61/72.34	56.35/48.78	56.61/68.22	56.48/56.89	**90.38/94.64**	59.13/66.67	71.49/78.23
RandomForest1	57.43/60.87	56.52/48.44	55.68/75.68	56.09/59.07	73.54/70.97	40.23/62.07	52.01/66.22	53.12/43.21	56.09/62.62	54.57/51.13	88.68/89.00	55.55/55.84	68.31/68.63
KNN1	54.50/58.91	30.79/29.81	62.16/78.38	41.18/43.19	74.69/67.11	42.54/67.24	54.20/67.17	57.57/47.53	55.29/65.42	56.41/55.06	90.07/93.36	57.32/63.20	70.06/75.38
MLP1	59.56/63.45	37.81/34.94	63.92/78.38	47.51/48.33	73.95/66.23	55.09/77.59	63.14/71.46	58.33/51.90	58.02/70.09	58.17/59.64	89.83/91.06	60.19/**69.26**	72.08/**78.68**
LDAsmall3	23.44/28.21	5.64/6.77	45.41/67.57	10.04/12.31	41.23/39.06	38.57/44.83	39.85/41.75	13.31/16.06	17.46/31.78	15.10/21.34	58.63/55.52	43.50/60.17	49.94/57.75
LDAbig3	23.68/30.00	5.84/7.76	47.70/72.97	10.41/14.03	39.57/35.71	38.05/46.55	38.80/40.42	13.74/18.75	17.98/33.64	15.58/24.08	59.15/57.99	43.33/60.61	50.02/59.27
QDAsmall3	31.38/29.99	25.98/35.82	35.95/59.46	30.16/44.71	19.30/12.56	41.06/55.17	26.25/20.46	14.35/10.46	55.58/68.22	22.81/18.14	67.33/57.35	44.65/42.42	53.69/48.77
QDAbig3	31.48/30.34	26.21/36.92	35.81/59.46	30.27/45.56	19.31/12.73	41.00/53.45	26.26/20.56	14.49/10.69	55.62/69.16	22.99/18.51	67.17/57.64	44.80/42.42	53.75/48.87
LinearSVM3	53.73/61.25	50.65/62.79	52.97/72.97	51.78/67.50	59.13/60.66	46.25/56.90	51.91/58.72	38.51/42.86	61.46/72.90	47.35/53.98	80.88/87.26	54.98/52.81	65.46/65.80
LinearSGD-SVM3	49.73/58.40	48.05/60.00	48.38/72.97	48.22/65.85	58.17/64.29	39.21/53.45	46.84/58.37	33.96/37.50	57.60/64.49	42.73/47.42	78.12/87.50	53.34/50.22	63.39/63.81
RBF-SVM3	46.35/50.63	20.15/21.14	47.03/70.27	28.21/32.50	63.22/59.65	41.96/58.62	50.44/59.13	41.85/40.62	57.65/64.49	48.50/49.85	82.50/85.71	56.21/57.14	66.87/68.57
RandomForest3	46.70/50.54	33.85/34.21	44.32/67.57	38.39/45.42	61.15/62.26	34.08/46.55	43.77/53.27	36.47/32.95	56.28/60.75	44.26/42.73	79.57/87.50	53.48/49.35	63.97/63.11
KNN3	45.70/47.53	31.72/27.37	45.27/72.97	37.31/39.81	48.00/38.10	39.27/62.07	43.20/47.21	34.67/30.17	57.84/66.36	43.35/41.48	84.60/91.01	49.44/51.08	62.41/65.44
MLP3	46.47/55.59	22.10/33.73	47.84/72.97	30.23/46.14	64.78/63.83	40.29/51.72	49.68/57.14	40.91/45.95	58.16/71.96	48.04/56.08	82.02/86.59	53.31/51.52	64.62/64.60
LDAsmall5	16.82/21.77	3.33/5.21	31.08/48.65	6.01/9.42	33.02/26.15	29.47/34.48	31.14/29.75	9.30/14.59	11.53/20.56	10.30/17.07	53.63/50.18	33.85/44.16	41.50/46.98
LDAbig5	17.38/21.65	3.56/4.49	37.57/54.05	6.51/8.29	31.96/26.56	30.30/32.76	31.11/29.34	9.19/15.82	12.52/24.30	10.60/19.16	53.65/50.00	35.21/44.59	42.52/47.14
QDAsmall5	23.49/26.72	15.22/33.33	23.65/40.54	18.52/36.59	14.50/15.44	39.27/53.45	21.18/23.96	10.92/9.52	52.99/60.75	18.10/16.47	55.95/50.00	34.75/27.27	42.87/35.29
QDAbig5	23.47/26.05	15.10/31.91	23.38/40.54	18.35/35.71	14.56/13.97	39.46/51.72	21.27/22.00	10.91/9.58	53.08/58.88	18.10/16.48	55.76/49.67	34.89/27.71	42.92/35.57
LinearSVM5	41.14/46.40	38.80/63.64	37.70/56.76	38.25/60.00	38.58/31.08	39.91/48.28	39.23/37.82	25.73/33.09	53.51/62.62	34.75/43.30	69.79/67.13	45.28/36.36	54.93/47.17
LinearSGD-SVM5	42.54/49.88	33.41/54.76	41.08/62.16	36.85/58.23	41.46/35.62	41.19/56.90	41.32/43.81	27.29/32.92	55.39/64.49	36.56/43.59	73.87/75.34	48.83/44.16	58.80/55.68
RBF-SVM5	37.65/44.29	12.84/20.56	36.49/59.46	18.99/30.56	52.39/55.10	37.99/51.72	44.04/53.36	33.92/34.38	50.40/51.40	40.55/41.20	73.04/73.86	49.83/46.75	59.24/57.26
RandomForest5	37.78/42.57	25.67/34.43	32.43/59.46	28.66/43.61	44.32/43.10	29.72/44.83	35.58/43.95	29.26/28.99	49.32/50.47	36.73/36.82	67.85/72.88	45.43/34.20	54.42/46.55
MLP5	37.51/44.10	15.86/31.43	36.62/56.76	22.13/40.46	46.67/38.89	37.73/51.72	41.73/44.40	30.07/33.33	49.88/55.14	37.52/41.55	74.02/74.13	46.54/38.53	57.15/50.70
LDAsmall9	9.72/13.44	2.70/4.00	26.89/35.14	4.90/7.18	22.65/20.37	14.80/15.52	17.90/17.62	3.63/7.91	3.53/10.28	3.58/8.94	43.00/35.53	21.24/24.24	28.43/28.82
LDAbig9	10.57/14.98	2.56/4.12	34.05/43.24	4.77/7.52	22.32/15.79	18.96/24.14	20.51/19.09	3.41/9.88	5.18/14.95	4.11/11.90	41.12/32.64	25.02/26.84	31.11/29.46
QDAsmall9	13.82/16.72	6.03/10.81	10.14/16.22	7.56/12.97	9.26/15.85	29.66/39.66	14.11/22.65	6.95/8.08	42.92/45.79	11.97/13.74	41.56/30.00	21.79/14.29	28.59/19.35
QDAbig9	13.98/15.97	6.17/10.81	10.41/16.22	7.75/12.97	9.36/12.35	30.49/39.66	14.33/18.83	6.91/7.85	42.78/45.79	11.90/13.41	42.07/30.59	22.04/14.72	28.92/19.87
LinearSVM9	28.09/31.16	19.73/34.21	21.76/29.73	20.69/31.81	24.55/27.27	31.33/32.76	27.53/29.77	17.75/18.46	43.76/39.25	25.26/25.11	56.92/62.63	34.87/29.00	43.24/39.65
LinearSGD-SVM9	29.31/33.22	25.10/41.38	24.46/32.43	24.78/36.36	23.72/28.85	31.77/37.93	27.16/32.77	17.03/21.62	46.92/41.12	24.98/28.34	56.75/55.00	35.77/26.84	43.88/36.08
RBF-SVM9	24.93/27.64	5.82/9.32	22.16/32.43	9.22/14.48	30.59/36.36	30.75/31.03	30.67/33.49	23.71/25.25	41.46/30.84	30.17/27.77	55.82/59.17	38.15/34.20	45.33/43.34
RandomForest9	25.90/29.65	12.96/36.36	16.22/29.73	14.41/32.71	31.06/29.55	22.81/29.31	26.30/29.43	20.96/21.57	43.20/36.45	28.22/27.10	53.24/43.40	34.80/22.51	42.09/29.64
MLP9	27.12/32.64	10.15/29.55	23.92/32.43	14.26/30.92	27.48/33.96	32.35/34.48	29.71/34.22	21.45/19.57	42.26/40.19	28.45/26.32	58.99/64.22	36.21/29.87	44.87/40.77
LDAsmall-GLCM3	35.03/33.48	16.55/12.44	45.95/62.16	24.34/20.73	32.26/27.27	55.22/77.59	40.73/40.36	20.71/17.10	45.88/67.29	28.53/27.27	76.50/67.95	40.82/46.32	53.23/55.09
QDAsmall-GLCM3	24.13/22.27	9.60/11.21	37.43/59.46	15.29/18.87	14.86/13.49	39.01/60.34	21.52/22.05	11.46/6.65	53.65/64.49	18.88/12.05	62.39/49.24	48.55/48.92	54.60/49.08
LinearSVM-GLCM3	45.37/48.05	49.37/51.02	42.43/67.57	45.64/58.14	38.10/31.94	**62.33/79.31**	47.29/45.54	22.56/20.53	**68.42/86.92**	33.93/33.22	76.18/71.93	46.65/52.38	57.86/60.62
RBF-SVM-GLCM3	44.92/52.59	42.04/52.00	40.68/70.27	41.35/59.77	49.33/48.96	40.29/62.07	44.36/54.74	28.33/24.74	58.40/75.70	38.15/37.29	78.42/87.23	46.26/48.92	58.19/62.68
RandomForest-GLCM3	42.32/47.03	47.25/55.56	39.46/64.86	43.00/59.85	44.31/44.74	30.17/46.55	35.90/45.63	24.32/20.38	58.49/63.55	34.36/30.86	78.20/81.70	49.25/45.02	60.44/58.05
KNN-GLCM3	42.51/47.14	37.53/43.10	37.03/67.57	37.28/52.63	42.83/36.36	49.90/72.41	46.09/48.41	25.37/21.65	56.99/76.64	35.11/33.76	73.86/77.93	42.73/45.45	54.14/57.42
CNN5	39.97/48.75	16.86/35.21	39.05/64.86	23.55/45.64	47.91/42.03	42.66/58.62	45.14/48.96	31.41/33.11	54.92/58.88	39.97/42.39	74.77/80.92	50.23/47.19	60.09/59.61
LDAsmall-GLCM5	34.05/33.13	13.16/12.57	38.65/54.05	19.64/20.39	31.74/26.05	48.43/60.34	38.35/36.39	29.06/25.69	39.06/50.47	33.33/34.05	65.09/51.19	45.48/44.59	53.55/47.66
QDAsmall-GLCM5	9.62/11.25	2.40/3.91	18.51/37.84	4.24/7.10	3.81/5.70	19.92/31.03	6.40/9.63	4.33/3.91	41.69/41.12	7.84/7.14	44.22/40.32	37.12/27.71	40.36/32.84
LinearSVM-GLCM5	34.56/37.60	26.47/46.34	27.30/43.24	26.88/44.74	30.49/25.00	48.30/62.07	37.38/35.64	18.70/17.09	63.01/71.03	28.84/27.55	66.88/63.25	38.92/35.50	49.21/45.47
RBF-SVM-GLCM5	30.42/38.61	11.69/28.33	23.38/40.54	15.59/33.36	37.27/40.74	37.22/56.90	37.24/47.48	21.54/22.16	53.04/61.68	30.63/32.61	61.16/57.93	39.70/34.20	48.15/43.01
RandomForest-GLCM5	31.89/35.77	35.23/60.71	26.76/35.14	30.41/44.51	29.66/27.71	22.93/37.93	25.87/32.03	16.66/15.20	54.73/58.88	25.54/24.16	69.44/77.98	40.84/34.20	51.44/47.55
CNN9	28.04/32.62	13.83/33.33	24.05/29.73	17.56/31.43	29.31/30.19	29.21/36.21	29.26/32.93	19.49/19.69	40.38/41.12	26.29/26.62	59.40/65.71	37.22/29.87	45.77/41.07
LDAsmall-GLCM9	23.84/22.10	4.92/4.75	31.08/48.65	8.50/8.65	32.05/33.33	28.25/32.76	30.03/33.04	37.79/35.29	23.67/25.23	29.11/29.43	45.09/24.64	42.00/33.33	43.49/28.34
QDAsmall-GLCM9	2.80/4.87	0.25/0.37	5.54/10.81	0.48/0.72	0.89/7.50	5.45/6.90	1.53/7.19	1.86/3.61	27.25/20.56	3.48/6.15	26.68/27.88	21.97/12.99	24.10/17.72
LinearSVM-GLCM9	19.89/22.12	17.45/27.78	11.49/18.92	13.85/22.51	14.21/18.09	26.91/34.48	18.60/23.73	10.54/10.92	53.13/52.34	17.59/18.07	53.14/50.57	25.58/16.45	34.53/24.83
RBF-SVM-GLCM9	20.12/21.19	4.95/8.04	14.19/24.32	7.34/12.08	23.60/26.79	18.83/22.41	20.95/24.41	18.62/15.72	43.48/40.19	26.07/22.60	48.30/32.16	35.49/28.57	40.92/30.26
RandomForest-GLCM9	18.20/20.81	11.83/23.68	13.92/24.32	12.79/24.00	12.08/13.70	15.12/18.97	13.43/15.91	10.19/11.23	47.81/46.73	16.80/18.11	56.04/53.49	28.79/18.18	38.04/27.14

Table A.2: Deforestation detection results of classifiers trained and tested on non-atmospherically-corrected TOA reflectance images. For each column, the best score is highlighted in bold. The columns show scores in the format a/b, where 'a' is the area based score and 'b' is the alert based score as defined in Section 5.2.7. The first column (Overall F1-Score) shows the geometric mean of the four tile's deforestation F1-Score. All values are percentages.

Metric	Overall	Cruzeiro do Sul Defo. Metrics			Placido de Castro Defo. Metrics			Porto Acre Defo. Metrics			Seringueiras Defo. Metrics		
	F1-Score	User's	Producer's	F1-Score	User's	Producer's	F1-Score	User's	Producer's	F1-Score	User's	Producer's	F1-Score
LDAsmall1	25.09/32.19	19.21/27.63	27.70/51.35	22.69/35.93	30.26/22.46	51.51/74.14	38.12/34.48	5.93/8.62	23.01/41.12	9.44/14.25	78.58/72.53	35.15/52.38	48.57/60.83
LDAbig1	24.58/31.67	17.87/26.32	25.81/48.65	21.12/34.16	30.18/23.08	50.61/72.41	37.81/35.00	5.96/8.29	22.49/41.12	9.43/13.80	78.94/72.92	34.98/52.38	48.48/60.97
QDAsmall1	34.69/34.85	77.13/80.00	37.84/56.76	50.77/66.40	16.73/15.31	28.76/56.90	21.15/24.12	13.57/8.44	65.08/71.96	22.46/15.11	89.93/94.38	45.06/45.02	60.03/60.96
QDAbig1	34.80/35.35	78.03/82.76	37.43/56.76	50.59/67.33	16.97/15.60	27.48/56.90	20.98/24.49	14.02/8.70	64.71/71.96	23.05/15.53	89.90/94.41	44.94/45.02	59.93/60.97
LinearSVM1	44.32/48.72	70.08/92.31	37.03/56.76	48.45/70.29	41.17/32.53	64.83/84.48	50.36/46.97	15.72/13.33	65.04/85.98	25.31/23.09	89.61/91.74	47.96/**61.90**	62.48/**73.93**
LinearSGD-SVM1	44.39/**48.89**	69.92/92.31	36.76/56.76	48.18/70.29	**41.75/33.13**	64.38/84.48	**50.66/47.59**	15.91/13.57	64.99/85.98	25.56/23.44	89.60/91.27	47.69/60.61	62.25/72.84
RBF-SVM1	**46.60**/48.46	67.27/91.30	35.00/54.05	46.04/67.91	41.23/30.11	64.13/81.03	50.19/43.91	**20.23/14.65**	73.32/86.92	**31.72/25.07**	91.74/95.39	49.57/60.17	**64.37**/73.80
RandomForest1	37.96/36.70	77.71/76.67	36.76/54.05	49.91/63.40	22.72/17.51	38.95/63.79	28.70/27.48	13.79/9.09	73.69/85.05	23.23/16.43	90.91/94.86	47.55/47.62	62.44/63.41
KNN1	45.62/44.96	**82.85**/88.89	38.51/59.46	**52.58**/71.26	38.45/29.47	48.43/77.59	42.87/42.72	19.30/11.26	72.94/80.37	30.52/19.75	92.22/95.34	47.78/52.81	62.95/67.97
MLP1	43.74/44.96	67.40/89.29	**41.62/64.86**	51.46/**75.14**	33.47/26.34	46.70/75.86	38.99/39.10	17.79/11.53	73.22/83.18	28.63/20.25	**92.30/96.83**	48.69/53.25	63.75/68.71
LDAsmall3	20.68/28.71	14.43/22.00	17.30/29.73	15.73/25.29	22.45/19.23	51.38/72.41	31.24/30.39	5.01/9.45	22.82/42.99	8.22/15.49	75.89/79.26	32.22/44.59	45.23/57.07
LDAbig3	20.10/27.86	13.03/19.23	15.68/27.03	14.23/22.47	22.03/19.32	51.19/70.69	30.80/30.34	4.97/9.27	22.59/42.99	8.14/15.26	76.56/77.13	32.55/46.32	45.68/57.88
QDAsmall3	26.25/25.54	28.75/36.67	34.19/56.76	31.23/44.55	12.31/8.39	42.92/56.90	19.13/14.62	9.09/7.12	61.55/76.64	15.84/13.04	67.62/64.17	39.89/41.13	50.18/50.13
QDAbig3	26.34/25.70	28.93/38.60	34.05/56.76	31.28/45.95	12.37/8.00	43.31/56.90	19.24/14.03	9.10/7.35	61.36/75.70	15.85/13.40	67.83/64.36	40.18/41.56	50.47/50.51
LinearSVM3	35.70/41.67	60.47/88.89	28.11/43.24	38.38/58.18	27.91/24.40	**72.45/87.93**	40.29/38.21	10.08/10.80	64.61/85.05	17.44/19.16	84.86/88.77	46.69/58.87	60.24/70.80
LinearSGD-SVM3	34.80/40.62	59.05/88.24	26.89/40.54	36.95/55.56	27.24/23.26	71.36/**87.93**	39.43/36.78	9.84/10.69	62.68/83.18	17.01/18.95	83.24/87.30	45.92/58.87	59.19/70.32
RBF-SVM3	40.27/44.52	67.90/**94.74**	29.73/45.95	41.35/61.88	28.96/25.00	66.69/82.76	40.38/38.40	14.61/12.99	77.41/86.92	24.59/22.61	87.34/95.40	**50.54**/59.31	64.03/73.14
RandomForest3	28.69/30.47	71.17/85.71	26.35/43.24	38.46/57.49	12.24/11.33	23.45/41.38	16.09/17.79	11.53/8.90	63.48/69.16	19.52/15.77	83.14/92.00	42.35/37.66	56.11/53.45
KNN3	36.34/39.08	71.58/79.17	28.24/48.65	40.50/60.26	22.27/18.01	57.27/84.48	32.07/29.69	13.28/11.70	**80.71/91.59**	22.81/20.75	89.50/95.83	43.84/46.75	58.85/62.85
MLP3	34.42/38.15	72.52/86.36	30.68/48.65	43.11/62.24	18.16/17.35	46.00/68.97	26.04/27.73	12.16/11.67	76.33/88.79	20.98/20.62	86.57/93.08	45.45/43.72	59.61/59.50
LDAsmall5	13.96/19.80	8.30/18.75	7.43/13.51	7.84/15.71	16.28/11.00	42.99/53.45	23.62/18.25	3.56/8.00	16.89/38.32	5.89/13.24	70.35/69.74	23.17/28.57	34.86/40.54
LDAbig5	14.27/21.16	8.07/20.69	8.24/16.22	8.16/18.18	15.35/11.29	43.56/58.62	22.70/18.93	3.63/8.21	19.11/41.12	6.10/13.68	70.66/71.43	24.84/30.30	36.76/42.55
QDAsmall5	21.13/24.50	15.09/31.91	22.70/40.54	18.13/35.71	11.93/12.20	41.26/56.90	18.51/20.09	8.61/8.03	55.91/61.68	14.92/14.21	54.26/51.66	31.42/26.84	39.79/35.33
QDAbig5	20.98/24.08	14.91/32.61	22.30/40.54	17.87/36.14	12.00/11.31	41.38/56.90	18.61/18.87	8.43/7.86	55.53/60.75	14.64/13.91	54.05/50.64	31.55/27.27	39.84/35.45
LinearSVM5	20.66/21.00	44.49/62.50	13.65/18.92	20.89/29.05	9.13/8.78	28.89/44.83	13.87/14.68	8.07/6.90	59.76/64.49	14.22/12.47	73.87/79.00	31.51/23.81	44.18/36.59
LinearSGD-SVM5	22.11/32.19	39.68/75.00	13.51/21.62	20.16/33.57	18.33/25.53	59.19/79.31	28.00/38.63	5.15/8.41	40.80/61.68	9.15/14.80	65.59/74.03	35.76/45.02	46.29/55.99
RBF-SVM5	32.66/40.59	59.92/94.44	20.41/40.54	30.44/56.73	22.38/23.33	63.42/74.14	33.08/35.50	11.62/13.85	73.22/84.11	20.05/23.78	80.46/83.59	43.32/42.86	56.32/56.66
RandomForest5	22.90/25.74	56.22/76.47	17.70/24.32	26.93/36.91	8.93/10.90	20.82/32.76	12.50/16.35	9.43/9.77	59.11/65.42	16.27/17.00	75.00/85.26	37.75/28.57	50.22/42.80
MLP5	26.55/28.64	57.49/93.33	19.19/29.73	28.77/45.10	13.53/10.75	40.42/58.62	20.28/18.17	9.79/10.53	66.87/78.50	17.08/18.56	77.13/81.90	36.84/30.30	49.86/44.24
LDAsmall9	8.11/9.40	8.30/10.53	2.57/2.70	3.92/4.30	10.06/10.24	24.86/37.93	14.33/16.13	2.45/4.62	10.68/28.04	3.99/7.94	56.15/50.43	11.68/8.23	19.34/14.14
LDAbig9	8.23/12.28	5.84/15.00	2.43/5.41	3.44/7.95	9.48/11.46	31.97/51.72	14.62/18.77	2.22/4.04	14.26/32.71	3.85/7.20	58.98/57.55	14.89/12.99	23.78/21.19
QDAsmall9	13.35/16.14	5.95/11.11	9.86/16.22	7.43/13.19	8.98/13.25	31.07/41.38	13.94/20.08	6.43/7.94	44.14/45.79	11.23/13.54	40.63/32.26	20.58/13.42	27.32/18.95
QDAbig9	13.55/15.94	6.10/10.81	10.14/16.22	7.62/12.97	9.16/12.99	31.84/41.38	14.23/19.77	6.43/7.58	44.24/46.73	11.22/13.05	41.13/31.82	20.90/13.85	27.71/19.30
LinearSVM9	14.07/17.01	32.47/40.00	3.38/5.41	6.12/9.52	9.79/12.00	37.86/43.10	15.56/18.77	6.13/9.52	56.09/62.62	11.05/16.53	67.43/64.29	25.76/18.18	37.28/28.35
LinearSGD-SVM9	18.16/22.60	37.06/33.33	7.16/8.11	12.00/13.04	13.85/20.48	62.08/72.41	22.65/31.93	5.50/9.94	58.21/69.16	10.05/17.38	71.45/78.38	27.57/23.38	39.79/36.01
RBF-SVM9	20.09/23.62	35.94/50.00	6.22/5.41	10.60/9.76	16.15/24.10	55.22/63.79	24.99/34.98	8.17/14.36	62.07/65.42	14.44/23.55	65.11/72.62	31.64/26.41	42.58/38.73
RandomForest9	15.35/18.46	33.74/54.55	7.43/13.51	12.18/21.66	5.92/8.91	17.10/24.14	8.80/13.02	7.48/7.95	53.04/46.73	13.10/13.59	59.12/63.38	29.70/19.91	39.54/30.31
MLP9	18.69/21.98	37.72/33.33	8.51/8.11	13.89/13.04	12.08/18.09	58.04/63.79	20.00/28.18	6.27/11.17	66.92/71.03	11.47/19.30	69.22/78.75	26.46/20.78	38.28/32.88
LDAsmall-GLCM3	30.31/33.16	31.13/37.04	26.76/43.24	28.78/39.90	24.93/19.35	59.06/81.03	35.06/31.25	10.80/10.72	49.04/71.96	17.70/18.66	75.47/71.88	34.43/40.69	47.29/51.96
QDAsmall-GLCM3	25.42/24.39	10.14/11.68	36.62/56.76	15.88/19.37	19.56/15.42	42.92/63.79	26.87/24.83	11.61/8.24	55.67/69.16	19.22/14.73	61.05/53.05	43.66/47.19	50.91/49.95
LinearSVM-GLCM3	37.37/39.75	59.56/81.25	36.22/62.16	45.04/70.44	26.19/19.73	68.74/82.76	37.93/31.86	12.76/10.92	73.98/90.65	21.77/19.50	75.66/71.05	40.14/47.62	52.45/57.02
RBF-SVM-GLCM3	29.81/29.73	8.95/10.50	35.27/**64.86**	14.28/18.08	32.35/24.85	61.76/72.41	42.46/37.00	15.53/12.43	61.88/78.50	24.83/21.46	62.95/55.66	44.96/53.25	52.45/54.43
RandomForest-GLCM3	30.95/33.45	58.21/85.19	30.68/51.35	40.18/64.08	13.87/14.22	31.13/51.72	19.19/22.30	12.49/9.23	66.21/76.64	21.02/16.48	80.53/90.44	43.64/37.66	56.60/53.18
KNN-GLCM3	36.97/40.93	43.64/59.09	35.68/**64.86**	39.26/61.84	27.27/23.33	65.21/74.14	38.46/35.50	14.31/12.65	70.16/88.79	23.77/22.14	70.57/71.37	41.25/48.48	52.07/57.74
CNN5	30.69/34.97	61.16/81.25	20.00/32.43	30.14/46.36	20.28/18.87	55.86/72.41	29.76/29.94	10.16/11.61	75.67/87.85	17.91/20.51	80.95/86.67	41.93/37.66	55.25/52.51
LDAsmall-GLCM5	32.00/32.90	47.21/60.00	21.76/29.73	29.79/39.76	23.76/18.50	48.30/65.52	31.85/28.85	14.25/13.04	53.88/71.03	22.54/22.04	69.17/65.31	38.01/35.93	49.06/46.36
QDAsmall-GLCM5	15.22/16.85	2.90/5.83	17.70/37.84	4.98/10.10	13.64/12.16	25.30/39.66	17.72/18.62	9.61/7.16	42.54/42.99	15.68/12.28	44.28/46.07	34.50/28.14	38.78/34.94
LinearSVM-GLCM5	30.47/34.08	34.81/61.54	23.38/37.84	27.97/46.86	21.84/20.61	56.63/72.41	31.53/32.08	12.61/11.67	71.67/80.37	21.44/20.38	67.36/68.39	34.47/32.47	45.61/44.03
RBF-SVM-GLCM5	22.70/26.41	3.84/7.87	22.03/40.54	6.54/13.18	30.02/29.63	45.04/58.62	36.02/39.36	16.84/13.86	55.29/62.62	25.82/22.70	50.30/52.15	38.63/34.20	43.70/41.31
RandomForest-GLCM5	24.91/27.41	47.94/81.25	20.41/29.73	28.63/43.53	10.57/11.92	25.69/39.66	14.98/18.33	10.79/9.04	60.24/61.68	18.29/15.77	72.87/82.80	36.98/30.74	49.06/44.83
CNN9	14.21/14.32	8.47/7.41	5.00/5.41	6.29/6.25	9.06/12.12	35.04/48.28	14.39/19.38	6.48/7.24	58.82/60.75	11.68/12.94	57.20/51.52	29.08/18.18	38.56/26.88
LDAsmall-GLCM9	23.78/22.62	33.01/22.73	9.32/13.51	14.54/16.95	16.76/16.83	25.88/32.76	20.34/22.24	15.84/12.62	46.02/46.73	23.56/19.87	57.96/46.21	38.01/28.14	45.91/34.98
QDAsmall-GLCM9	6.32/5.73	0.60/0.59	5.54/10.81	1.08/1.12	5.17/3.90	6.92/10.34	5.92/5.66	6.28/6.19	27.29/20.56	10.21/9.51	28.26/29.17	21.56/12.99	24.46/17.97
LinearSVM-GLCM9	18.95/22.51	21.01/42.86	10.14/18.92	13.67/26.25	12.80/16.30	32.29/36.21	18.34/22.48	8.93/9.87	58.64/56.07	15.49/16.78	53.96/61.25	24.00/16.45	33.23/25.93
RBF-SVM-GLCM9	16.01/19.00	3.80/11.11	9.46/18.92	5.42/14.00	16.51/14.44	23.57/25.86	19.42/18.54	12.10/13.61	45.08/43.93	19.08/20.78	42.37/40.00	26.64/17.32	32.71/24.17
RandomForest-GLCM9	13.39/14.81	16.24/30.00	7.70/13.51	10.45/18.63	4.57/5.38	13.00/17.24	6.76/8.20	7.39/7.14	47.25/45.79	12.79/12.36	54.75/52.00	26.35/16.88	35.58/25.49

Table A.3: Deforestation detection results of classifiers trained and tested on the Landsat 8 OLI surface reflectance images provided by USGS. For each column, the best score is highlighted in bold. The columns show scores in the format a/b, where 'a' is the area based score and 'b' is the alert based score as defined in Section 5.2.7. The first column (Overall F1-Score) shows the geometric mean of the four tile's deforestation F1-Score. All values are percentages.

Metric	Overall	Cruzeiro do Sul Defo. Metrics			Placido de Castro Defo. Metrics			Porto Acre Defo. Metrics			Seringueiras Defo. Metrics		
	F1-Score	User's	Producer's	F1-Score	User's	Producer's	F1-Score	User's	Producer's	F1-Score	User's	Producer's	F1-Score
LDAsmall1	24.52/32.89	18.61/26.32	26.76/45.95	21.95/33.46	28.97/24.74	53.04/77.59	37.47/37.52	5.77/9.44	22.31/42.06	9.16/15.42	75.69/71.33	35.07/52.38	47.94/60.40
LDAbig1	24.43/32.37	18.28/26.32	26.35/45.95	21.58/33.46	29.17/22.84	52.40/75.86	37.48/35.11	5.80/9.35	22.16/42.06	9.20/15.29	76.25/71.73	34.93/53.25	47.91/61.12
QDAsmall1	35.01/35.29	77.75/76.92	38.24/54.05	51.27/63.49	26.24/22.16	24.34/44.83	25.26/29.66	10.60/7.02	63.81/71.96	18.18/12.79	89.11/92.70	49.66/49.35	63.78/64.41
QDAbig1	35.02/35.00	77.96/76.92	38.24/54.05	51.31/63.49	25.78/21.31	23.89/44.83	24.80/28.89	10.91/6.97	63.58/70.09	18.62/12.67	89.19/93.30	49.26/49.35	63.47/64.55
LinearSVM1	51.27/54.75	69.29/89.29	45.14/64.86	54.66/75.14	**59.49/50.00**	52.59/81.03	55.83/61.84	22.43/15.81	72.33/81.31	34.24/26.47	92.38/96.23	51.49/58.87	66.12/73.05
LinearSGD-SVM1	**52.29**/55.32	72.07/96.15	**47.43/67.57**	**57.21/79.37**	58.42/49.21	59.32/**84.48**	**58.87/62.19**	21.69/15.45	**77.22**/85.98	33.87/26.20	**92.47**/95.19	50.76/58.44	65.54/72.42
RBF-SVM1	51.18/55.28	72.77/85.71	41.89/62.16	53.17/72.06	55.52/44.80	51.51/82.76	53.44/58.13	24.45/18.20	71.72/80.37	36.47/29.68	92.02/**97.69**	**51.69/61.04**	**66.19/75.13**
RandomForest1	41.89/40.97	70.88/61.11	44.73/59.46	54.85/60.27	43.58/33.11	42.86/65.52	43.22/43.99	12.07/8.91	68.94/82.24	20.54/16.08	91.95/96.76	48.23/50.22	63.28/66.12
KNN1	49.64/53.11	**83.47**/92.00	40.27/56.76	54.33/70.20	55.19/43.64	43.95/72.41	48.93/54.46	23.54/17.47	70.16/82.24	35.26/28.82	92.42/95.63	49.86/58.01	64.77/72.21
MLP1	51.17/**55.81**	68.67/95.83	38.51/56.76	49.35/71.29	59.07/44.07	53.62/75.86	56.21/55.75	**26.09/20.89**	71.62/84.11	**38.25/33.47**	92.19/95.83	49.73/58.87	64.61/72.94
LDAsmall3	20.42/28.32	14.77/21.15	18.92/29.73	16.59/24.72	21.81/18.97	54.64/72.41	31.18/30.07	4.53/9.31	21.98/43.93	7.51/15.37	74.36/75.13	32.03/45.02	44.77/56.30
LDAbig3	19.72/27.15	13.40/18.87	17.16/27.03	15.05/22.22	21.18/19.25	53.49/72.41	30.34/30.42	4.45/8.63	21.98/42.06	7.40/14.33	75.22/75.69	31.86/44.59	44.76/56.12
QDAsmall3	26.36/26.60	29.72/42.59	34.46/59.46	31.91/49.63	13.89/9.25	40.55/51.72	20.69/15.70	8.07/6.85	61.65/75.70	14.28/12.57	69.17/66.47	40.62/41.56	51.18/51.14
QDAbig3	26.43/26.50	29.25/40.35	34.19/59.46	31.53/48.08	14.17/9.64	41.00/51.72	21.06/16.26	8.10/6.77	61.60/73.83	14.31/12.41	68.96/64.53	40.94/41.99	51.38/50.88
LinearSVM3	41.19/44.99	63.68/91.67	36.49/59.46	46.39/72.13	35.22/27.46	48.24/70.69	40.71/39.56	14.48/13.08	76.19/87.85	24.34/22.77	88.84/95.07	48.36/47.19	62.63/63.07
LinearSGD-SVM3	42.63/46.50	61.62/91.30	32.97/54.05	42.96/67.91	37.81/26.06	55.54/70.69	44.99/38.08	16.25/15.64	75.11/87.85	26.72/26.56	89.16/94.30	49.89/53.25	63.98/68.06
RBF-SVM3	44.29/47.62	75.98/**100.00**	34.19/56.76	47.16/72.41	38.20/26.28	53.81/68.97	44.68/38.05	17.26/16.18	76.71/86.92	28.18/27.28	87.75/92.94	51.38/54.11	64.82/68.40
RandomForest3	32.42/33.83	77.59/89.47	31.35/43.24	44.66/58.31	19.81/17.86	31.39/50.00	24.29/26.32	9.49/8.35	65.98/73.83	16.59/15.00	87.63/94.44	47.21/40.69	61.36/56.88
KNN3	40.86/43.96	76.09/**100.00**	30.54/54.05	43.59/70.18	31.94/22.96	51.31/77.59	39.37/35.43	16.50/13.48	76.19/86.92	27.13/23.34	89.11/95.65	45.09/48.48	59.88/64.35
MLP3	42.00/45.32	76.62/95.24	33.65/56.76	46.76/71.13	34.13/24.85	52.85/70.69	41.48/36.77	15.37/14.55	76.94/86.92	25.63/24.93	88.79/93.88	48.37/49.35	62.63/64.69
LDAsmall5	13.79/21.33	8.17/18.75	7.84/16.22	8.00/17.39	15.36/12.04	44.91/58.62	22.89/19.97	3.41/8.89	18.16/42.99	5.75/14.73	70.20/69.28	22.78/28.57	34.39/40.46
LDAbig5	13.84/21.96	7.71/20.00	8.24/16.22	7.97/17.91	14.64/13.00	46.19/67.24	22.23/21.79	3.30/8.37	19.39/42.06	5.64/13.96	71.37/77.70	24.67/29.44	36.67/42.70
QDAsmall5	20.99/25.35	15.34/30.43	22.70/43.24	18.31/35.73	12.69/14.39	38.05/51.72	19.03/22.51	7.90/8.03	55.25/60.75	13.83/14.19	55.11/52.05	31.72/27.71	40.27/36.16
QDAbig5	21.08/24.68	15.36/32.61	22.84/43.24	18.37/37.18	12.90/11.89	38.57/55.17	19.33/19.56	7.87/7.85	55.29/59.81	13.77/13.88	55.10/53.06	31.82/28.14	40.34/36.78
LinearSVM5	30.63/35.19	53.44/85.00	23.11/40.54	32.26/54.90	20.85/18.59	43.75/62.07	28.24/28.61	10.34/12.09	67.67/78.50	17.94/20.95	81.70/85.44	40.17/32.03	53.85/46.60
LinearSGD-SVM5	32.03/37.40	52.90/88.24	20.95/40.54	30.01/55.56	24.40/19.44	48.11/63.79	32.38/29.80	11.50/13.61	70.31/83.18	19.77/23.39	82.96/87.39	40.90/35.50	54.79/50.49
RBF-SVM5	35.52/42.33	60.90/**100.00**	23.78/40.54	34.21/57.69	27.50/24.41	53.17/67.24	36.25/35.82	13.36/16.05	70.49/85.05	22.46/27.00	80.24/82.61	44.40/44.16	57.16/57.55
RandomForest5	26.76/30.13	70.61/**100.00**	23.38/35.14	35.13/52.00	15.23/15.15	27.61/34.48	19.63/21.05	7.58/9.38	61.04/66.36	13.48/16.44	79.48/89.89	42.27/30.74	55.19/45.81
MLP5	30.71/35.12	63.24/89.47	21.62/37.84	32.23/53.18	20.46/19.21	48.75/68.97	28.83/30.04	10.57/11.30	69.08/83.18	18.34/19.89	79.61/84.68	38.85/33.33	52.22/47.84
LDAsmall9	7.96/8.81	7.69/17.65	2.70/2.70	4.00/4.69	9.79/9.09	27.74/44.83	14.47/15.12	2.22/3.54	11.39/28.04	3.72/6.29	57.95/50.00	11.10/7.79	18.63/13.48
LDAbig9	7.82/10.80	4.64/10.00	2.16/2.70	2.95/4.26	9.22/13.42	34.91/58.62	14.58/21.84	2.13/3.96	15.72/32.71	3.75/7.06	59.38/59.00	14.44/12.55	23.23/20.70
QDAsmall9	13.09/16.03	6.03/11.76	10.00/16.22	7.52/13.64	8.75/13.16	28.06/36.21	13.34/19.30	6.05/7.77	43.91/45.79	10.64/13.29	40.87/31.82	20.72/13.42	27.50/18.88
QDAbig9	13.24/16.28	6.08/12.12	10.14/16.22	7.60/13.87	8.91/12.50	28.76/36.21	13.61/18.58	6.08/8.06	44.33/46.73	10.69/13.76	41.06/30.34	21.04/14.72	27.83/19.82
LinearSVM9	16.12/15.97	29.82/36.36	6.89/8.11	11.20/13.26	9.93/11.70	32.93/37.93	15.26/17.89	5.96/5.36	54.96/49.53	10.76/9.67	63.19/64.56	25.92/18.18	36.76/28.37
LinearSGD-SVM9	18.52/20.25	39.47/50.00	8.11/8.11	13.45/13.95	11.96/18.31	35.36/43.10	17.88/25.70	6.90/8.77	58.07/57.01	12.33/15.20	66.31/68.42	28.34/19.91	39.71/30.85
RBF-SVM9	21.74/26.83	31.98/40.00	7.43/10.81	12.06/17.02	19.42/25.00	48.62/48.28	27.75/32.94	8.98/14.51	56.09/57.94	15.49/23.21	62.76/70.65	32.81/27.71	43.09/39.80
RandomForest9	17.79/17.75	50.37/71.43	9.19/8.11	15.54/14.56	9.33/9.26	22.68/29.31	13.22/14.07	6.00/8.06	53.65/48.60	10.79/13.82	65.85/73.85	34.31/22.94	45.11/35.01
MLP9	19.13/20.85	37.41/50.00	7.43/8.11	12.40/13.95	14.16/18.75	41.38/48.28	21.10/27.01	7.55/9.29	56.89/56.07	13.33/15.94	63.72/69.14	27.44/20.35	38.36/31.44
LDAsmall-GLCM3	30.36/31.69	29.49/29.41	25.95/43.24	27.61/35.01	23.02/17.74	64.13/77.59	33.87/28.88	12.25/11.14	54.54/78.50	20.01/19.52	74.32/71.36	32.69/39.83	45.41/51.12
QDAsmall-GLCM3	25.39/24.84	10.94/12.30	36.08/56.76	16.79/20.22	19.79/16.81	41.64/60.34	26.83/26.29	10.60/8.04	56.09/69.16	17.83/14.41	62.62/53.06	44.06/46.75	51.72/49.71
LinearSVM-GLCM3	41.14/45.43	59.87/76.67	38.51/59.46	46.88/66.98	32.13/25.97	**64.45**/81.03	42.88/39.33	15.25/15.24	74.31/**88.79**	25.30/26.02	78.53/80.49	43.90/50.65	56.32/62.17
RBF-SVM-GLCM3	34.87/36.27	16.61/21.37	35.81/64.86	22.70/32.15	34.40/28.77	60.73/74.14	43.92/41.45	17.02/12.79	63.39/82.24	26.83/22.14	69.63/66.67	45.85/52.38	55.29/58.67
RandomForest-GLCM3	32.73/37.03	60.38/85.19	34.19/59.46	43.66/70.03	19.51/18.69	30.69/50.00	23.85/27.21	11.14/9.83	64.85/75.70	19.02/17.40	81.17/91.37	45.08/41.13	57.97/56.72
KNN-GLCM3	38.47/43.29	44.78/65.00	36.49/**67.57**	40.21/66.26	30.72/26.52	62.01/75.86	41.09/39.30	15.30/13.72	69.18/85.05	25.06/23.62	71.70/72.25	41.96/47.19	52.94/57.09
CNN5	30.07/34.27	55.60/88.24	20.14/32.43	29.56/47.43	22.59/20.42	37.92/60.34	28.31/30.52	10.52/11.17	65.04/72.90	18.10/19.37	77.84/85.12	41.30/34.63	53.96/49.23
LDAsmall-GLCM5	29.88/30.29	44.26/40.74	17.70/27.03	25.29/32.50	22.86/17.48	55.54/72.41	32.39/28.16	13.21/11.43	58.45/72.90	21.54/19.77	69.21/69.06	33.55/35.06	45.19/46.51
QDAsmall-GLCM5	13.93/15.70	2.97/5.78	17.84/37.84	5.09/10.02	11.95/11.69	23.38/34.48	15.82/17.46	7.02/5.84	42.26/41.12	12.04/10.23	44.44/44.97	34.40/27.27	38.78/33.95
LinearSVM-GLCM5	31.42/35.55	33.59/59.26	23.65/37.84	27.76/46.19	25.18/22.45	53.62/70.69	34.27/34.08	12.97/13.30	69.93/77.57	21.88/22.70	68.18/69.59	35.69/32.90	46.86/44.68
RBF-SVM-GLCM5	26.71/29.72	7.49/12.80	22.16/40.54	11.20/19.46	31.71/28.44	42.92/60.34	36.47/38.66	17.85/14.80	56.66/63.55	27.15/24.01	56.52/59.88	38.69/33.77	45.93/43.18
RandomForest-GLCM5	25.12/26.80	43.43/70.00	21.89/32.43	29.11/44.33	13.22/11.92	25.69/36.21	17.45/17.94	9.13/8.06	58.49/61.68	15.80/14.26	73.01/84.21	37.60/31.17	49.64/45.50
CNN9	17.11/18.93	36.64/50.00	6.49/10.81	11.02/17.78	11.00/12.15	26.20/37.93	15.50/18.40	7.09/6.96	53.08/51.40	12.51/12.26	65.36/69.51	28.96/20.78	40.13/31.99
LDAsmall-GLCM9	23.24/24.32	18.47/25.00	7.84/16.22	11.01/19.67	22.79/20.99	27.23/29.31	24.81/24.46	15.53/12.05	44.71/45.79	23.05/19.08	59.54/51.30	37.93/30.30	46.34/38.10
QDAsmall-GLCM9	5.39/5.48	0.50/0.53	5.54/10.81	0.91/1.02	3.90/3.70	6.66/8.62	4.92/5.18	4.43/6.13	27.29/20.56	7.62/9.45	28.32/29.47	21.71/12.99	24.58/18.03
LinearSVM-GLCM9	18.63/22.06	19.48/37.50	10.14/18.92	13.33/25.15	13.20/16.49	29.92/36.21	18.31/22.66	8.45/9.54	56.94/53.27	14.71/16.19	53.61/58.23	24.41/16.45	33.55/25.65
RBF-SVM-GLCM9	16.09/18.72	3.86/11.11	9.73/18.92	5.53/14.00	16.76/13.10	22.17/24.14	19.08/16.98	12.26/14.20	44.66/43.93	19.24/21.46	42.69/39.45	26.89/17.32	33.00/24.07
RandomForest-GLCM9	14.02/16.43	24.08/43.75	7.97/13.51	11.98/20.65	5.31/9.76	13.77/20.69	7.66/13.26	6.55/6.20	48.24/44.86	11.53/10.89	57.70/51.32	26.68/16.02	36.48/24.41

APPENDIX B — TESTING ALTERNATIVES TO MEDIAN NORMALIZATION

In this appendix, we show results for two alternative normalization to the subtraction by the median used through out this thesis. The first one is the subtraction by the mean forest vector (see Table B.1), and the second one is the subtraction by the mean vector followed by the division by the forest variance vector (see Table B.2). We have run the experiments for these alternatives normalization just for a few selected classifiers due to lack of time. By comparing the best methods from each of these two tables with the best one from Table 5.2 (which shows results for the median normalization) we can see that the best method using median normalization obtains better results than the alternatives.

Table B.1: Overall test tiles deforestation detection F1-Scores (defined in Section 5.2.7) for classifiers trained and tested on TOA reflectance images that were normalized using the ground truth forest **mean** vector. Each table cell shows F1-Scores in the format a/b, where 'a' is the area based score and 'b' is the alert based score as defined in Section 5.2.7. The best area and alert F1-Scores are highlighted in bold.

Method/Context Size	1×1	3×3	5×5	9×9
LDAsmall	27.42%/30.88%	23.01%/28.79%	16.3%/21.5%	9.2%/12.04%
QDAsmall	47.6%/49.91%	31.65%/30.39%	23.53%/26.78%	13.86%/16.75%
LinearSVM	**65.65%/72.84%**	54.64%/62.21%	41.57%/46.78%	27.32%/30.36%
RBF-SVM	55.3%/60.93%	44.06%/50.81%	34.95%/41.71%	24.82%/27.4%

Table B.2: Overall test tiles deforestation detection F1-Scores (defined in Section 5.2.7) for classifiers trained and tested on TOA reflectance images that were normalized by subtracting the ground truth forest **mean** vector and dividing by the forest pixel intensities **variances**. Each table cell shows F1-Scores in the format a/b, where 'a' is the area based score and 'b' is the alert based score as defined in Section 5.2.7. The best area and alert F1-Scores are highlighted in bold.

Method/Context Size	1×1	3×3	5×5	9×9
LDAsmall	17.56%/23.85%	13.95%/22.69%	10.87%/14.74%	7.95%/11.25%
QDAsmall	57.28%/58.59%	39.99%/42.94%	28.49%/36.6%	17.15%/20.31%
LinearSVM	**65.08%/71.18%**	56.12%/64.69%	43.26%/50.99%	27.44%/28.96%
RBF-SVM	60.81%/68.47%	49.79%/58.34%	39.13%/43.43%	27.28%/25.83%