Bachelor Thesis

Cocoa segmentation in Satellite images with deep learning

Guillem Bonet Filella

June 5, 2018

Institute of Geodesy and Photogrammetry
ETH Zurich

Professorship
Prof. Dr. Konrad Schindler

Supervisor
Dr. Jan Dirk Wegner

Abstract

Cocoa is the basic ingredient needed to produce chocolate. It is grown in many tropical countries like Ivory Coast, Ghana, and Ecuador. For market research, big chocolate companies need to know the overall acreage of cocoa and a rough estimation of annual yield to adjust their buying strategy accordingly. Today, this is done manually by sending experts to the field that regularly measure the size of cacao beans and estimate the area of cocoa plantations.

The aim of this project is to combine satellite images of the new ESA constellation Sentinel-2 and deep learning to segment cocoa planting sites in Africa and Latin America. What makes this task particularly hard is the high similarity of cocoa and surrounding plants, often smallholder farms especially in Africa, and the inhomogeneous acquisition frequency due to frequent cloud coverage.

For this task, we have developed a complete method, from the preprocessing of Sentinel-2 imagery to classification of cocoa with a convolutional neural networks. Furthermore, we have analyzed the properties and main features that play a key role in the process of cocoa segmentation and next steps for the development of the project are suggested.

Acknowledgements

I would like to thank:

- **Dr. Jan Dirk Wegner** for giving me the opportunity and his trust to work with him on this project, for the permanent open door and the multitude of helpful tips and advice.
- **Andrés Rodríguez Escallón** for his constant support and valuable feedback to my questions.
- **Dr. Montserrat Filella** for helping me in the process of expressing my thoughts in words.

Contents

1 Introduction

Why is the use of land cover segmentation important for agriculture?
With the world's population rapidly growing and nearly reaching eight billion people, supplying all human beings with food is becoming one of the biggest challenges for the humankind. The massive increase of the world population is augmenting the pressure on the agricultural production and the need for reliable information of crop status all over the Earth. This leads the agriculture to a critical situation where it has to optimize its principles and methods of functioning to maximize their production. In order to achieve these goals the management of the resources, especially in the developing countries, has to be massively improved.

Figure 1: Forastero beans[1]

Often one of the biggest problem in agriculture is the lack of correct or complete information about the crop distribution of the farms, the status of the plants or even the landownership of the farms. By using land cover maps generated trough segmentation, a lot of problems can be solved or at least simplified. For instance, predictions of the seasonal production can be established with the information of the amount, type, distribution and health status of the plants. This information can be not only important for the food supply of the own country, but also for the economy of the country and the international market. Even the anthropogenic influence of the human being on the Earth can be analyzed by the means of global population, land use and land cover maps (Ellis and Ramankutty 2008). With this information we can better understand the history and impact of the human kind on the system Earth.

As outlined in the previous paragraph, land cover segmentation is a powerful analysis tool that can be used in a multitude of scientific, technological and economic domains.

[1] https://www.barry-callebaut.com/about-us/media/press-kit/history-chocolate/theobroma-cacao-food-gods

Why is the knowledge of cocoa segmentation important?
Cocoa is one of the most economically important crops on Earth. For the majority of the producing countries it is one of the critical exports and for the consuming countries a key import. It is typical that the countries with the highest consumption of cocoa do not produce cocoa themselves, since these country usually do not have appropriate climates for the production of this sensitive crop.

Cocoa is mostly produced in Africa, Asia and South America. In 2016 the biggest cocoa producing countries were the Ivory Coast (1,472,313 tonnes per year[2]), Ghana (858,720 tonnes per year[2]), Indonesia (656,817 tonnes per year[2]) and Cameroon (291,512 tonnes per year[2]). The two biggest South American producing countries were Brazil (213,843 tonnes per year[2]) and Ecuador (177,551 tonnes per year[2]).

Unlike the more industrialized crops, $80 - 90\%$ of cocoa is still produced in small farms[3]. Most of the farmers work with outdated farming practices and have limited organizational influence on the market. In the producing countries there are many different practices of growing cocoa and its distribution has rapidly changed in the past decades. The typical method of cocoa farming is the cocoa agroforest where cocoa is planted beside mature timber trees and under giant trees, to provide shades for the crops. In these agroforests cocoa is often planted with other varieties of crops as this increases the income security of the farmers over the whole year. The agroforests arose in a time where the population density was low, land and forests abundant, fertilizers unknown and the limiting factors were labor and capital. This method reduces the maintenance work and increases biodiversity, but needs more land and cocoa is quite slow to mature (Ruf 2011).

Figure 2: Farm in Ecuador[4]

[2]Food and Agriculture Organization of the United Nations: www.fao.org
[3]Cocoa Market Update: http://www.worldcocoafoundation.org/wp-content/uploads/Cocoa-Market-Update-as-of-4-1-2014.pdf

As population increased and migration intensified in the last decades, other strategies, such as the full sun farming, became more favored. In full sun cocoa farming, the cocoa is planted in a single layer structure and hybrid cocoa plants, that can resist the direct sun exposure and give higher yields, are used. In a period of 20 to 25 years, the unshaded and hybrid farms are more profitable than the shaded variant, since their peak yields are earlier and higher (Obiri et al. 2007). Combined with the moderate use of pesticides, fertilizers and herbicides, the abundant yield and return of unshaded cocoa farms can be maintained for 25 to 30 years (Ruf 2007). For this method, large areas of forests and agroforests are felled or burned to win land for new farming. This is called slash-and-burn farming. These methods to acquire land have significantly damaged biodiversity and the forests of the majority of the cocoa-growing countries, such as the Haut-Sassandra forest in Ivory Coast (Barima et al. 2016) and the Bia Conservation Area and Krokosua Hills Forest Reserve in Ghana (Asare et al. 2014). Figure 2 shows an example of a full sun cocoa farm in Ecuador.

All these issues around cocoa such as the amount and efficiency of cocoa production, biodiversity and forest preservation can be observed and analyzed trough cocoa segmentation and cocoa mapping of the affected regions. But not only processes directly linked to cocoa production can be distinguished. For instance, also migration and child labor on the farms can be spotted.

Why do we use satellite images for the task of segmentation?
Data collection is a difficult task in most technical and scientific research fields. Before satellites and air-based methods were available, the obtainment of the necessary data to produce a map of a region was an expensive and long lasting undertaking, since the work had to be manually fulfilled. Satellites provide large amount of data of vast areal extent and high temporal resolution. This quantity of data would not be possible to obtain using the typical land surveying methods.

Therefore, plenty of new remote sensing have been developed or refined in the past years. For instance, space-based radar missions, such as Radarsat-2 launched in 2007, light detection and ranging (LIDAR) missions, such as ICESat launched in 2003, or multispectral imagery missions, such as Sentinel-2 launched in 2015.

Data collected by satellite is nowadays used in multiple applications. For instance, the observation of the deforestation of the tropical forest in Central Africa in order to monitor the change and optimize forest management (Duveiller et al. 2008), the mapping of galamsey gold mines in Ghana to analyze their relationship with the cocoa agriculture (Snapir et al. 2017) or the mapping crop types to "provide crucial information for agricultural monitoring and management" (Inglada et al. 2016).

[4]https://www.confectionerynews.com/Article/2015/10/07/Chocolate-firms-eye-Ecuador-for-single-estate-cocoa-Hacienda-Victoria

[5]https://news.mongabay.com/2016/07/huge-cacao-plantation-in-peru-illegally-developed-on-forest-zoned-land/

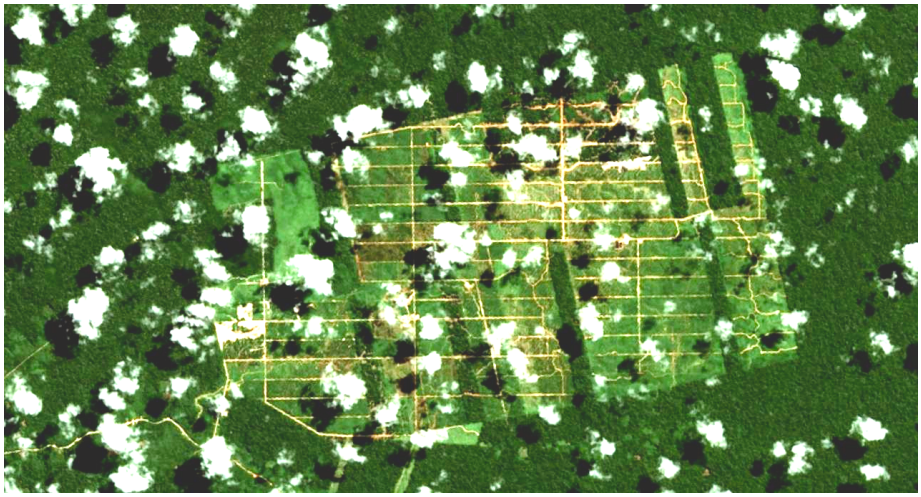


Figure 3: Satellite image of a cocoa farm on forest zone in Peru[5]

Why do we use deep learning and convolutional neural network for this task?

Deep learning, a type of machine learning techniques, is a family of self-optimization methods that have more than one layer between input and output and, thus, show a complex inner structure. The term deep learning was created by Rina Dechter in 1986 and a multitude of varieties have emerged since then such as the multilayer perceptrons (Ivakhnenko 1965), backpropagation algorithms, deep neural networks and convolutional deep neural networks (LeCun et al. 1989).

A big revolution in the field of computer vision has been going on since 2012 when it started with the introduction of stronger convolutional deep neural networks such as the AlexNet (Krizhevsky et al. 2012) that outperform all the previous classifiers at that time. This evolution has been enhanced by the availability of big, already labeled data sets and powerful GPU implementations.

Nowadays, convolutional deep neural networks are the front runner for the task of image classification as it is a self learning method that combines different features and properties of objects and can detect complex topological relations between different items in an image.

The objectives of this Bachelor thesis are:

- Develop a method for cocoa segmentation by combining deep learning methods and satellite-based multispectral imagery and, thus, improve the feasibility of cocoa segmentation.
- Analyze and identify the decisive properties and features of cocoa during the segmentation process.

2 Theoretical Principles and Multispectral Images

2.1 Image Classification

Image Classification is the problem of assigning a certain label (object) to an image as a whole or to each pixel of an image. This task of recognizing objects in images is not as trivial for a computer as it is for a human being. What for a person is a banality represents an immense challenge for programmers to implement. The main challenges of this task are viewpoint variations, scale variations, deformations, occlusions, illumination conditions, background clutters and intra-class variations (Karpathy 2018). A good classifier should be able to overcome all these difficulties and still be accurate in its prediction.

There is a great variety of classifiers that are used nowadays in different tasks such as robot applications, land cover segmentation or even autonomous driving cars. Some of the classifiers are:

- **Textual case-based reasoning**: Type of classifiers that tries to analyze images based on its textual characteristics. For instance, Co-occurrence matrix, Laws' Technique (selection of filters is limited to the existing banks), Histogram of oriented gradients and Local Binary Patterns (Wegner 2017).
- **Extraction of interest points**: These methods extract certain attributes of an image to deduce its content. For example, Moravec corner detector, Harris corner detector and Förstner corner detector (ibid.).
- **Nearest neighbor classifier** is one of the most basic classifiers. It compares two images by calculating the sum of the distances between the pixels of the test image and the training image. The distance can be computed with a L1 distance (linear sum) or a L2 distance (square root of the quadratic sum). The classifier then chooses the label (training image) with the smallest score (distance). The result gives an idea of the similitude between the two images. The nearest neighbor classifier is really simple to implement, but can neither detect spatially nor radiometrically translated, rotated or scaled images (Karpathy 2018).
- **k-nearest neighbors classifier** is the general form of the nearest neighbor classifier. It determines the k nearest training images to the test image and determines then the label of this image. In the particular case that $k = 1$, the k-nearest neighbors classifier turns into the simple nearest neighbors classifier. These methods have a long testing time duration, as they have to compare the test image to every single training image. There are new, more refined classifiers based on the same concept such as several **approximate nearest neighbor**. The nearest neighbor classifiers are not often used for classifying high-dimensional images due to the contradictory interpretation of distances in high-dimensional color spaces (ibid.).
- **Neural networks** (see section 2.1.1)
- **Convolutional neural networks** (see section 2.1.2)

There is no best or universal classifier suited for every task, since each classifier has its strengths and weaknesses. For instance, a convolutional neural network is a powerful classifier that once trained is really fast and accurate but needs a considerable amount of data and time to be correctly trained.

2.1.1 Neural Networks

The conceptual structure of a neural network finds its origin in the biological structure of a neuron. Even if there are a lot of different nerve cells in the human body and each neuron has its own procedures and functions in the human system, there are also a lot of similitudes between them and the neural network neurons (Figure 4). Both have connections to previous neurons (dendrites), process the received information (nucleus) and have an activation (axon) to decide whether or not to pass information to the next neurons.

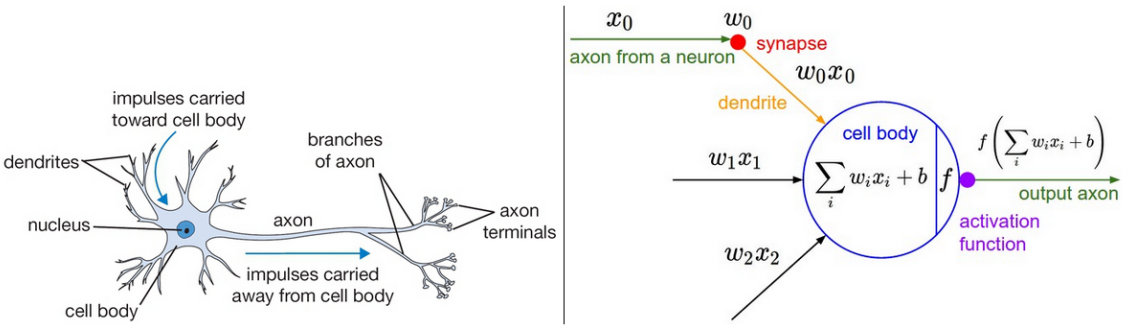


Figure 4: Representation of a biological neuron (left) and a neural network neuron (right) (Karpathy 2018)

Data split For neural networks, data sets are often divided into three separate sets that have different functions during the classification process (Figure 5):

- **Training data set** is the data set used to train the model and its parameters in order to fit the given problem.
- **Validation data set** is the data set used to evaluate the model fit during training. It is used to fine-tune the hyperparameters that determine the structure of the network and how it is trained.
- **Test data set** is the data set used to evaluate the model fit at the end of training. Since the test data set has never been in contact with the model during training, it is an independent evaluation of the result.

It is important to consequently separate the three data sets since they should be independent from each other.

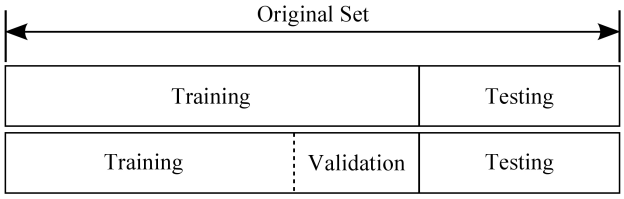


Figure 5: Data Split[6]

Main components The basic structural approach of a neural network has three main components: a **score function** to assign the training data into predefined classes, a **loss function** to determine the correlation between the result of the score function and the ground truth and an **optimizer** to upgrade the parameter of the score function in order to minimize the loss.

Linear classifier A simple score function is the linear classifier:

$$f(x_i, W, b) = Wx_i + b \tag{1}$$

where $\boldsymbol{x_i}$ are the pixels of an image flattened out into single column. **W** and **b** are the parameters of the score function, called weights and bias vector.

The product $\boldsymbol{Wx_i}$ performs parallel separate classifiers for each predefined class. The number of rows of the weight matrix is the number of different classifiers and therefore of distinct classes. These parameters are learned with the training data set. It is important to note that after training these parameter, we do not need the training data anymore to test further data sets. Therefore, a linear classifier is called a parametric approach. The result of the linear classifier is a vector with one score for each class. Logically, the prediction of the classifier is the class with the highest score. Since the linear classifier only performs a matrix multiplication and an addition, it is a really fast method to obtain a prediction.

Loss function The result from the score function has to be transformed into a comparable score. This is achieved by the loss function. A high loss implies that the classifier is working badly and the prediction from the score function is far from the ground truth; a low loss that the classifier is correctly identifying the images. There are two major methods to calculate a loss: the **Support Vector Machine** (SVM) and the **Softmax classifier** (Karpathy 2018).

The SVM loss is formulated as:

$$L_i = \sum_{j \neq y_i} max(0, s_j - s_{y_i} + \Delta) \tag{2}$$

[6]http://www.rpubs.com/charlydethibault/348566

where $\boldsymbol{s_j}$ is the class score of the correct class and $\boldsymbol{s_{y_i}}$ are the class scores of the other classes. $\boldsymbol{\Delta}$ is a hyperparameter.

The SVM is a method that seeks the score $\boldsymbol{s_j}$ of the correct class to be $\boldsymbol{\Delta}$ higher than the score $\boldsymbol{s_{y_i}}$ of the incorrect classes. The $max(0, -)$ function is a threshold function, called hinge loss. It sets every negative number to zero.

The Softmax classifier corresponds to the cross-entropy of the scores for each class:

$$L_i = -\log\left(\frac{e^{s_{y_i}}}{\sum_j e^{s_j}}\right) \tag{3}$$

where $\boldsymbol{s_{y_i}}$ is the class score of the correct class and $\boldsymbol{\sum_j e^{s_j}}$ the sum of all the class scores. The Softmax function takes a vector and squeezes it into a vector with values between zero and one, that sum up to one. From the point of view of probability theory, the output can be seen as a generalized Bernoulli distribution that represents the normalized outcome probability for each class.

Whereas the SVM loss directly sets the loss to 0 when the correct score is Δ higher than the score of the incorrect classes, the Softmax classifier also gives an information about how much higher the predicted score is from the other ones. For example, the SVM loss ($\Delta = 1$) for [4, 4, 5] and [1, 2, 25] would be the same, however the Softmax loss would not. In practice the SVM loss and the Softmax loss are equally used because there are no big divergences between the results of both methods.

Optimization The optimizer tries to find the best possible set of weights $\boldsymbol{W}$ for the given classification problem. Due to the fact that this is a nearly impossible task, the optimizer attempts at each iteration to find a set of new weights $\boldsymbol{W}$ that is just a little bit better then the previous one. By starting at a random matrix $\boldsymbol{W}$ and improving it at each iteration, the optimizer slowly finds a good set of weights $\boldsymbol{W}$.

Gradient descent is the most common method used to optimize neural networks. The gradient descent computes the gradient of the loss function $\boldsymbol{\nabla_W L(W)}$ with respect to the parameters $\boldsymbol{W}$. The loss function $\boldsymbol{L(W)}$ is then minimized by updating the parameters $\boldsymbol{W}$ in the opposite direction of the gradient. This can be compared to always walking downhill until finding the lowest point of a valley.

The **learning rate** defines the pace at which the parameters are updated at each iteration. This is an important hyperparameter that has to be determined during validation because a high learning rate can prevent the optimizer to converge on the desired minima. The learning rate is often adjusted during training to increase the amount details learned.

There are three different main variants of the gradient descent that vary between the accuracy of the computed gradient and the computation time for each update. The **batch gradient descent** always uses the whole training data set to compute the gradient of

the loss function. This computes the best possible gradient but is very time consuming. On the contrary, the **stochastic gradient descent** (SGD) only executes the gradient over a single training example. This method often works well because successive updates frequently have the same direction. Compared to the batch gradient descent, the SGD is less accurate, but therefore less time consuming. The **mini-batch gradient descent** is midway between the two other methods, as it computes the gradient of small parts of the training set, called mini-batches.

Some methods have been developed to optimize the SGD performance in difficult parameter spaces. For instance, the **momentum** method that, by adding a fraction of the previous update to the current update, reduces oscillation in parameter spaces with big slope differences. This basic idea has been refined in methods such as the **Nesterov accelerated gradient** that calculates the current upgrade with an approximation of the parameters of the next step.

A recently created optimizer is the **Adam optimizer** (Kingma and Ba 2014). It is a first-order gradient-based stochastic optimization combining the advantages of two other established methods: the AdaGrad (Adaptive Gradient Algorithm (Duchi et al. 2011)) and the RMSProp (Root Mean Square Propagation). On one side, it adjusts the learning rate during the training (AdaGrad) to perform bigger updates for infrequent and smaller updates for frequent parameters. On the other side, it also adapts the learning rate (RMSProp) based on the average magnitude of the recent gradients. The Adam optimizers can be seen as a heavy ball with friction rolling downhill in the parameter space (Heusel et al. 2017).

Neural Network Architecture A Neural Network can be interpreted as a sequence of layers with a certain amount of neurons. Each neuron works like a linear classifier and the outputs of the neurons of a layer are the inputs of the neurons of the next layer. There are no cycles in neural networks as this would produce infinite iterations! The most common type of layers is the **fully-connected layer**, in which every neuron of a layer is connected to all the neurons of the next layers. Figure 6 shows an example of a simple neural network with 2 hidden layers (layers between input layer and output layer) and 4 neurons per hidden layer.

The input layer has the function to pass the input data to the neural network. Each channel of the input data is then separately classified by the neurons of the first hidden layer and passed on to the next layer. At the end, the output layer receives the result of the network, which represent the class score from that iteration. This score is then transformed into a loss, which then can be minimized by an optimizer. This procedure is then iterated until reaching a satisfactory result. For a neural network, increasing the depth (amount of hidden layers) of the network to more than 4 hidden layers does not imply any significant improvement. This is not true for convolutional neural networks, where the depth plays an important role (Karpathy 2018).

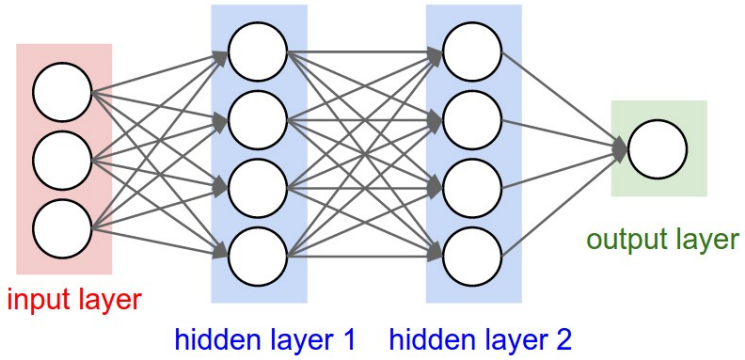


Figure 6: Basic neural network architecture (Karpathy 2018)

Data preprocessing Before training, data has to be preprocessed in such way that the different images have approximately the same range and distribution of pixel values. This is normally computed in two steps.

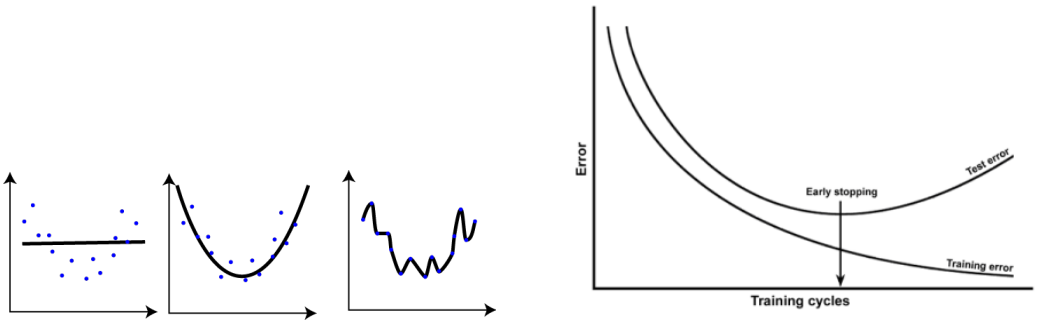
First, the **mean subtraction** is applied to the data in order to achieve a 0 centered data set. The mean is calculated for each band separately or the whole image and then subtracted from the bands or the whole image.

Secondly, the data is compressed between $[-1; 1]$ using the **normalization**. For this, the data is divided by the square root over the separate bands or the whole image.

The mean and square root of the data should only be calculated on the training data set and then applied to the training, validation and test data sets. This assures that the 3 data sets are still independent from each other and that no information of the test set is used during training. After these two preprocessing steps, the data is standard normally distributed.

Regularization A common problem while training a model is **overfitting**. This occurs when the parameters of the network are trained in such a way that the network "fits" too closely to the training data set and, thus, may not correctly classify other data sets (Figure 7 (a)). This can be monitored during training by surveilling the loss and the accuracy of the validation data set. Figure 7 (b) displays an overfitting model. We can observe that at a certain point the validation loss (test error) is not declining as the training loss (training error). At that point, the model starts to overfit and we can stop the model (early stopping) since it is the best solution for that training session. Regulation methods are added to the neural networks to prevent overfitting. Two commonly used methods are **L2 loss regularization** and **Dropout**.

[7] https://shapeofdata.wordpress.com/2013/03/26/general-regression-and-over-fitting/



(a) Example of overfitting (Left: underfitting. Center: good. Right: overfitting)

(b) Monitoring of the loss

Figure 7: Overfitting[7]

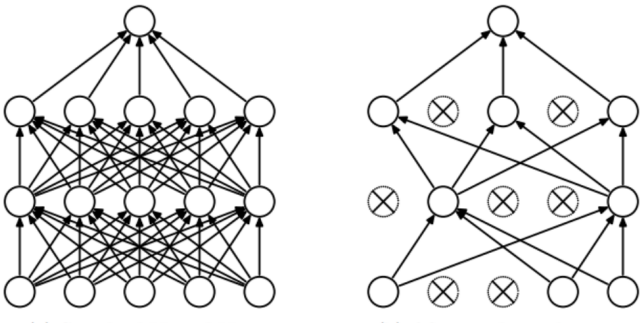
The regularization penalty **R(W)** is the sum of the L2 norm of the weights from the neural network:

$$R(W) = \sum_k \sum_l W_{k,l}^2 \tag{4}$$

$$L = L_{data} + \lambda R(W) \tag{5}$$

The term **R(W)** is added to the data loss $\boldsymbol{L_{data}}$ (SVM or Softmax) to prevent large weights in the neural network. These large weights could influence the network not to use all the given input data but only a small part of it. $\boldsymbol{\lambda}$ is a hyperparameter and defines the strength of the regularization.

Dropout (Srivastava et al. 2014) is a new method that, added to other methods, efficiently counteracts overfitting. The Dropout method consists in randomly dropping connections between the different neurons during training. This should prevent the different units to adjust too much to one another. Figure 8 shows a neural network before (left) and after applying Dropout (right).



(a) Standard Neural Net

(b) After applying dropout.

Figure 8: Dropout (Srivastava et al. 2014)

2.1.2 Convolutional Neural Networks

Convolutional Neural Networks (CNN) is a class of neural networks used to analyze image data. Introduced by LeCun et al. 1989, convolutional neural networks have produced excellent results in the tasks of classifying handwritten digits and face detection. By using certain characteristics of the images, they reduce the amount of parameters and can thus solve much more complex tasks than other standard classifiers. Since they have more hidden layers (depth) than an ordinary neural network, they need also less preprocessing because they can learn it by adjusting its own filters.

There are a lot of similarities between neural networks and convolutional neural networks such as the hidden layers, the weighs and biases, the loss function and the optimizer. The main difference is that convolutional neural networks use a multitude of different layers and have a bigger depth than the neural networks.

The reasons why convolutional neural networks have become so popular in the past few years are the availability of big, already labeled data sets, powerful GPU implementations and new regularization methods such as the dropout method. This recent evolution favors the development of convolutional neural networks.

Layers of a convolutional neural network In a convolutional neural networks there are various types of layers (Karpathy 2018):

- **Input Layer** is the first layer of the network. As for the neural network (see section 2.1.1), it passes the input data to the model.
- **Convolutional Layer** is the core element of the vast majority of convolutional neural networks. It can be seen as a set of learnable filters. However, each filter only uses a small area of the input data through all the depth (channels) of this input. The number of filters for every layer will represent the depth of its output.

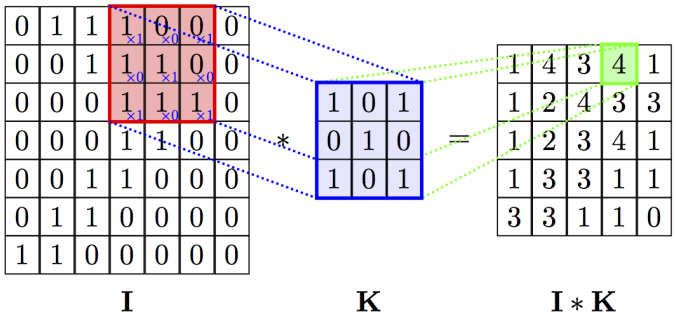


Figure 9: Convolution[8]

[8] https://cambridgespark.com/content/tutorials/convolutional-neural-networks-with-keras/index.html

Figure 9 is an example of a convolution with a 3×3 kernel, $stride = 1$ and $padding = 0$. The stride defines the moving steps of the filter over the image and the padding the width of the border zone filled with zeros. The kernel size, stride and padding are hyperparameters and determine the size of the layer output.

- **ReLU Layer**: The rectified linear unit is an activation function:

$$f(x) = max(0, x) \tag{6}$$

where $\boldsymbol{x}$ is the input of the ReLU layer. This activation function is applied element-wise to the input. By this, the negative values are set to 0. The introduction of ReLU layer in a model has shown to accelerate the convergence of the stochastic gradient descent by up to a factor 6 compared to an equivalent model with the activation function **tanh** (Krizhevsky et al. 2012).

Since in certain circumstances it has been observed that a ReLU layer can have the effect of "killing" the learning process (by reaching a point where the gradient computed by the model is always 0, being equivalent to no learning), more sophisticated varieties of this activation layer have been developed. An example is the **Leaky ReLU** where the negative values are not directly set to 0 but are instead set to a very small number to prevent this "dying effect".

- **Pooling Layer** is a downsampling layer. There are several forms of pooling such as **max pooling**, **average pooling** or **L2 norm pooling**. The most common one is the max pooling. This filter gives back as output the highest value of an area of the size $N \times M$ and repeats this with a certain stride. Figure 10 is an example of a max pooling with size 2×2 and stride 2.

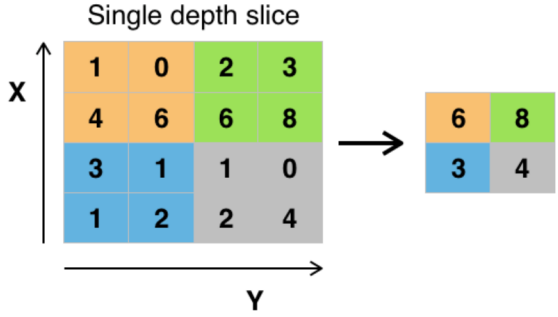


Figure 10: Max Pooling[9]

The purpose of the pooling layer is to reduce the amount of parameters and computations in a convolutional neural network. Additionally, it also controls overfitting to a certain extent.

[9]https://adeshpande3.github.io/A-Beginner%27s-Guide-To-Understanding-Convolutional-Neural-Networks-Part-2/

- **Batch Normalization** is often used to increase the stability of neural networks. This layer normalizes the output batch of the previous activation layer. This is implemented with a subtraction of the batch mean and a division by the batch standard deviation. This is comparable with the data preprocessing completed before training.

  The batch normalization allows the use of higher learning rates as it directly controls the outputs of the different activation layers. It also accelerates the learning process since the different layers do not have to learn themselves the normalization already computed by the batch normalization layer. Additionally, it has a small regularization effect (Ioffe and Szegedy 2015), sometimes even to the extent of eliminating the need for dropouts in a model.

- **Fully-Connected Layer** is the same as in the neural network (see section 2.1.1).

- **Transposed Convolutional Layer** is an upsampling layer. This is used to upsample the data that has been downsampled with the pooling layers back to its initial size. The transposed convolution can be seen as a backwards strided convolution.

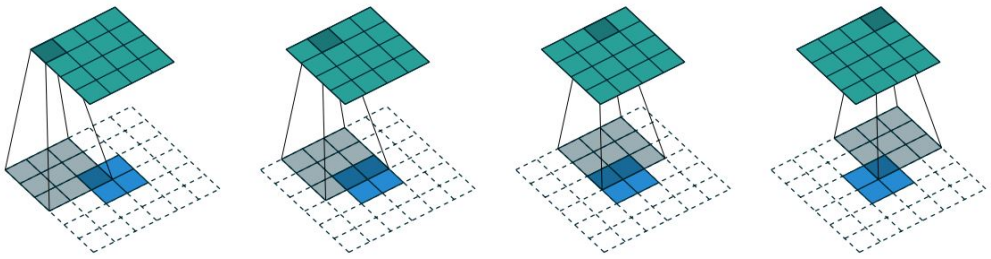


Figure 11: Transposed convolution[10]

Figure 11 can be interpreted in two different ways. First, this illustration shows a convolution (from green to blue) with a 3×3 kernel on a 4×4 input with $stride = 1$ and $padding = 0$. The size of the output is then 2×2. Secondly, it can also be interpreted as a transposed deconvolution (from blue to green) of a 2×2 input image with a 3×3 kernel, $padding = 2$ and $stride = 1$. In this case, the output has the size of 4×4.

Architectures of convolutional neural networks There are a lot of different architectures that have been developed in the past years. They differ from each other in the number, types and combination of different layers. Each one has its own strengths and weaknesses and thus also its applications. Some of the best known architectures are:

- **LeNet-5** (LeCun et al. 1998) is a pioneer in the world of computer vision. This model has long time been used to recognize handwritten numbers. Since the computation

[10] https://nrupatunga.github.io/convolution-2/

resources at that time were not as powerful as today, it had a small depth compared to modern architectures.

- **AlexNet** (Krizhevsky et al. 2012) outperformed all the previous classifiers in 2012. Having a similar architecture to LeNet-5, it was much deeper using SGD with momentum, dropouts and, for the first time, ReLU activation layers.
- **GoogleNet**, also called Inception, is an architecture from Google that won the ILSVRC 2014 competition. It performs at such a high level that it nearly beats the perception performance of a human being. It uses a new element ("Inception" module) based on a lot of small convolutions to reduce the amount of parameters. As a result, it reduces the number of parameters from 60 million (AlexNet) to 4 million. It also has no fully connected layers.
- **ResNet** (He et al. 2015) is an 152-layer model and is the winner of the ILSVRC 2015 competition. The extreme depth of the ResNet has been a revolution combined with the use of residual blocks that merge the output information from before and after a block build of two convolutional layers. The ResNet model has shown results outperforming the human being.

Data Augmentation The convolutional neural network is a classifier that, once trained, is fast, efficient and thus really powerful, but requires a big amount of training data to acquire good and stable results. Therefore, several techniques, such as data augmentation, have been developed to increase the amount of data when data are scarce. There are different types of data augmentation:

- **Translation**: The batches can be translated over the original image to create overlapping batches. This can be done with a fix or a randomly changing stride.
- **Rotation**: The batches can be rotated to create new batches. This is usually done with 90, 180 and 270 degrees to quadruple the amount of batches. However, it can also be computed with finer angles.
- **Flipping**: The batches can also be flipped on an horizontal or an vertical axis.
- **Scaling**: The batches can also be created with differently scalings in order to obtain more and less zoomed batches. This is the same as taking an image of the same scenery from different distances.
- **Radiometric Transformations**: The batches can also be transformed in radiometric ways by brightening, darkening or changing the contrast of the batches. This can be compared to taking the same image at different times of a day.

Unbalanced data Another frequent problem of classifications are unbalanced data sets. These are data sets where the proportion of the different classes in the images are widely different. If this unbalance is too high, it can hinder the convolutional neural network to correctly learn, since it will only predict the predominant class without considering the other classes.

This can lead to deceptive results because the accuracy can easily reach values over 90 % only predicting the more frequent class. This phenomenon is called **accuracy paradox**. Different methods to avoid or minimize the effects of unbalanced data sets are:

- Collecting **more data**.
- Using **other evaluation metrics** such as recall, precision and intersection over union (see section 4.1).
- **Undersampling** is a method that consists in eliminating the batches with solely the majority class. This can help to eliminate redundant data, balance the proportion between classes and thus accelerate the processing time and improve the results. But it must be noted that important information can be lost during this process.
- **Higher penalization** for errors of the minority class. This is an efficient way to increase the importance of minority classes during training.

2.2 Multispectral Images

A multispectral image is a superposition of sub-images of the same scenery taken at many different wavelengths. The best known type of multispectral images is the RGB image, composed of a red, a green and a blue sub-image (visible light), each representing a different wavelength in the electromagnetic spectrum.

The discrete area of the electromagnetic spectrum covered by each sub-image of a multispectral image is called a **band**. The sub-images are then stacked on a third dimension to one single multispectral image. As a consequence, the multispectral images have the dimension *[height of the sub-images, width of the sub-images, number of bands]*. While the RGB image only has three bands, there are cameras able to measure a large amount of bands not only in the visible but also in the non-visible part of the spectrum such as the near infrared or the short-wavelength infrared. This possibility of extracting information from different spectral bands has shown to be a powerful tool in a big variety of applications. The multispectral images are, for example, used for vegetation segmentation in satellite imagery, for detection of skin diseases or for analyses of art works.

A disadvantage of multispectral images is the increasing need for memory and with that the increasing processing time when working with them. However, this will not be a relevant problem in the future since the speed, capacity and memory of computers are rapidly increasing.

Further, **spectral signatures** can be created from multispectral images. These are compositions of the single values of the electromagnetic energy reflected by a certain object for each band. Spectral signatures result in distinct curves for every type of object containing the complete information of the reflectance of this object.

This method is often used in remote sensing to characterize and thus differentiate between various plants, soil, forest or other items on the earth's surface. In our case, we assume that the convolutional neural network will use the information of the spectral signature of the different objects to segment the images between them.

2.2.1 Sentinel-2

Sentinel is one of the latest Earth observation missions from ESA. This project has replaced older ESA-missions, which reached or are reaching their respective retirement dates, such as the ERS missions (1991 - 2011). The Sentinel satellites are equipped with a wide range of technologies, such as radars or multispectral imaging instruments used for land, ocean and atmospheric monitoring. There are five different Sentinel missions, each focusing on different aspects of Earth observation. Each mission has a constellation of two satellites to provide a periodic coverage of observation data of the whole Earth[11]:

- **Sentinel-1**: land and ocean monitoring using radar imaging
- **Sentinel-2**: land monitoring focusing on vegetation, soil and coastal areas using high-resolution optical multispectral imagery
- **Sentinel-3**: marine observation
- **Sentinel-4** and **Sentinel-5**: air quality monitoring

General Information Sentinel-2 is a high-resolution, multispectral imaging mission forming part of the Copernicus Programme, the largest Earth observation programme. The Sentinel-2 system is composed of two satellites. The S2A satellite was launched on the 23th of June 2015 and S2B on the 7th March 2017. Both satellites have an operation lifespan of approximately 7.5 years. The constellation of two sun-synchronous satellites flying in the same orbit with an altitude of 786 km and an inclination of 98.62° enables the possibility of monitoring the Earth with a frequency of five days and to always maintain the same angle of sunlight on the Earth's surface. These properties are useful for the creation of a consistent time series data collection.

[11] https://sentinel.esa.int/web/sentinel/home
[12] https://earth.esa.int/web/sentinel/user-guides/sentinel-2-msi/resolutions/spatial

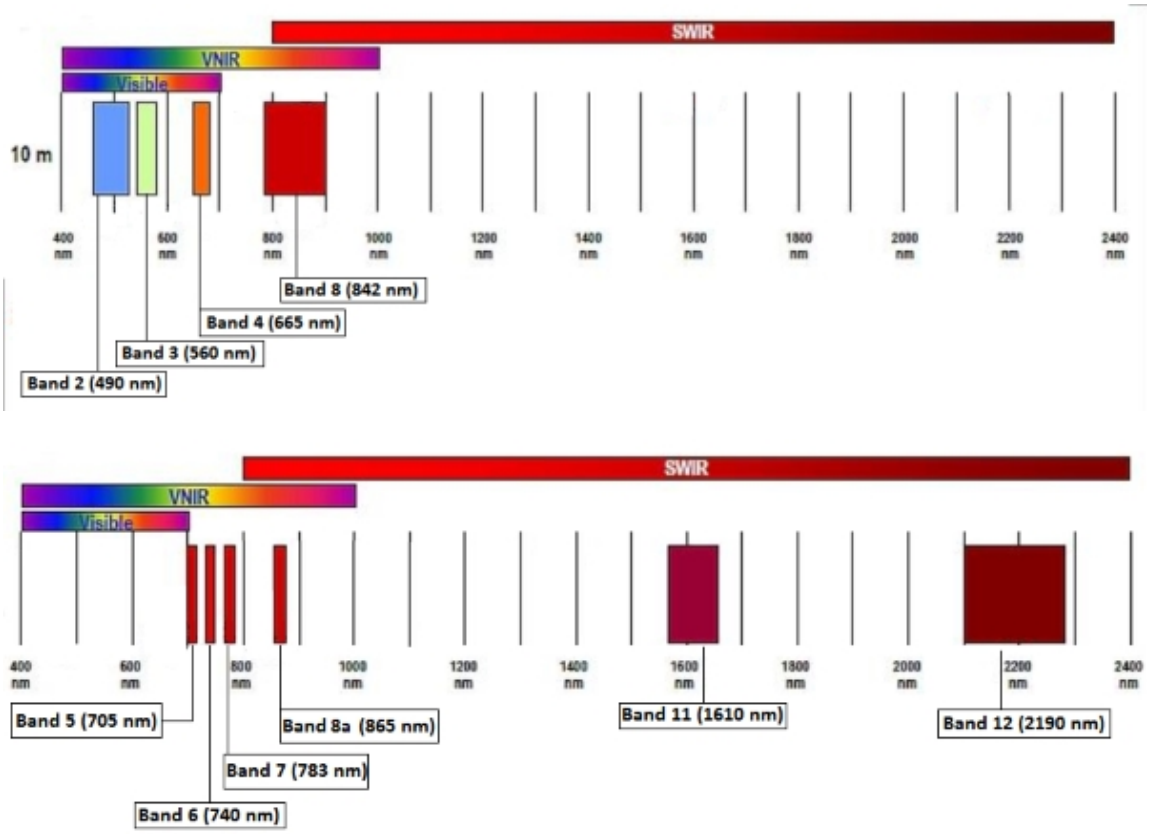


Figure 12: SENTINEL-2: 10 m (top) and 20 m (bottom) spatial resolution bands[12]

Bands Sentinel-2 measures 13 different bands. The spatial resolution depends on each observed spectral band. They are grouped in different resolutions: four at 10m (Figure 12), six at 20m (Figure 12) and three at 60m.

Each band covers a part of the electromagnetic spectrum and, therefore, contains a certain amount of information that can be valuable for segmentation:

- **B2** (490 nm, blue), **B3** (560 nm, green) and **B4** (665 nm, red) are the three RGB bands that cover the visible part of the spectrum.
- **B8** (842 nm) is in the near-infrared and has a large bandwidth (115 nm). It is often used in different indices such as the NVID, the normalized difference vegetation index to separate vegetation from other objects.
- **B5** (705 nm), **B6** (740 nm), **B7** (783 nm) are in the near-infrared between Band 4 and Band 8. With a spatial resolution of 20 m, they carry some extra information about the spectral reflectance of the terrestrial surface.
- **B11** (1610 nm) and **B12** (2190 nm) are in the short-wavelength infrared. Since both bands are good indicator of the moisture content of Earth's surface, they are excellent to differentiate between vegetation types.

The bands with a 60 m spatial resolution are used to measure the atmospheric conditions needed during the preprocessing of the images.

Product types The data measured by the Sentinel-2 satellites have to be preprocessed before being ready for applications. There are different product types corresponding to the different stages of this preprocessing[13]:

- **Level-1B**: The Level-1B product type contains the information of the top-of-atmosphere radiance values in sensor geometry. It also includes geometric information needed to generate the Level-1C product type
- **Level-1C** The Level-1C product type are $100x100km^2$ tiles containing the top-of-atmosphere reflectance in cartographic geometry (ortho-images in UTM/WGS84 projection). This product is generated using a Digital Elevation Model to project the imagery.
- **Level-2A** The Level-2A product type are bottom-of-atmosphere reflectance images in cartographic geometry. They are also $100x100km^2$ in UTM/WGS84 projection. At this moment, ESA is only generating this product type for images of the European continent. The Level-2A product type can be individually created by the users from the Level-1C product type using the **"Sentinel-2 Toolbox"**, called SNAP (Sentinel Application Platform).

[13]https://sentinel.esa.int/web/sentinel/user-guides/sentinel-2-msi/product-types

3 Methodology

In this section, we will illustrate the cocoa segmentation method developed for this project. For this purpose, we will individually introduce each component of the method and explain their importance, functioning and effect on the final results. Additionally, we will outline some methods in order to better understand the learning process and identify the decisive features during cocoa segmentation.

Considering that cocoa is planted with different techniques (section 1), we will distinguish between full sun farming and agroforestry by using Ecuadorian farms as a model for the full sun farms and Ghanian farms for the agroforestry.

3.1 Convolutional Neural Network

For the task of cocoa segmentation, we need a convolutional neural network that performs a semantic segmentation, which is a classification at pixel-level. As opposite to classifying the whole image to one label, semantic segmentation is necessary for this project because we not only want to determine the presence of cocoa in the analyzed area but also its distribution and the total amount.

The program developed in this project is implemented on **TensorFlow**, an open-source software library for dataflow programming based on Python and developed by the Google Brain team.

3.1.1 Architecture: U-Net

We choose an architecture based on U-Net (Ronneberger et al. 2015), a popular semantic segmentation model. The U-Net model is an encoder-decoder architecture where the input data are downsampled to a lower spatial dimension and then upsampled back to its initial spatial resolution to allow a pixelwise classification. This architecture has shown good results with small data sets and strong use of data augmentation methods.

The modified U-Net architecture (Nowaczynski et al. 2017) used for this project is only build of convolutional, transposed convolutional, ReLU, batch normalization and maxpool layers (see section 2.1.2). There are no fully-connected layers in this architecture, since the upsampling part is achieved using only transposed convolutional layers. The structure of the model can be seen in Figure 13. The main differences between the original U-Net architecture and our modified U-Net model are changes in the input image dimensions and various hyperparameters of layers, for instance the number of kernels per convolutional layer.

The architecture is mainly build of BN_CONV_RELU and BN_UPCONV_RELU blocks (Figure 13). The BN_CONV_RELU block is used during the whole model and is composed

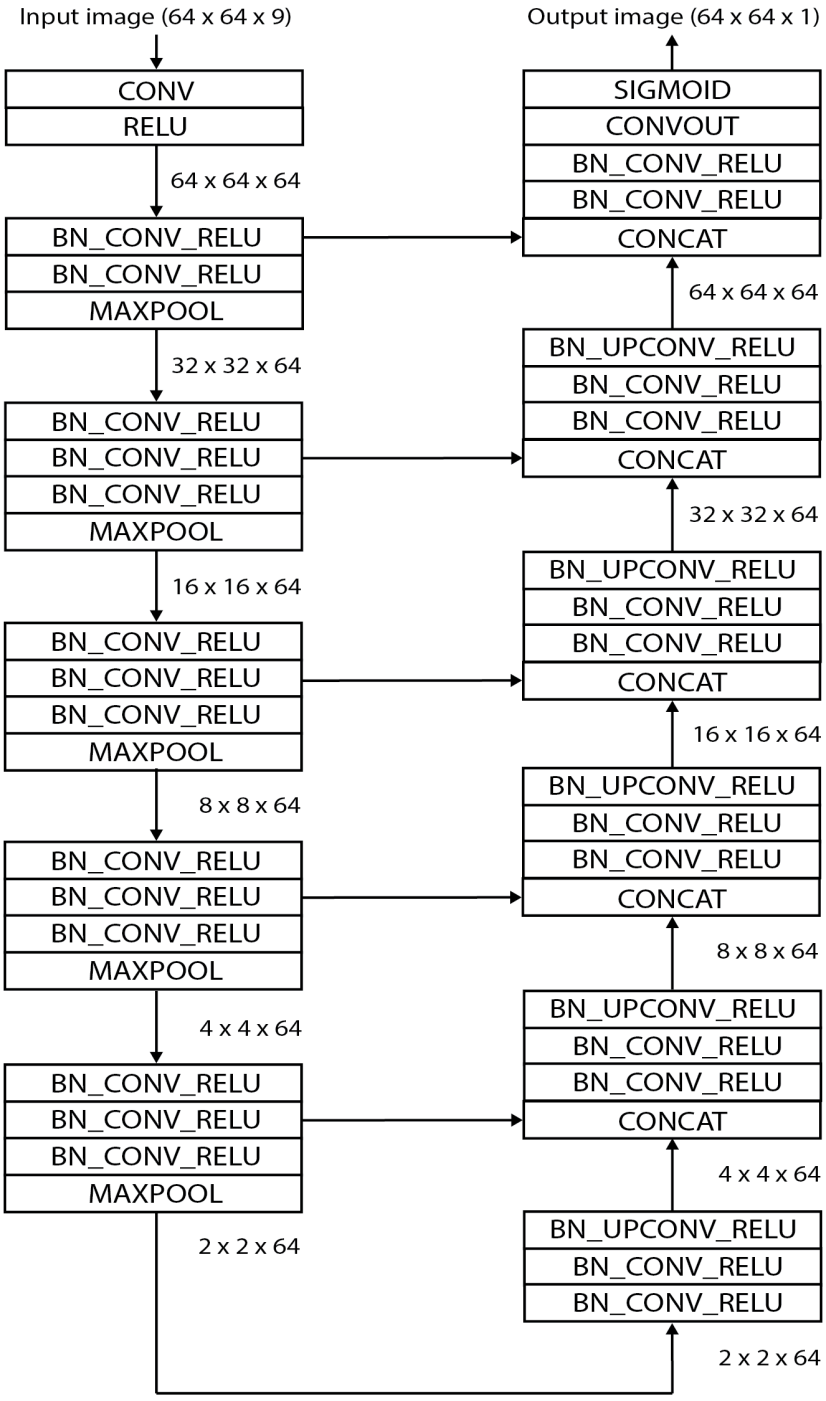


Figure 13: Modified U-Net architecture

of:

- a batch normalization layer,
- a convolutional layer with 64 kernels of size = 3×3, stride = 1 and padding = 1,
- a ReLU layer (activation function).

The BN_UPCONV_RELU block is only used during the upsampling phase (right side of the model) and is composed of:

- a batch normalization layer,
- a transposed convolutional layer with 64 kernels of size = 3×3, stride = 2 and padding = 1,
- a ReLU layer (activation function).

The U-form of the architecture stands for the downsampling (left side) and upsampling phases (right side) of the model (Figure 13). The downsampling is achieved with five max-pooling layers with kernel size = 2×2, stride = 2 and padding = 0. As a result of the hyperparameter choice of the pooling layer, the output size (height and width) of these layers is always half of the input size. On the other side of the model, transposed convolutional layers are used to upsample the data back to its initial size in order to be able to determine the label for every pixel of the input batch.

It is important to note that, during the upsampling phase, the output of the transposed convolutional layer is consistently concatenated with the same size data from the downsampling phase. This can be observed in Figure 13 pictured by horizontal arrows and the box CONCAT. The concatenation is executed on the third dimension. This allows the model to combine the more detailed information from the downsampling phase with the much more generalized information of the upsampling phase.

The CONVOUT layer is a convolutional layer with 2 kernels of size = 1×1, stride = 1 and padding = 0 followed by a sigmoid function (activation function). The output of the U-Net model is a vector with the class score between 0 and 1 of each pixel ("No Cocoa"= 0/"Cocoa"= 1).

3.1.2 Training Details

The model is trained with multispectral images from the Sentinel-2 satellites. The class score vector resulting from the U-Net architecture is then transformed into a loss score using a Softmax cross-entropy function. The L2 loss of the weights is then added to the cross-entropy loss in order to prevent overfitting (Regularization). The model is trained using an Adam optimizer minimizing the total loss.

The hyperparameters of our network and their corresponding default values are:

- **Number of training epochs** is adjusted in respect to the amount of batches of the input data set for the respective training session.
- **Learning rate** is set to a standard value of 10^{-5}.
- **L2 regularization** is set to a standard value of 10^{-2}.
- **Mini-batch size** is the number of batches trained at once and thus used to compute the training updates. This parameter is set to 32.

3.2 Ground Truth and Data Split

One of the most important components when performing a good classification is to have an accurate and complete ground truth of the analyzed area. The information of the scenery is often difficult to acquire since it mostly has to be manually assembled. As a consequence, this part of the project data is usually the most incomplete and inaccurate part of the information used during segmentations. This problem also affects our project and has been carefully handled.

In order to minimize the probability of incorrect labeling and, by that, of interfering with the learning process of the convolutional neural network, the images are labeled in three different classes:

- **Cocoa**
- **Background**
- **Uncertain**

This partition using a third class should prevent to falsely label cocoa pixels with the label "background". It is important to note that the pixels labeled as "uncertain" will not be used during training, since these would completely confuse the learning process of the network.

Since data are sparse in our project, we decided to split our data set in only two independent data sets: a training data set and a validation data set. The validation data set will not only be used for the fine-tuning of the hyperparameters but also for the final test. The validation data set will be independent enough to be a good reference for our experiments since it will just have been sparsely in contact with the model before the final test.

3.2.1 Ecuador

In Ecuador the position of tree different farms was available (Figure 14); two in the region of "Los Ríos" and one in "Guayas" (near the cities of Quevedo and Velasco Ibarra). Their position was measured using a GPS antenna.

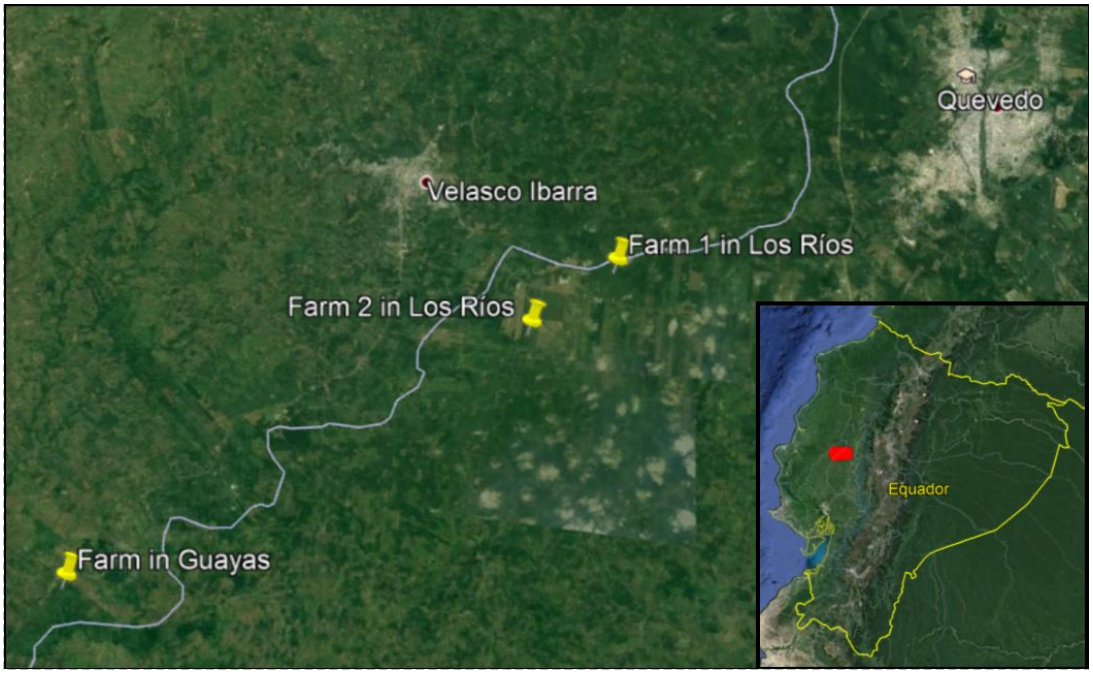


Figure 14: Farms in Ecuador

We used Google Earth images to locate and map the farms, since the GPS points just indicated the position of the farm entrances. This mapping task is doable since in Ecuador cocoa is mostly planted in full sun farms, easy to recognize on satellite imagery. Unfortunately, all the Google Earth orthophotos from the area of the farm in "Guayas" were blurred. Therefore, it was impossible to correctly identify and, thus, map the farm. We decided to use only the other two farms in the region of "Los Ríos", which are next to each other. Since the orthophotos of that region were very clear, we could easily locate, map and create shape-files of the ground truth of both farms. The first farm has an area of 150 hectares and the second one an area of 125.

To correctly train the farms in Ecuador, we divided the area of the two selected farms into two spatially separated areas, a big "Training Area" and a smaller "Validation Area". This subdivision is shown in Figure 15.

3.2.2 Ghana

In Ghana and Ivory Coast, we have a data set of approximately 175 farms, that have been mapped between April 2016 and December 2017. These farms are well distributed over both countries, mostly between the latitudes of 5°N and 7°N, and cover the regions of Eastern (GH), Ashanti (GH), Western (GH), Comoé (CI), Lagunes (CI), Bas-Sassandra (CI), Sassandra-Marahoué (CI) and Montagnes (CI). As for the farms in Ecuador, the position of the farms was measured with a GPS-antenna.

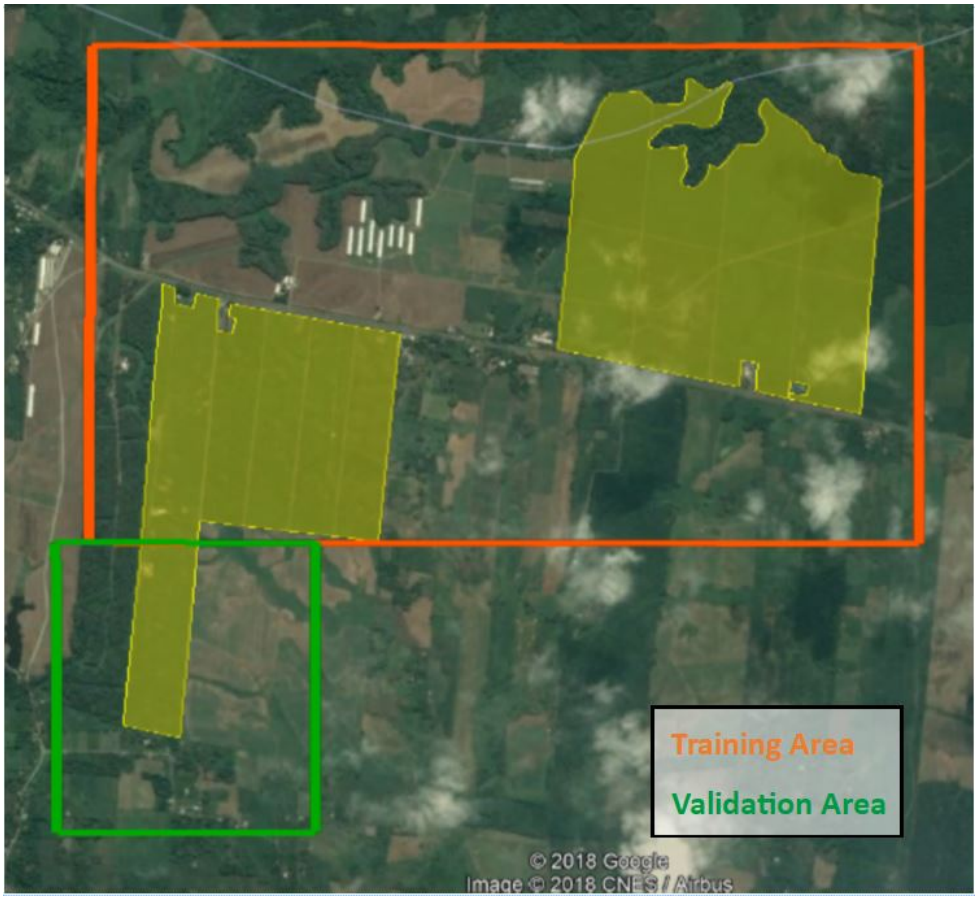


Figure 15: Farms in Los Ríos

The big difference between the farms in Ecuador and the ones on the African continent is the difficulty to recognize the size and form of the farms from the Google Earth orthophotos. This is due to the fact that most of the Ghanaian and Ivorian farms are still agroforests and, as a consequence, are difficult to visually separate from other crops or even normal forest.

Fortunately, we also had GPS positions of the borders of 15 farms. They are well distributed through Ghana. In order to prevent mislabeling, we decided to nearly only use the bordered farms. But, since after discarding the really small farms (less then 10 pixels in the Sentinel-2 image), only eight farms were left, we decided to add four farms from the bigger data set in order to increase the amount of training information. Since the form of these farm had not been measured, we just labeled a small area (10×10 pixels per farm) as "cocoa" and left the rest (forest-like area) to the "uncertain" class. In this way, we added some more vital information to the Ghanian data set.

(a) GPS points of the farm

(b) Ground truth of the farm

Figure 16: Farm ER20 in East Ghana

3.3 Preprocessing

3.3.1 Procedure

The preprocessing procedure has been computed equally for all the Sentinel-2 images used during this project:

- First, the Sentinel-2 imagery is downloaded from the "Copernicus Open Access Hub", where the data of all Sentinel missions is publicly available. At this point, the image is in the product type **Level-1C**. The worldwide product type **Level-2A** will be directly available at the end of 2018.
- The **Level-1C** image is then processed to the product type **Level-2A** using the Sentinel Application Platform (SNAP). This Toolbox can also be used to control the quality of the images, perform band arithmetics and create masks such as cloud masks.
- A **GeoTIFF** file is then created from the Level-2A image. The GeoTIFF format, being a special form of the TIFF format (Tagged Image File Format), is a metadata standard allowing the storage of georeferencing information in high spatial resolution.
- In a next step, the GeoTIFF file is clipped with the Shapefile of the ground truth (labels) to one single raster graphic, in order to obtain an array with the pixel information of the different bands and its respective ground truth class.
- At this point, we only select the bands with 10 m and 20 m spatial resolution: B2, B3, B4, B8, B5, B6, B7, B11 and B12.
- The array is then subdivided into **batches** of a size 64×64 pixels. The batches are created using different types of data augmentation methods such as overlapping (using a fixed stride) and rotation (90, 180 and 270 degrees).

- Mean subtraction and normalization are applied to these batches. The mean and the standard deviation are calculated separately for every band of the training data set and is then applying to the entire data set.

3.3.2 Selection of the Imagery

The images used for this project have been chosen considering the coverage area of the image, the main crop season (September - February) and the percentage of cloud cover when the data was acquired:

- **Ecuador**: We used imagery of the position *MPU* (Sentinel-2 code for the different measurement positions) that covers the lower center part of the country
  - 8th of November 2017: cloud coverage of 31.02%
- **Ghana** The Ghanian observation area, being larger than the one in Ecuador, is distributed over four different Sentinel-2 images with the positions *NWM*, *NXN*, *NYM* and *NYN*. Additionally, we choose four different imagery dates to increase the amount of image data and have the possibility to use the temporal information of the cocoa growth.
  - 23th of December 2017: cloud coverage between 28.85% and 84.17%
  - 2nd of January 2018: cloud coverage between 0.00 % and 20.19%
  - 12th of January 2018: cloud coverage between 5.81% and 14.62%
  - 27th of January 2018: cloud coverage of 0.00%

3.4 Analysis of the Learning Process

As mentioned earlier, convolutional neural networks are powerful classifiers that achieve high accuracies in many different tasks. However, it is still difficult to determine which are the factors and properties of the multispectral images that a convolutional neural network uses to recognize and segment cocoa. In this section, we will analyze some components of the convolutional neural networks and multispectral images in order to better understand which aspects of the cocoa crop influence the classifier.

3.4.1 Importance of the Different Bands

In this section, we will determine which bands of the Sentinel-2 multispectral images have pivotal information for the segmentation process. Therefore, we will test different combinations of bands and compare the test results of the different trainings:

- **RGB Bands** (B2, B3 and B4): This is equivalent to taking an image with a normal three band camera. If this combination reaches comparable results, then we will be able to say that the use of multispectral images is unnecessary for this task.

- **Dropping each band separately**: This will reveal if one of the nine bands is indispensable for cocoa segmentation.
- **NDVI Band** (B4 and B8): The normalized difference vegetation index is a well-known index often used for vegetation segmentation.
- **RGB-NIR Bands** (B2, B3, B4 and B8): Combined RGB and near infrared cameras are nowadays a popular tool for image segmentation.

Beside these combinations, we will also further analyze bands that have shown some significant relevance during these experiments.

3.4.2 Spectral Signature of Cocoa

As outlined in section 2.2, spectral signatures are often used to differentiate between different surface objects and show clear visible differences between various plant types. Therefore, we will analyze the spectral signature of cocoa and nearby elements such as forest or other plants. Further, we will investigate the different correlations between these objects and their spectral signature.

- In Ecuador, we will compute the spectral signature of cocoa, forest, bare soil and a neighbor crop field.
- In Ghana, we will compute the spectral signature of cocoa from two different farms for three different dates.
- Last, we will compare the spectral signatures of cocoa between Ecuador and Ghana. Additionally, we will also compute the spectral signature of two Ghanian forests (Mamiri Forest Reserve and Boin Tano Forest Reserve) in order to obverse the difference between tropical forests and agroforests.

These three analyses will give us information about the distinctness between cocoa and its nearby areas, the evolution of the cocoa crop and the difference between cocoa in full sun farming (Ecuador) and in agroforests (Ghana).

To compute the spectral signature of an object, we will cut out a certain area covered by that object and calculate the mean value for each band reflectance values separately. This results in one mean value per band. For the spectral signature of cocoa in Ghana (third analysis), we will additionally average the reflectance values over the three data acquisition dates.

3.5 Cocoa Segmentation in Ghana

Since the situation in Ghana with the agroforests farming and the small amount of suitable information is particularly difficult, we approach this issue separately in this section.

The data set used to train the Ghanian farms is sparse (12 small farms) and really unbalanced (proportion between cocoa and the rest of the image is smaller then 1/30). These two factors, also being present in the Ecuadorian data set, are more extreme in the Ghanian data set. The sparsity and imbalance of data are two big problems while training a convolutional neural network because the model is fed with a very small amount of incomplete information. We will try to use different techniques to counteract these two issues.

3.5.1 Temporal Data

The first method used to deal with the problem of the limited quantity of data is the use of multispectral images from different dates. These temporal data give to the network extra information about the growth process not only of the cocoa crop but also of the other plants surrounding them, such as shading trees.

Therefore, as mentioned in section 3.3.2, we selected Sentinel-2 images from four different days in December 2017 and January 2018. The multispectral images of the same area are then stacked over one another on the band dimension, in order to obtain an input image of the size $64 \times 64 \times 36$. This change of the input dimensions does not alter the architecture (see section 3.1.1) or the hyperparameters, still set to the standard parameters defined in section 3.1.2. By following this method, we will have multispectral images of certain areas with the information of the development of the farms during December and January.

3.5.2 Unbalanced Data Set

Methods that artificially create, select and delete information of the original data set in order to diminish the imbalance between classes are:

Data augmentation A method described in section 2.1.2 to increase the amount of data in a data set is data augmentation. For this project, we only used the translation and rotation methods. We implemented the translation by creating overlapping 64×64-pixel batches from the original multispectral images. Then, these batches are rotated by 90, 180 and 270 degrees to quadruple the amount of batches. These two augmentation methods have been applied not only on the Ghanian but also on the Ecuadorian data set.

We did not use scaling as a data augmentation method since the Sentinel-2 imagery is always taken from the same satellite altitude and this would have created useless information for the convolutional neural network.

Undersampling We used the method of undersampling in different degrees to compare the effects of this technique. Undersampling consists in setting a minimum amount of cocoa per batch and thus eliminates the batches with a lower amount of cocoa. As a consequence,

the proportion between cocoa, background and uncertainty in the data set will prominently shift toward cocoa.

We tested different values of the minimal amount of cocoa per batch. For the batch of size 64×64, we computed the undersampling for the values 100 and 200, representing a minimum of 2.5% and 5% of cocoa per batch.

Batch size Another possibility to counteract the imbalance of our Ghanian data set is to reduce the batch size (height and width) from 64 to 32. This reduction will decrease the area covered by a batch by four and thus, combined with selective undersampling, create a multitude of batches with a higher percentage of cocoa.

This method will also reduce the amount of information of the surrounding of the farm fed to the convolutional neural network. Consequently, the results will give us an indication about the importance of the environment of the farms for cocoa segmentation.

4 Results and Discussion

4.1 Evaluation

The most common way of evaluating the results of a binary classification problem is the use of the concepts of Recall, Precision, Accuracy and Intersection over Union. These four auxiliary variables visualize the correctness of our predictions. To define them, we need to introduce first the notions of "True Positive", "True Negative", "False Positive" and "False Negative".

Four cases arise in a binary classification. True Positive occurs when the condition ("there is cocoa") is true and the prediction is positive ("predicts cocoa"), True Negative when the condition is true and the prediction is negative ("predicts no cocoa"), False Positive when the condition is false ("there is no cocoa") and the prediction is positive and False Negative when the condition is false and the prediction is negative. These four cases can easily be assembled in a so-called **confusion matrix** (Table 1).

		Prediction outcome		
		p	**n**	**total**
ground truth	**p′**	True Positive	False Negative	P′
	n′	False Positive	True Negative	N′
	total	P	N	

Table 1: Confusion matrix

From these different auxiliary variables we can compute many ratios. The most used ones are:

- **Accuracy** is the most intuitive of these variables. It represents the ratio between the sum of all the correct predictions and the total population:

$$Accuracy = \frac{TruePositive + TrueNegative}{TruePositive + FalsePositive + FalseNegative + TrueNegative}$$

- **Recall** (Kent et al. 1955) is the ratio between the True Positive and the sum of the positive conditions. It indicates the part of the positive conditions that has been

correctly predicted:

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

- **Precision** (Kent et al. 1955) is the ratio between the True Positive and the sum of the positive predictions. It characterizes the predictions that are correct and thus actually useful:

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

- **Intersection over Union** (IoU), also known as Jaccard index (named after the Swiss botanist Paul Jaccard), is the ratio between the intersection and the union of the predictions and the conditions (Figure 17). This metric, being a good a midpoint between Recall and Precision, will be used as the main comparison metric:

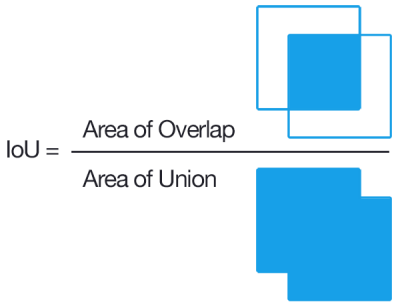


Figure 17: Intersection over Union

4.2 Ecuador: the case of full sun farms

The data set used to train the Ecuadorian farms is processed from a Sentinel-2 image from the 8th of November 2017. The two farms are full sun cocoa plantations and will therefore be a good indicator for the usability of this method on this kind of farms.

Using data augmentation (Rotation and Translation), we obtain a training data set of 8190 batches and a validation data set of 1027 batches. The learning rate is set to 10^{-5} and the L2 regularization to 0.01. The U-Net model is trained during 18 epochs, corresponding to circa 4.600 training iteration with mini-batch size 32 and lasting 5 min and 50 sec.

The evaluation of the results is shown in Figure 18. At the end of the training process, we computed the final metric on the validation data set:

- Final recall: 93.0%
- Final precision: 98.8%

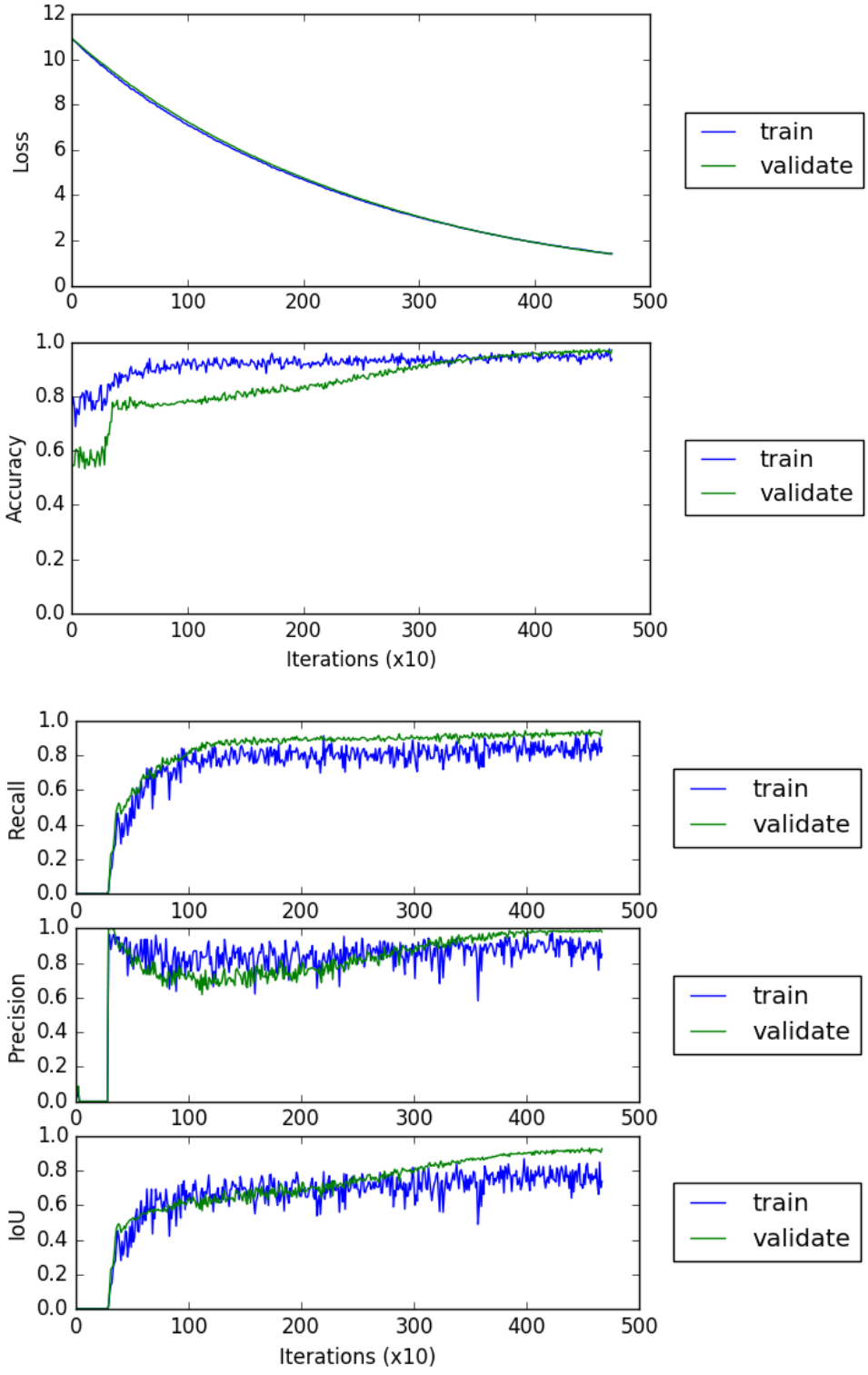


Figure 18: Results of the Ecuadorian training and validation data set

- Final intersection over union: 92.0%

These are impressive results considering the limited amount of training data. We notice that the precision of the predictions is nearly 100%. This means that the convolutional neural network rarely assigns cocoa to a pixel labeled as background in the ground truth. This error is known as false positive error or type I error.

When examining the loss and accuracy curves (Figure 18 (a)), no signs of overfitting can be observed since the training and the validation curves nicely converge to the same values and no significant drifting apart occurs. On the intersection over union curve, we can see that the intersection over union of the validation reaches a much higher value than the training. This is probably due to fact that the training area (Figure 15) contains more complex shapes and a higher diversity of plants.

In Figure 19, the visualization of the RGB image, the ground truth and the prediction of a batch from the validation data set are shown. A visualization of the superposition of the ground truth and the prediction can be found in Figure A.1. We observe that the convolutional neural network has correctly recognized the longitudinal form of the farm. However, although identifying the rough shape of the farm, the classifier has some difficulties with the cloud on the upper part of the validation area. This is due to the fact that multispectral images do not contain any information of the Earth's surface covered by clouds, since the multispectral camera only measures the reflectance of the cloud. Hence, we recommend in a further step to incorporate a method to detect clouds and handle them separately from the rest of the data set.

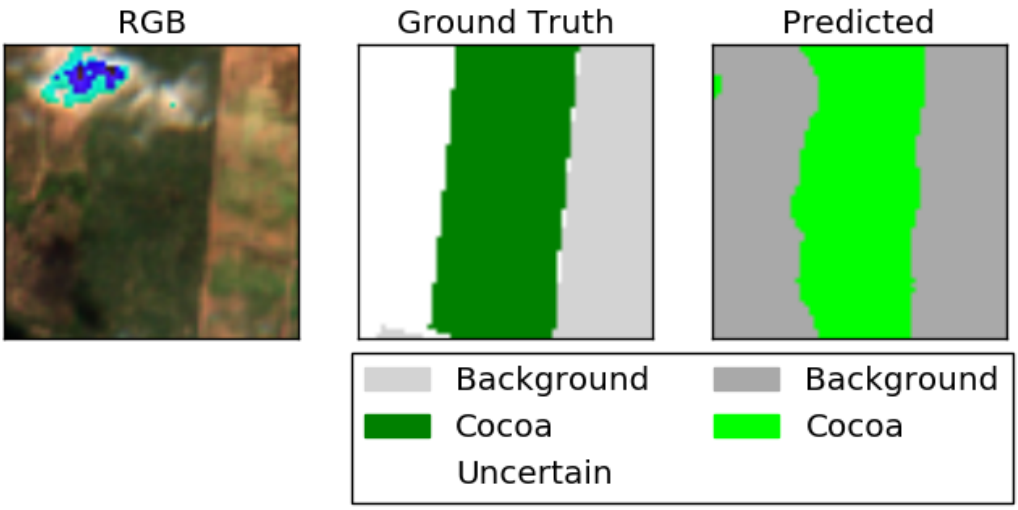


Figure 19: Visualizations of a validation batch

We can also observe in Figure 19 that the greatest part of the incorrect predictions happen on the border of the farm. Thus, we suggest the possibility of integrating edge detection

methods to improve the results on the transitions between "cocoa" and "background".

Furthermore, it is important to note that the results of this section are the training outcome of a data set with only two, very similar farms. Therefore, these results have to be validated with a larger data set containing more farms distributed trough the whole country before being able to confirm the viability of this method on all full sun farms.

4.3 Analysis of the Learning Process

In this section we will analyze which aspects and properties of the convolutional neural network and the multispectral images are decisive during the process of cocoa segmentation. This will give us a better understanding of the learning process and, thus, provide the necessary knowledge to optimize the different steps and sub-procedures of this method. The detailed explanation of the different analyses can be found in section 3.4.

4.3.1 Importance of the Different Bands

We compared the results of the trainings (Table 2) with different combination of bands from the Ecuadorian Sentinel 2 image. The first run with all the bands (B2, B3, B4, B5, B6, B7, B8, B11 and B12) has an intersection over union of 92.0%. This will be the reference point to compare the other runs.

In a first step, we analyzed the importance of multispectral images by only testing the RGB bands. We observe that the RGB-bands alone (Figure 2: IoU = 45.3%) are not enough to properly detect cocoa. The cocoa predictions were randomly distributed over the validation area and did not have the characteristic form of the validation (Figure 16). This means that the use of multispectral images and bands further than the visible electromagnetic waves are important for cocoa segmentation. This statement is further confirmed by the result of the training run without the visible bands (Figure 2: IoU = 92.7%); the intersection over union being comparable to the reference run. This can be explained by the fact that RGB is mostly used for the detection of textural features and structural patterns. This is quite unnecessary because the image resolution of the Sentinel-2 data is 10×10 m and no texture should be recognizable in the images since there is more than one tree per image pixel.

Band B11 (1610 nm) is important, since the run without band B11 only reached an intersection over union of 83.1%, nearly 10% less than the reference run. The other bands do not show significant differences compared to the reference run when dropping them out during training.

It is interesting to observe that the traditional methods such as NDVI (B4 and B8) and RGB-NIR (B2, B3, B4 and B8) show comparable results to the reference run with all the bands. The differences in the results between these two methods and the reference run is due to the accuracy of the predictions in the border region of the farm.

	Accuracy	Recall	Precision	Intersection over Union
All Bands	96.7%	93.0%	98.8%	92.0%
Only RGB	72.3%	51.6%	78.9%	45.3%
Without RGB	96.7%	95.0%	97.4%	92.7%
Without B2	96.6%	95.3%	96.6%	92.2%
Without B3	96.1%	91.7%	98.7%	90.6%
Without B4	96.1%	92.1%	99.0%	91.3%
Without B8	95.6%	93.4%	96.2%	90.0%
Without B5	96.3%	92.4%	98.9%	91.5%
Without B6	96.8%	95.1%	97.4%	92.8%
Without B7	97.1%	94.3%	99.3%	93.7%
Without B11	91.9%	83.5%	99.4%	83.1%
Without B12	96.1%	92.1%	99.6%	91.7%
NDVI Bands	93.8%	89.1%	95.4%	85.4%
RGB and NIR	95.2%	90.8%	97.2%	88.5%
Only Band B8	92.3%	87.2%	94.8%	83.2%
Only Band B11	87.3%	82.0%	83.9%	70.9%
Only Band B12	90.5%	86.7%	90.1%	79.2%
Bands B8, B11 and B12	95.0%	90.0%	98.2%	88.5%

Table 2: Band Analysis

From these results, we can deduce the importance of the bands B8, B11 and B12. This follows the poorer results of the training without band B11 and the good results of the NDVI and RGB-NIR trainings. When training these bands alone, the convolutional neural network reaches higher intersection over union than for the RGB training. All three single-band combinations segmented the general form of the farms, but poorly segmented their border areas.

Furthermore, it is interesting to note that the combination of bands B8, B11 and B12 only gives a slightly lower result (Figure 2: IoU = 88.5%) than the reference run. This is not surprising since bands B11 and B12 (short-wavelength infrared) are good indicators of the moisture content of the soil and vegetation and are therefore often used to differentiate between different types of vegetation.

4.3.2 Spectral Signature of Cocoa

As explained in section 3.4.2, spectral signatures carry information about the reflectances of different objects of the Earth's surface. We will therefore try to find a relation between the different land covers and their spectral signature.

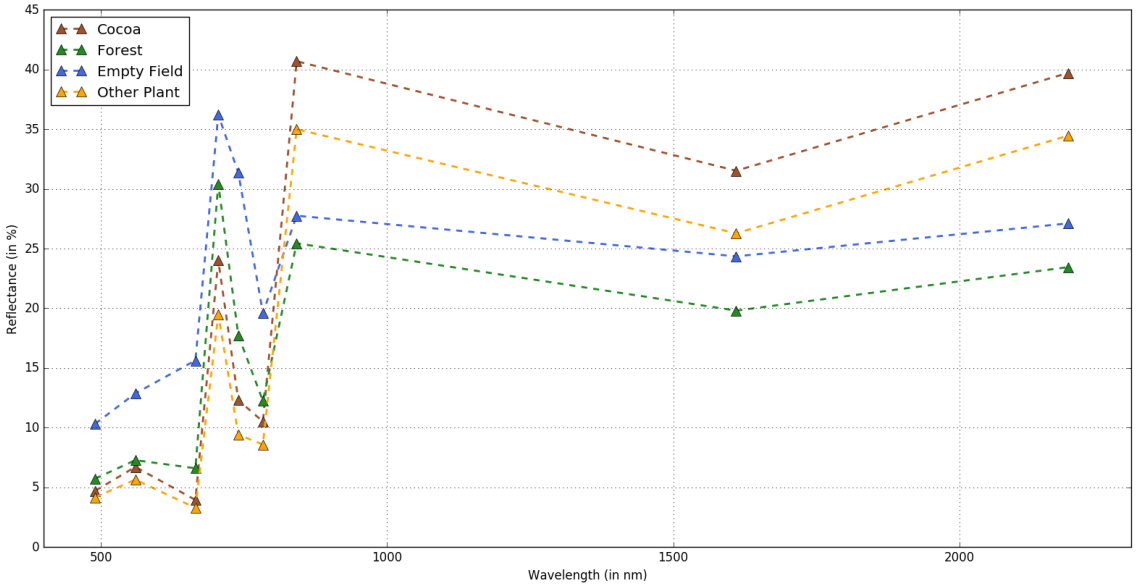


Figure 20: Spectral signatures of cocoa, forest, bare soil and another crop type in Ecuador

Cocoa in Ecuador First, we analyzed the situation in Ecuador calculating the spectral signatures of cocoa, forest, an empty field and an unknown crop cultivated beside the cocoa farms. The reflectances of the multispectral bands are illustrated as triangles in Figure 20.

First, we observe that, while in RGB (first three bands) bare soil differentiates itself very well from cocoa, forest and other crops have a similar RGB signature compared to cocoa. The difference between these types of vegetation gets more conspicuous when observing the near infrared (700 nm−1400 nm) and the short-wavelength infrared (1400 nm−3000 nm).

Furthermore, bands B11 (1610 nm) and B12 (2190 nm) being good indicators of the moisture content of surface objects, we deduce that the cocoa fields and the neighbor crop are irrigated since these two bands have higher values compared to the forest or the empty field. This is not a surprise as the Ecuadorian farms are full-sun plantations that need strong care and thus also intensive irrigation.

Cocoa in Ghana In Ghana, we computed the spectral signatures of two farms (ER6 and ER20) for three different dates in January (Figure 21).

It is interesting to note that, while there is a small difference between the spectral signature of the two farms on the same day, this difference is nearly constant for the three

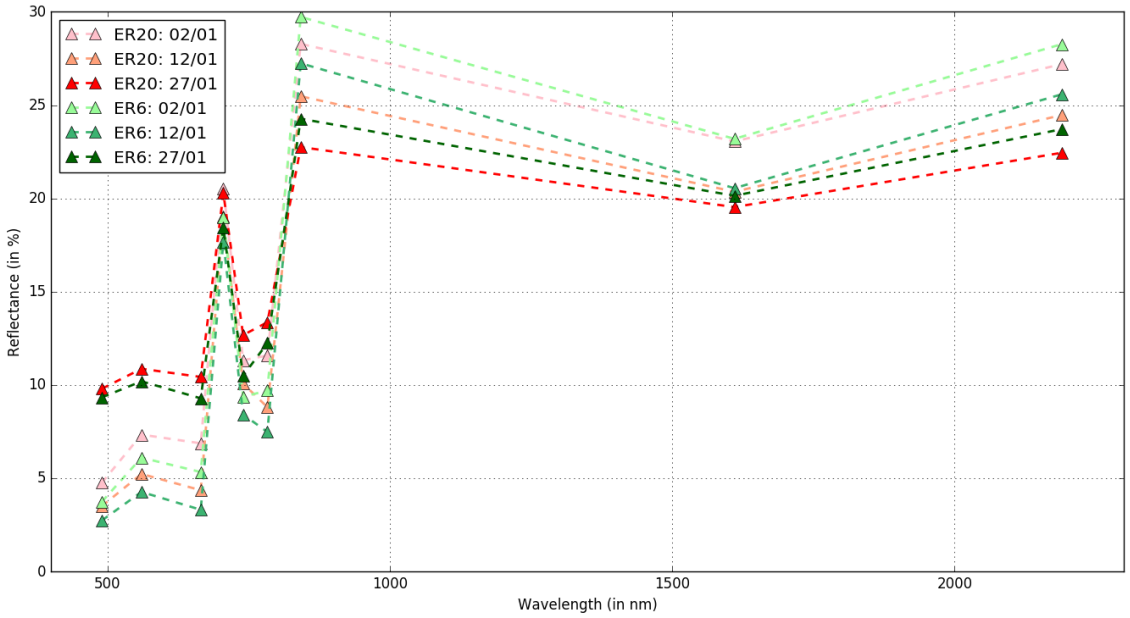


Figure 21: Spectral signatures of two farms in Ghana on three different days

January dates. We can thus deduce that the farms have a similar growing process that can be learned by the convolutional neural network and, therefore, used during segmentation when training with temporal data (Section 3.5.1).

A significant evolution with time occurs in the visible part of the electromagnetic spectrum. This indicates a color change of the agroforest. This should mostly correspond to the development in association with the shading trees covering the farm. Hereby, we can say that the growth and development of the shading trees are a potential key in the process of detecting cocoa.

Furthermore, a diminution of the water content in the cocoa plants can be observed, as band B11 and B12 decrease during January. This is probably due to the fact that the rainy season ends in November and that January is on average the month with the lowest amount of precipitations in Ghana.

Comparison between cocoa in Ecuador and Ghana In order to better understand the relationships, similarities and differences between full sun cocoa, agroforest cocoa and normal rain forest, we computed the spectral signature of all these types of vegetation (Figure 22).

The biggest differences between the spectral signatures is noticeable in the bands B8 (842 nm, third from right), B11 and B12. First, a major gap can be observed between the

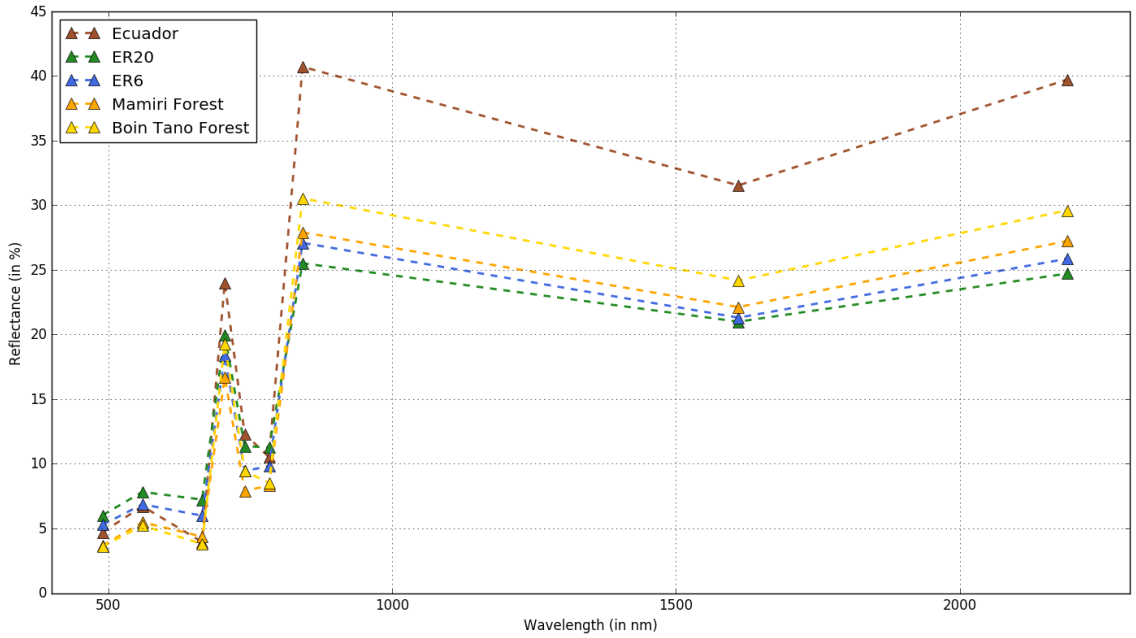


Figure 22: Spectral signatures of two Ghanian farms, the Mamiri Forest Reserve, the Boin Tano Forest Reserve and an Ecuadorian farm

Ecuadorian cocoa and the rest of vegetation types. As mentioned above, this might be due to the fact that Ecuadorian farms are irrigated and Ghanian are not. This shows the discrepancy between the two countries and the delay in the development of farming techniques that the majority of African countries have compared to South American ones.

In Ghana, the difference between the agroforests and the forest is minor compared to Ecuador. This is not surprising considering that agroforests often contain the same trees as a normal forest functioning as shading trees for the cocoa crops. Nevertheless, some small differences between the spectral signatures of forest and cocoa agroforest can be observed.

These small differences, combined with the parallel growing process of cocoa when using temporal data, should contain enough spectral distinctness for the task of cocoa segmentation in agroforests. Furthermore, the importance of the bands B8, B11 and B12, deduced in section 4.3.1, can clearly be seen in the three analyzed figures in this section.

4.4 Ghana: the case of agroforestry farms

The Ghanian data set is composed of twelve small farms and is, because of the small amount of available data, really unbalanced. These farms are agroforests, as explained in the introduction (section 1), and will therefore be a good indicator of the feasibility of our

method for this kind of cocoa farms.

Since the situation is more complex than for the full sun farms, we introduced some methods in section 3.5 to counteract the issues of the sparsity and imbalance of the data set. In this section, we will discuss the results and effects of these methods.

First, we will compare and analyze the use and outcome of temporal data and undersampling. Here we understand by non-temproal data the non-stacked and by temporal data the stacked batches (section 3.5.1) of the four different dates in December 2017 and January 2018 for which data are available.

- **Undersampling on Non-temporal Data:**

Minimal amount	Percentage of cocoa	Accuracy	Recall	Precision	Intersection over Union
0	$2-3.5\%$	56.5%	0.0%	/	0.0%
100	5%	45.2%	0.0%	/	0.0%
200	7%	73.7%	47.4%	99.2%	47.2%

Table 3: Results of the final validation using undersampling on non-temporal data

‘

- **Undersampling on Temporal Data:**

Minimal amount	Percentage of cocoa	Accuracy	Recall	Precision	Intersection over Union
0	2%	53.2%	0.0%	/	0.0%
100	5%	75.7%	61.1%	99.7%	58.2%
200	7%	49.8%	44.2%	99.6%	44.1%

Table 4: Results of the final validation using undersampling on temporal data

First, it is clear that the use of undersampling is vital when training such a sparse and unbalanced data set. This is clearly recognizable in the first rows of Tables 3 and 4 since in both cases training without undersampling did not predicted any cocoa (IoU = 0.0%).

Some increase in the training stability seems to occur when training with temporal data. This is shown by the fact that the temporal data already gives cocoa predictions with low undersampling while non-temporal data is still useless. For non-temporal data, the minimum number of cocoa pixels per 64×64 batch has to be set to at least 200 to show cocoa predictions.

It is interesting to note that all the training sessions with cocoa predictions have led to extremely high precision values (over 99.0%). This was already the case in Ecuador (section 4.2). This shows again that the convolutional neural network does nearly not predict cocoa on ground truth background (false positive error or type I error). Thus, it is clear that the increase of the recall is the more complex task during cocoa segmentation.

Furthermore, it must be said that all the trainings with temporal data have resulted in final training accuracies of approximately 95.0%. This is a perfect example of the accuracy paradox introduced in section 2.1.2 where the accuracy is extremely high but the values of the other evaluation metrics are considerably lower. This indicates that, even after applying strong undersampling, our data set continues to be quite unbalanced and therefore less-than-ideal for the given task.

In Table 4, we can observe a slight decrease of the intersection over union value when increasing the amount of undersampling. This might be due to the fact that undersampling eliminates a lot of batches and thus also a great deal of relevant information that the convolutional neural network could need during the segmentation process, such as information about the borders and the surroundings of the farms.

Batch size In order to increase the percentage of cocoa per training batch, we proposed in section 3.5.2 to compute batches of the size 32 combined with the use of undersampling.

The validation results quickly reached intersection over union values over 80.0% when applying undersampling. Along with it, the percentage of cocoa per training mini-batch reached values between 15.0% and 25.0%. Therefore, the percentage of background drops considerably, since the areas directly surrounding the farms are mostly labeled as uncertain.

Furthermore, the predictions were not only distributed over the ground truth cocoa but also over the majority of the ground truth uncertain. This outcome being suspicious, we decided to validate the trained model on Sentinel-2 imagery of the forest area of the Höggerberg in Switzerland as we were sure not to find cocoa in this area of the world. The Swiss validation of the trained model predicted non-existent abundance of cocoa in the forest of Zurich.

Therefore, we deduced that the model generally predictes forest areas. This is surely caused by the missing forest information in the background label when reducing the batch size. Since these results look far from good, we decided to continue working with the batch size 64. Nevertheless, we assume better results are possible with the batch size 32, in case more data or better information on the farm surroundings are available.

Analyzing the best choice of hyperparameters We will further discuss the best results achieved on the Ghanian data set. The set of hyperparameters chosen for this training run is:

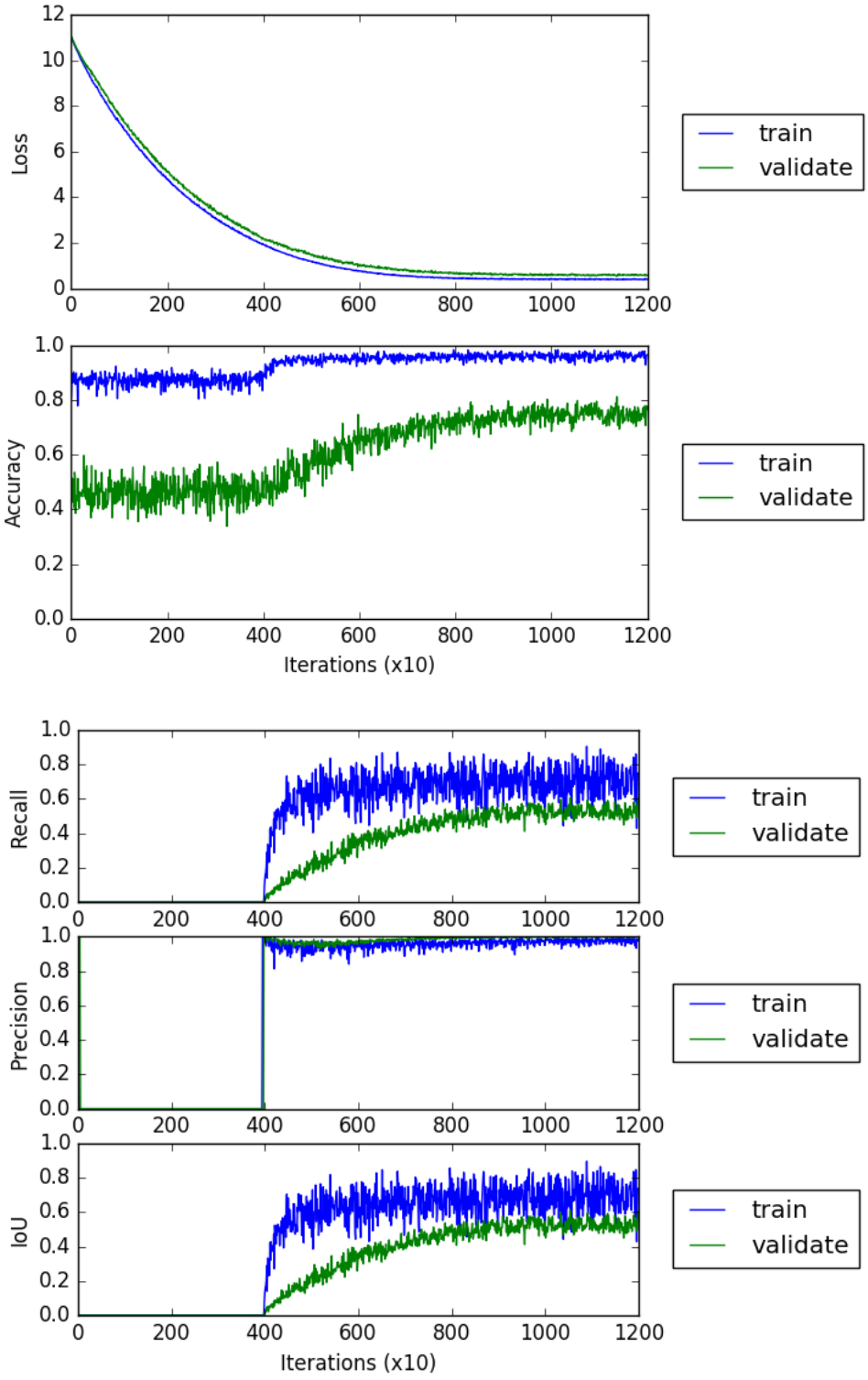


Figure 23: Results of the Ghanian training and validation data set

- Use of temporal data,
- Undersampling with 100 pixels as minimal amount of cocoa per batch,
- Batch size of 64.

The evaluation of the results is shown in Figure 23 and the final metric can be found in the second row of Table 4.

The loss and accuracy curves in Figure 23 indicate that overfitting occurs to some extent during training. This is probably due to the small amount of data. However, the recall and intersection over union curves seem to nearly converge and, thus, we can say that the overfitting is not excessive and can be neglected.

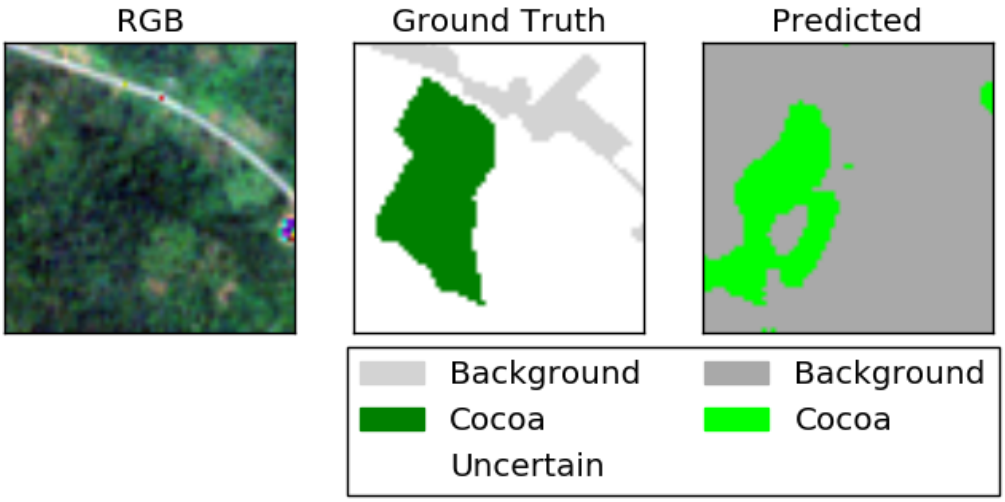


Figure 24: First visualizations of a validation batch

Figures 24 and 25 are visualizations of the RGB image, the ground truth and the prediction for the validation farm of the Ghanian data set. A visualization of the superposition of the ground truth and the prediction can be found in Figures A.2 and A.3. In Figure 24, we perceive that the convolutional neural network finds the rough position and form of the farm. However, there is still a hole in the middle of the farm and the predicted borders do not match the ones on the ground truth. These imperfections can have two opposite causes. On one side, we cannot estimate how accurate the GPS points that form the ground trut, have been measured. On the other, the amount of data given to the convolutional neural network is below the usual amount of data used to train this kind of methods. Furthermore, since agroforests can be as diverse as normal forests, it can be that we do not have enough farms covering all possible types of vegetation found in agroforests.

Figure 25 shows the same farm as Figure 24 just shifted a bit to the left. In this visualization, we see that the convolutional neural network predicts some cocoa in the ground

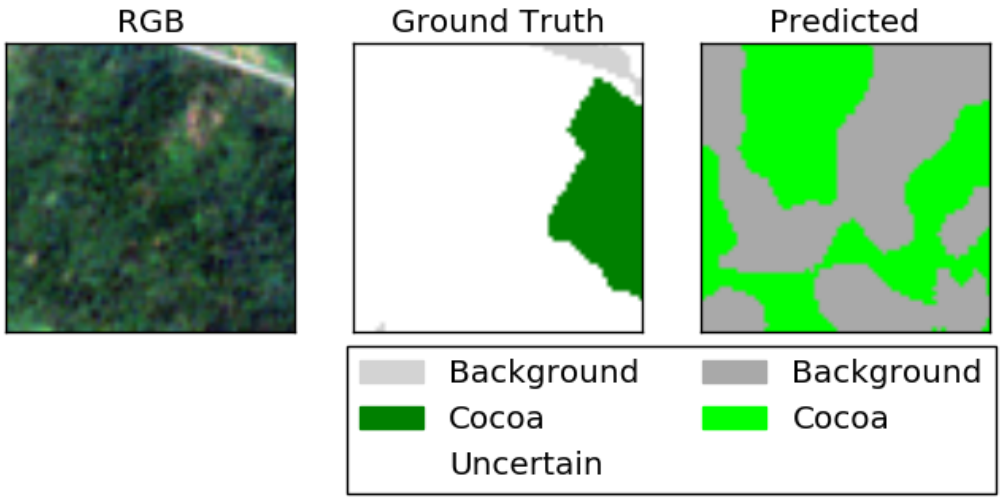


Figure 25: Second visualizations of a validation batch

truth uncertain area beside the ground truth farm. Since we do not have any information about the area around the farms, we cannot say if these areas have or not cocoa. Yet, when observing the overlapping batches, we can see a spatial congruence between the different batch predictions.

5 Conclusions

The main aim of this Bachelor's thesis was to test the feasibility of cocoa segmentation by combining the use of satellite multispectral imagery and deep learning methods. The second objective was to analyze and, thus, understand the learning process and determine the decisive properties of convolutional neural networks and multispectral images during cocoa segmentation. Considering that cocoa is cultivated with different methods depending on the country, we analyzed our method separately for full sun farms and agroforesty farms, using Ecuador and Ghana as representatives for both farming methods.

We can assert that the method developed during this project works very well for full sun farms, such as the ones found in countries like Ecuador. However, since these remarkable results have only been achieved on two farms, they need to be validated using a bigger data set with larger amounts of farms distributed trough Ecuador or, even, with farms from other countries using full sun farming.

We showed the need for multispectral imagery and information of near-infrared and short-wavelength infrared reflectance for the task of cocoa segmentation. Along with it, we found a significant difference between the spectral signature of cocoa cultivated in full sun farms and agroforests. This reinforces the initial decision to treat full sun cocoa and agroforest cocoa farms separately.

In the case of agroforestry, we cannot affirm the applicability of our method due to the uncertainty associated with of our results. However, they indicate some promising directions towards the use of this method for agroforestry.

Furthermore, we propose some improvements and next steps for our project:

- Implementing a more **powerful architecture**.
- Increasing the **amount of data** not only in Ghana but also in Ecuador. This will solve the issue of the data sparsity and additionally decrease the imbalance of the data set. In addition, it will further validate our results and give the possibility of definitively proving the feasibility of the method.
- Increasing the information of the **surrounding of the farms**. This will, on the one hand, counteract the imbalance of the data set and, on the other hand, validate the cocoa predictions that at this point of the project are still uncertain.
- Implementing a method to detect **clouds** and threat them separately from the rest of the data set. This should be integrated in a complete and more sophisticated management of the cloud problem that occurs when using satellite multispecral imagery in tropical regions.
- As a consequence of the previous point, introducing **more labels** to separate different type of crops and vegetation. This will increase the complexity of the model and help the convolutional neural network to better understand the surrounding of the

agroforest farms. For instance, improving its ability to separate streets from different types of vegetation. Of course, this requires that the surrounding of the farms is accurately mapped.

- Computing larger **time series**. In a future step, the idea of temporal data could be generalized to all available cloud free Sentinel-2 images over a span of a year. This will increase the amount of information, not only of the cocoa growth, but also of other types of vegetation such as shading trees that form part of an agroforest.
- Adding **batches of uncultivated forest** to the data set. This will, when using undersampling, help to give the convolutional neural network enough information about other types of vegetation to distinguish between normal forest and agroforest.
- Implementing a **penalization method** for errors on the minority class of the unbalanced data set. This will force the convolutional neural network to take this class more into account.
- As a last step, performing a **hyperparameter fine-tuning** will allow us to find the optimal set of hyperparameters for cocoa segmentation and, thus, also the best possible results.

References

Bibliography

Asare, R. et al. (Dec. 2014). "Cocoa agroforestry for increasing forest connectivity in a fragmented landscape in Ghana". In: *Agroforestry Systems* 88, pp. 1143–1156.

Barima, Y. et al. (2016). "Cocoa crops are destroying the forest reserves of the classified forest of Haut-Sassandra (Ivory Coast)". English. In: *Global Ecology and Conservation* 8.Complete, pp. 85–98.

Duchi, J., E. Hazan, and Y. Singer (July 2011). "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization". In: *J. Mach. Learn. Res.* 12, pp. 2121–2159. ISSN: 1532-4435.

Duveiller, G. et al. (2008). "Deforestation in Central Africa: Estimates at regional, national and landscape levels by advanced processing of systematically-distributed Landsat extracts". In: *Remote Sensing of Environment* 112.5. Earth Observations for Terrestrial Biodiversity and Ecosystems Special Issue, pp. 1969–1981. ISSN: 0034-4257.

Ellis, E. and N. Ramankutty (2008). "Putting people in the map: anthropogenic biomes of the world". In: *Frontiers in Ecology and the Environment* 6.8, pp. 439–447.

He, K. et al. (2015). "Deep Residual Learning for Image Recognition". In: *CoRR* abs/1512.03385.

Heusel, M. et al. (2017). "GANs Trained by a Two Time-Scale Update Rule Converge to a Nash Equilibrium". In: *CoRR* abs/1706.08500.

Inglada, J. et al. (2016). "Improved Early Crop Type Identification By Joint Use of High Temporal Resolution SAR And Optical Image Time Series". In: *Remote Sensing*.

Ioffe, S. and C. Szegedy (2015). "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift". In: *CoRR* abs/1502.03167.

Ivakhnenko, A. G. (1965). *Cybernetic Predicting Devices*. CCM Information Corporation.

Karpathy, A. (2018). *CS231n: Convolutional Neural Networks for Visual Recognition*. Stanford CS. URL: `http://cs231n.github.io/`.

Kent, A. et al. (1955). "Machine literature searching VIII. Operational criteria for designing information retrieval systems". In: *American Documentation* 6.2, pp. 93–101.

Kingma, D. and J. Ba (2014). "Adam: A Method for Stochastic Optimization". In: *CoRR* abs/1412.6980.

Krizhevsky, A., I. Sutskever, and G. Hinton (2012). "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25*. Ed. by F. Pereira et al. Curran Associates, Inc., pp. 1097–1105.

LeCun, Y. et al. (Sept. 1989). "Backpropagation Applied to Handwritten Zip Code Recognition". In: *Neural Computation* 1.4, pp. 541–551. ISSN: 0899-7667.

– (Dec. 1998). "Gradient-Based Learning Applied to Document Recognition". In: *Proceedings of the IEEE* 86, pp. 2278–2324.

Nowaczynski, A. et al. (2017). *Deep learning for satellite imagery via image segmentation*. Blog post.

Obiri, B. et al. (Oct. 2007). "Financial analysis of shaded cocoa in Ghana". In: *Agroforestry Systems* 71, pp. 139–149.

Ronneberger, O., P. Fischer, and T. Brox (2015). "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *CoRR* abs/1505.04597.

Ruf, F. (2007). "Current Cocoa production and opportunities for re-investment in the rural sector. Côte d'Ivoire, Ghana and Indonesia." In: *World Cocoa Foundation, Amsterdam, 23-24 May 2007*.

– (2011). "The Myth of Complex Cocoa Agroforests: The Case of Ghana". In: *Human Ecology* 39.3, pp. 373–388.

Snapir, B., D.M. Simms, and T.W. Waine (2017). "Mapping the expansion of galamsey gold mines in the cocoa growing area of Ghana using optical remote sensing". In: *International Journal of Applied Earth Observation and Geoinformation* 58, pp. 225–233. ISSN: 0303-2434.

Srivastava, N. et al. (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15, pp. 1929–1958.

Wegner, J. D. (2017). *103-0274-00L Bildverarbeitung*. Lecture, ETH Zurich. Course at ETH Zurich.

List of Figures

List of Tables

Appendices

A Appendix

A.1 Additional Visualization of Predictions in Ecuador

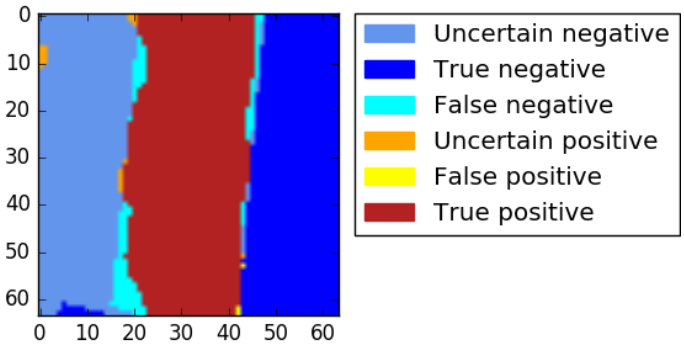


Figure A.1: Visualization of the superposition of ground truth and the predictions for an Ecuadorian farm

A.2 Additional Visualization of Predictions in Ghana

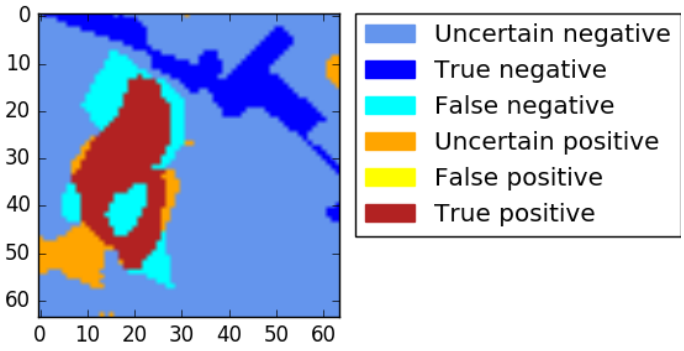


Figure A.2: First visualization of the superposition of ground truth and the predictions for an Ghanian farm

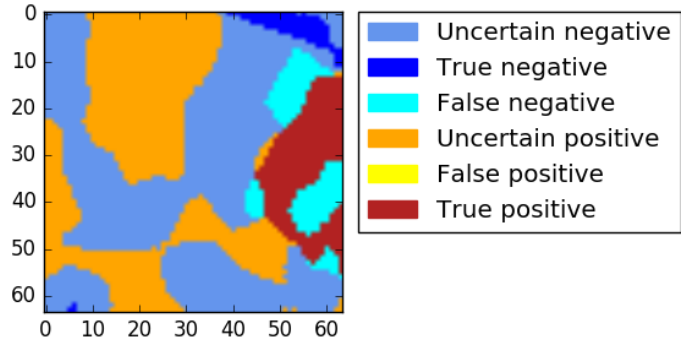


Figure A.3: Second visualization of the superposition of ground truth and the predictions for an Ghanian farm