# Research Internship

## Investigating the potential of Machine Learning to Map Changes in Forest based on Earth Observation


| | |
|---|---|
| Author: | Xuzheng Song |
| Matriculation Number: | S184790 |
| Address: | Connolly Straße 3, T038 80809 München |
| Advisor: | Prof. Marc Van Droogenbroek |
| | Dr. Pietro Ceccato, |
| | Anthony Cioppa, |
| | Dr. Rapaël Marée, |
| | Ulysse Rubens |
| Begin: | 01.12.2018 |
| End: | 28.02.2019 |

# Abstract

The aim of the project is to investigate the use of Machine Learning to analyze time series of satellite images from SENTINEL-2, an Earth observation mission that systematically acquires optical imagery at high spatial resolution. The ultimate purpose is to detect changes in the phenology of different tree species and abrupt changes in the forest of one tile over the Wallonia region in Belgian. However, due to lack of ground truth, there is not sufficient usable data for the research. In order to shorten the tedious work of gathering the information collected on location of satellite image, we have introduced a fully connected artificial neural network model to predict the unknown area based on only small number of annotations which are made by a powerful open source tool Cytomine. The model has an outstanding performance and reach the accuracy of more than 99%. At last, the input importance analysis is carried out, and we can figure out which input is more important in our model.

# Acknowledgements

Firstly, I would like to thank Prof. Van Droogenbroek for the recommendation of the internship, and his team for supporting me for the whole internship. Especially thanks to Anthony Ciopia for supervising me for the whole project.

Secondly, I would like to thank Spacebel for offering me the chance and supervise me of the internship. Especially thank for the guidance of Dr. Ceccato, the internship is able to move on smoothly.

Last but not least, I would like to express my sincere thankfulness for the Cytomine Team, Dr. Rapaël Marée and Ulysse Rubens, who are very friendly and kind to help me though out the internship to solve the problem that I met in Cytomine.

# Contents

**Contents** **iii**

**1 Introduction** **1**

**2 Research Carried Out** **3**
2.1 Cytomine . . . 3
2.1.1 Collaborative Principles . . . 3
2.1.2 Cytomine-WebUI . . . 4
2.1.3 Cytomine for the Project . . . 5
2.2 Data . . . 5
2.3 Artificial Neural Network . . . 9
2.3.1 Activation Function . . . 10
2.3.2 Keras . . . 12
2.3.3 Loss Function and Optimizer Algorithms . . . 12
2.4 Model . . . 13
2.5 Results . . . 14
2.6 Discussion . . . 15

**3 Conclusions and Perspectives** **23**
3.1 Conclusions . . . 23
3.2 Perspectives . . . 23

**List of Figures** **25**

**List of Tables** **27**

**Bibliography** **28**

# Chapter 1

# Introduction

Machine learning algorithms, which learn the representative and discriminative features from the data have been introduced into the geoscience and remote sensing (RS) community. RS techniques have opened a door to help people widen their ability to understand the earth[GLL90][BCM12]. Detecting changes in the phonology of different tree species and abrupt changes in the forest is a fairly important topic in the ecological and environmental studies.

Hence, the aim of the study is to investigate the use of Machine Learning to analyze time series of satellite images, based on SENTINEL-2, to detect changes in the phonology of different tree species and abrupt changes in the forest of one tile over the Wallonia region in Belgian (i.e. impact of pest such as the scolytes bark beetles, fires or deforestation processes). As the southern portion of the country, Wallonia is primarily French-speaking, and accounts for 55% of Belgium's territory and a third of its population.

To meet the objective of this project, Cytomine[Tea19a][MRS+16] application is introduced. As an "Open-source rich Internet application for collaborative analysis of multi-gigapixel images", which has three main properties:

**Open Source**: The source code [Tea19b] is available for everyone and protected by an *Apache License*. Each module is also documented.
**Open Company**: A non-profit company is promoting, coordinating and providing support to this project.
**Open Research**: Cytomine researchers at the Montefiore Institute of the University of Liège are working on machine learning development and image processing for this application.

By the time the research internship was carried out, Cytomine had never been used to satellite images. Some new functions were added by the Cytomine team (Ulysse Rubens) during the project, e.g. support for handling JPEG 2000 format image.

This report is divided into 3 Chapters. After this Introduction, Chapter 2 introduces the

Cytomine, data we used, algorithm details, the implemented model, the result and some discussions. Chapter 3 includes our conclusions for the project and perspectives for the future.

# Chapter 2

# Research Carried Out

In this chapter, we will introduce the Data first and then Cytomine application. After that the model of neural network used for the project will be discussed in detail. At last we come to the result and analysis.

## 2.1 Cytomine

Figure 2.1: Cytomine logo

Cytomine is a rich internet application for remote visualization, collaborative and semantic annotation, and (semi-)automated analysis of high-resolution (multi-gigapixel) images using recent web development, and machine learning techniques[Mar17].

The platform has been designed to faciliate accessibility, curation, and dissemination of imaging data, and to be widely applicable and extensible. Although it has been motivated by biomedical applications, it can be used in other application domains where there are very large images (small ones also). Also, although it has been designed for remote collaboration, it can be installed and used on a local computer for small-scale studies[Mar17].

### 2.1.1 Collaborative Principles

Cytomine platform permits active collaboration between distributed groups of life scientists, computer scientists and citizen scientists. It allows seamless online sharing and reviewing of semantic and quantitative information associated with large images, either produced manually or automatically using machine learning algorithms, as schematically illustrated in 2.2.

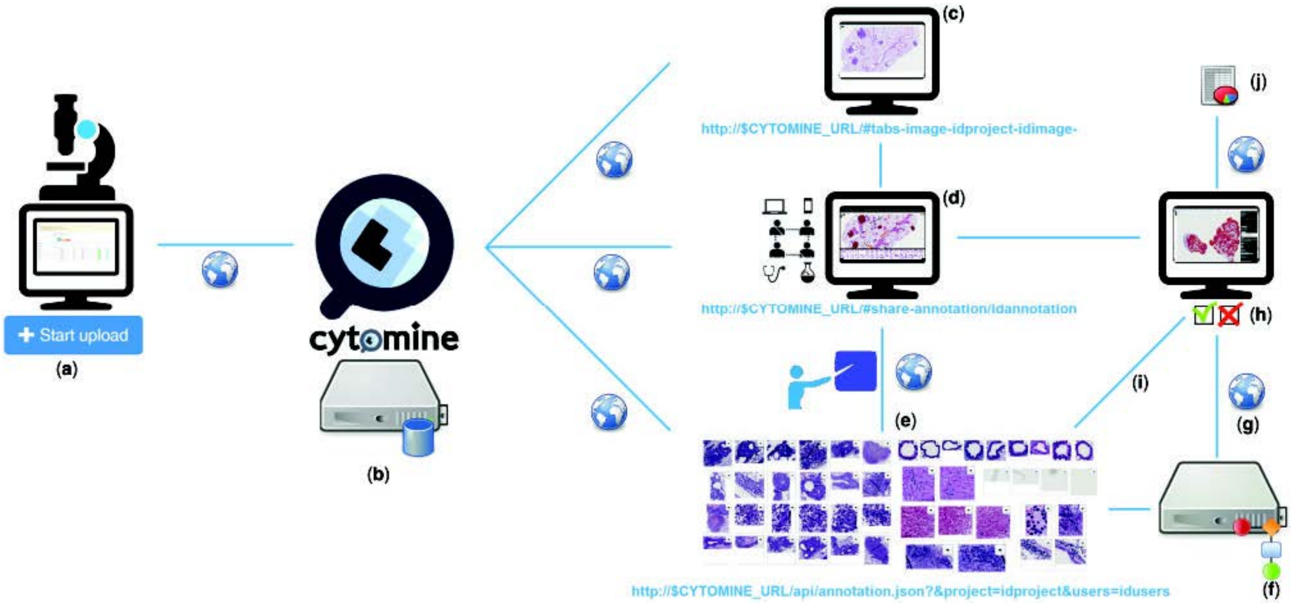


Figure 2.2: Overview of multidisciplinary collaborative principles illustrated for tumor segmentation in H&E lung cancer whole tissue slides: (a) Images are uploaded using Cytomine-WebUI or remote clients. (b) Images and related data are stored by Cytomine-Core and Cytomine-Image Management System. (c) Once uploaded, multi-gigapixel images are de facto available to other distributed users according to access rights and referenced by URLs. (d) Remote, multidisciplinary individuals are collaboratively and semantically annotating regions of interest in images and each annotation is referenced by its URL. (e) Expert annotations can be filtered and sets of annotations can be displayed or retrieved through the API. (f) Distributed algorithms can exploit these annotations, here a segmentation recognition model is built by supervised learning based on expert training examples. (g) An algorithm or recognition model can be applied remotely on new multi-gigapixel images for automatic annotation. (h) Experts review other user and automatic annotations by using Cytomine-WebUI proofreading tools. (i) Reviewed annotations can eventually be reused to refine and re-apply the recognition model. (j) Once image annotations are validated by an expert, final quantification results of the reviewed layer are exported in standard formats[Mar17]

### 2.1.2 Cytomine-WebUI

Cytomine-WebUI is a customizable and responsive rich internet application 2.3, accessible through regular web browsers and mobile devices. It allows to create, organize, visualize and edit all data. It includes a zoomable, tile-based viewer for multi-gigapixel images with the visualization of overlaid (human or computer-generated) annotation layers and their properties. Furthermore, an ontology editor, several modules to derive annotation statistics and visualize annotation galleries, a textual search engine and proofreading tools for expert reviewing of annotation objects are part of this user interface. In addition, we have implemented functionalities to allow various forms of collaborative works. One of them is the tracking of all user activities to e.g. allow multiple users to follow remotely another user's observation paths and actions. Conversely, a blinded mode can be activated to hide image and user information to allow independent studies and reduce bias when analyzing imaging data. An additional module (Cytomine-IRIS, the interobserver reliability study module) also allows independent ground-truth construction and inter-observer annotation statistics e.g. to identify cell type classification disagreements among experts. For more details please find in the [Tea19a].

### 2.1.3 Cytomine for the Project

We have created ULG-INTELSIG-EARTHECO ontology, which includes 6 terms. Each term is addressed with different colors: Agricultural Fields (Yellow), Cities Infrastructures (Purple), Clouds (White), Snow (Grey), Water (Blue) and Forest (Green). There are two sub-terms for Forest– Deciduous (Dark Green) and Evergreen (Light Green). In our project, the term "Snow" is not used since the time we select are free from snow. Dr. Ceccato and I did the labeling in one true color image.

In our later stage of the research, we made the predictions to different landscapes, in order to visualize the prediction, we uploaded the predicted annotation onto Cytomine, but the visualization is not satisfying. We will talk about more details in the following sub-chapters.

## 2.2 Data

Forests cover almost one third of the Earth's land surface and they play a major role in global carbon and water cycles, but are also an important source of raw material for industry, fuel and other ecosystem services. The multiple functions of forests can lead to conflicting requirements for their management, with e.g. sustainability as the leading demand from environmental point of view. Reliable data on forest resources is required to provide objective information for sustainable forest management[AHS+19].

The European Copernicus program with its Sentinel satellites is an operational environment monitoring system providing free satellite data for decades to come. The main Copernicus instrument for forestry purposes is the Sentinel-2 satellite series. Two satellites provide images from 10- to 60-meter spatial resolution with 13 spectral bands and five

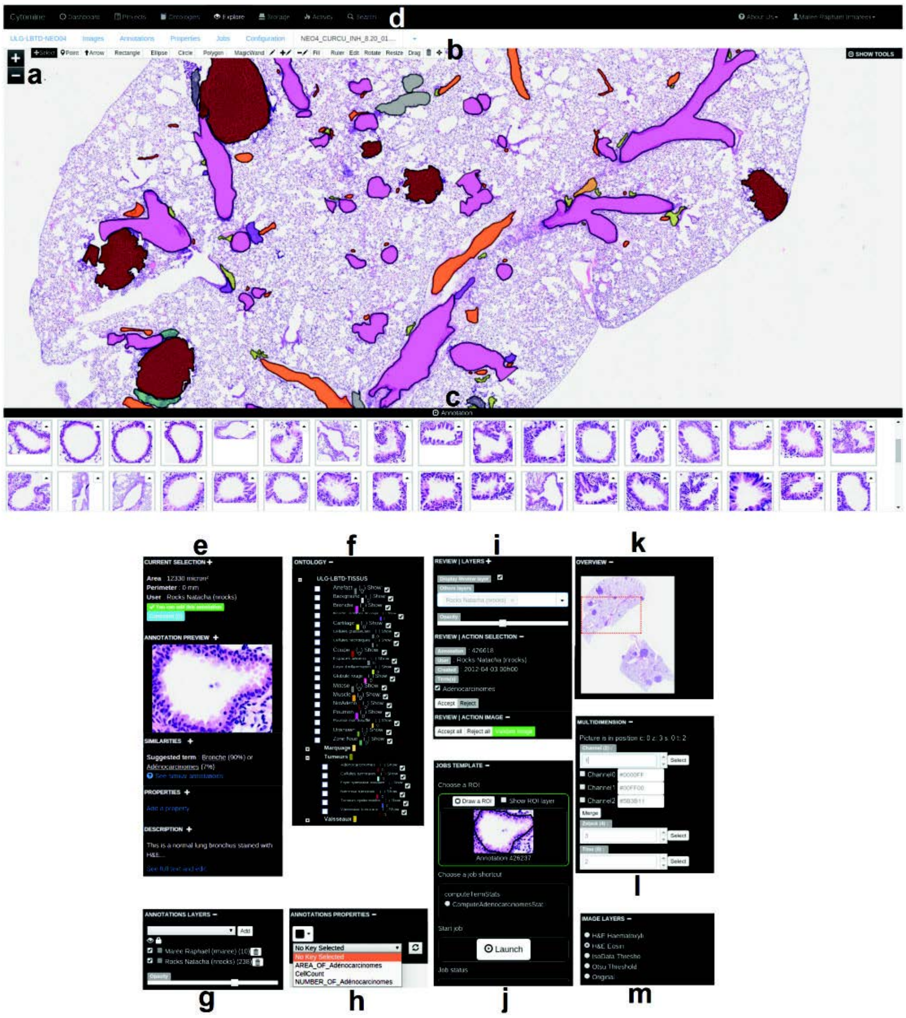


Figure 2.3: Overview of Cytomine-WebUI: (a) Zoomable multi-gigapixel image viewer (a la Google Maps) with overlaid annotations colored according to ontology terms (Original image size: 19960×25088 pixels). (b) Annotation drawing tools including various shapes and operations on polygons. (c) Gallery of bronchus annotations in current image. (d) Main menu including project listing, ontology editor, storage to upload images, user activity statistics, textual search engine. (e) Selected annotation panel with thumbnail, suggested terms (based on content-based image retrieval algorithm), textual description.(f) Project-specific, userdefined ontology for semantic annotation. (g) Activation of annotation layers of possibly distributed users and softwares. (h) Annotation properties (key-value pairs). (i) Proofreading tools to accept or edit annotations. (j) Job template panel to launch pre-configured processing routines on regions of interest. (k) Gigapixel image overview with current position. (l) Multidimensional image panel with selectors for channel, slice in a z-stack, and time point. (m) Image layer panel to apply on-the-fly tile image processing[Mar17]

day revisit frequency at the equator [DDBC+12]. The spectral bands for the Sentinel-2 are shown in table 2.1.

Table 2.1: Spectral bands for the SENTINEL-2 sensors[Age18a]

| Sentinel-2 bands | Sentinel-2A | | Sentinel-2B | | |
|---|---|---|---|---|---|
| | Central wavelength (nm) | Bandwidth (nm) | Central wavelength (nm) | Bandwidth (nm) | Spatial resolution (m) |
| Band 1- Coastal aerosol | 442.7 | 21 | 442.2 | 21 | 60 |
| Band 2 - Blue | 492.4 | 66 | 492.1 | 66 | 10 |
| Band 3 - Green | 559.8 | 36 | 559 | 36 | 10 |
| Band 4 - Red | 664.6 | 31 | 664.9 | 31 | 10 |
| Band 5 - Vegetation red edge | 704.1 | 15 | 703.8 | 16 | 20 |
| Band 6 - Vegetation red edge | 740.5 | 15 | 739.1 | 15 | 20 |
| Band 7 - Vegetation red edge | 782.8 | 20 | 779.7 | 20 | 20 |
| Band 8 - NIR | 832.8 | 106 | 832.9 | 106 | 10 |
| Band 8A - Narrow NIR | 864.7 | 21 | 864 | 22 | 20 |
| Band 9 - Water vapour | 945.1 | 20 | 943.2 | 21 | 60 |
| Band 10 - SWIR - Cirrus | 1373.5 | 31 | 1376.9 | 30 | 60 |
| Band 11 - SWIR | 1613.7 | 91 | 1610.4 | 94 | 20 |
| Band 12 - SWIR | 2202.4 | 175 | 2185.7 | 185 | 20 |

The area that we will study is one tile over the Wallonia region in Belgian. This region is always covered with snow in the winter and clouds all around the year. Clouds, cloud shadows, and snow significantly influence the spectral bands of optical sensors. Their presence can cause serious problems for a variety of remote sensing activities including: image compositing; correction for atmosphere effects; calculation of vegetation indices; classification of land cover; and most importantly in change detection[ZWW15]. Therefore, it is important to get rid of the influence of clouds, cloud shadows, and snow in satellite images. Due to the time limit of the project, Spacebel offered a set of data which is clouds and snow free in the year of 2017. Only five times in the year were selected, which are 27th March, 26th May, 7th July, 29th August, and 15th October. At those times, most of the region in selected area are free from snow and clouds, although there are some small cloud in particular time, the influence are neglected or considered as the noise of the data. The True Color Images (TCI) are also offered, which is an RGB image built from the B02 (Blue), B03(Green), and B04 (Red) Bands. The reflectances are coded between 1 and 255, 0 being reserved for 'No Data'. The saturation level of 255 digital counts correspond to a level of 2000 for the individual bands (i.e. reflectance of 0.2)[Age18b]. TCI of a small fraction of the selected area is are shown in 2.4.

During the study, we used all the bands except for B10. So in total we have 55 images (5 times * 11 bands) to be operated. There are 3 resolutions of 11 bands, so there are 3 sizes of image, 1830 * 1830, 5490 * 5490, 10980 * 10980. Since all the bands are correlated to

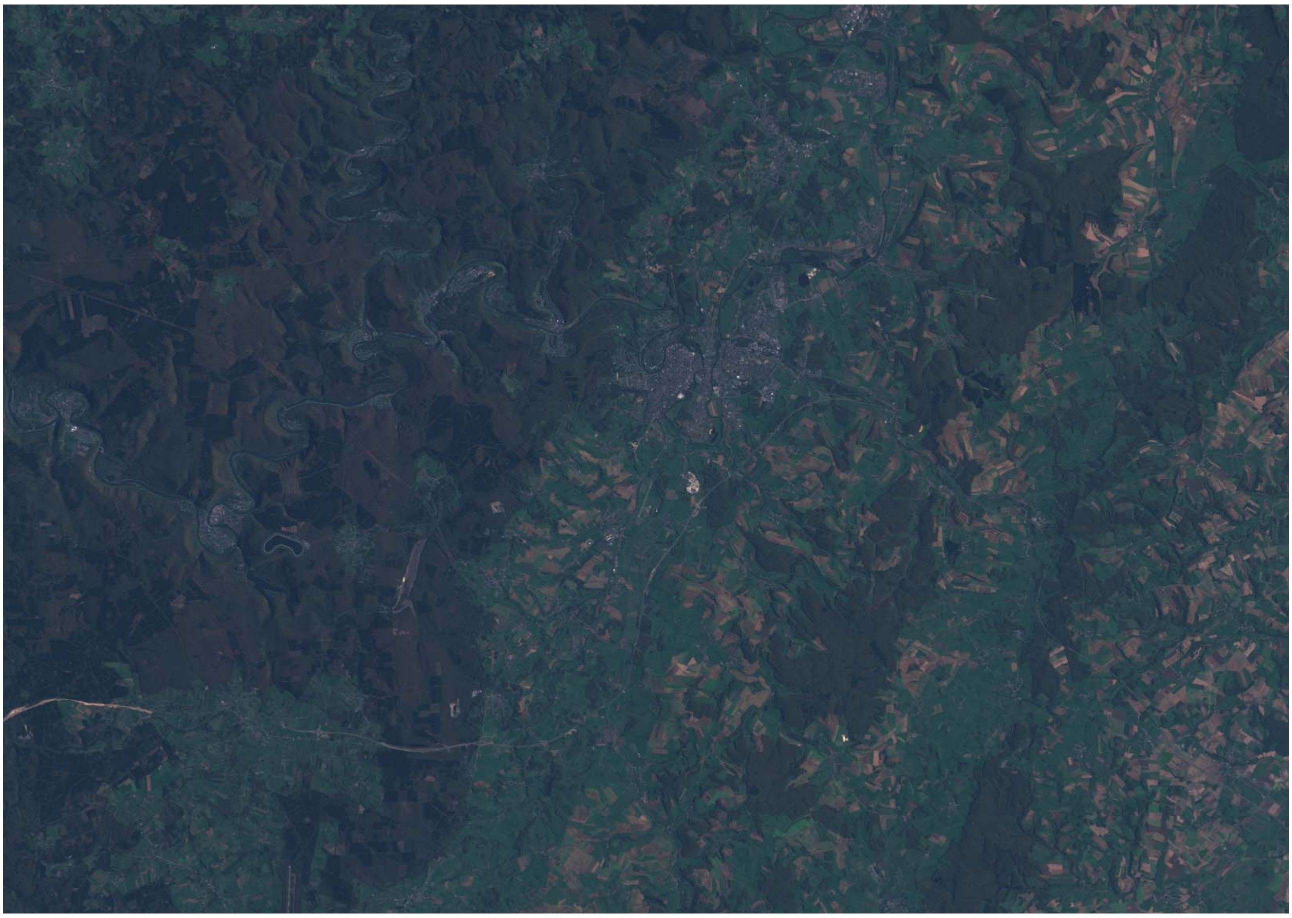

Figure 2.4: TCI sample image

the same region, in order to do the labeling easier and do not lose useful information of the images, the images are reshaped to the size of 10980*10980 using the bilinear interpolation.

Since all we care is only the forests (Deciduous and Evergreen), we suppose the region of the forests of one time will also remain forests of another time, because the forest are quite stable throughout the year. Under this assumption we can duplicate the annotations from one time to another times through Cytomine. That is to say, as long as we label one image on Cytomine, we can duplicate the labeling to all other images.

Dr. Ceccatto and I have done the labeling the image randomly in one true color image, until we get enough labels for our algorithm, part of annotations are shown in 2.5. To be mentioned, we have 7 terms of the ontology, we made the labeling corresponded to the landscape. But in later analysis, we simply classified 7 terms into 3 categories – Deciduous, Evergreen, Other, because we only focus on studying forest. The category "Other" is composed of the other terms except "Deciduous" and "Evergreen". We picked up the annotation locations randomly.

Through Cytomine, all the labeled crops are downloaded. Using Python script, the information of each point of the crops is extracted, as a point in the image stands for 55 pixel values (11 bands * 5 times), which are stored in a numpy array. So the time and spectral information are taken into consideration when we run the neural network algorithm.

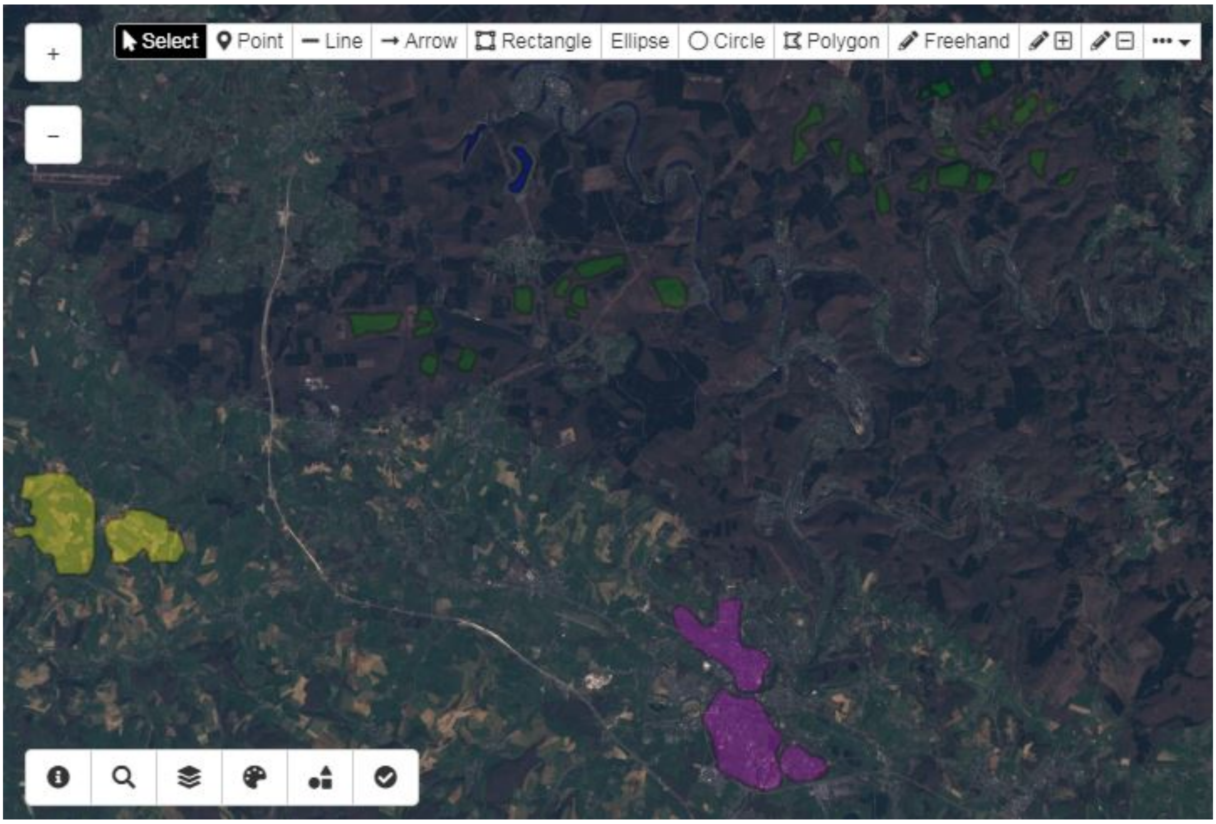


Figure 2.5: Part of annotations on Cytomine

## 2.3 Artificial Neural Network

A neural network is a massively parallel distributed processor made up of simple processing units that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects[HHH+09]:

1. Knowledge is acquired by the network from its environment through a learning process.

2. Interneuron connection strengths, known as synaptic weights, are used to store the acquired knowledge.

Artificial neural networks (ANN) offer the following useful properties and capabilities[HHH+09]:

1. Nonlinearity. An artificial neuron can be linear or nonlinear. A neural network, made up of an interconnection of nonlinear neurons, is itself nonlinear. Moreover, the nonlinearity is of a special kind in the sense that it is distributed throughout the network. Nonlinearity is a highly important property, particularly if the underlying physical mechanism responsible for generation of the input signal (e.g., speech signal) is inherently nonlinear.

2. Adaptivity. Neural networks have a built-in capability to adapt their synaptic weights to changes in the surrounding environment. In particular, a neural network trained to operate in a specific environment can be easily retrained to deal with minor changes in the operating environmental conditions.

3. Fault Tolerance. A neural network, implemented in hardware form, has the potential to be inherently fault tolerant, or capable of robust computation, in the sense that its performance degrades gracefully under adverse operating conditions. For example, if a neuron or its connecting links are damaged, recall of a stored pattern is impaired in quality.

The neural network structure is composed of input layer, hidden layer and output layer. A typical neural network structure is shown in 2.6. There are one input layer with two dimensions, two hidden layer and one output layer.

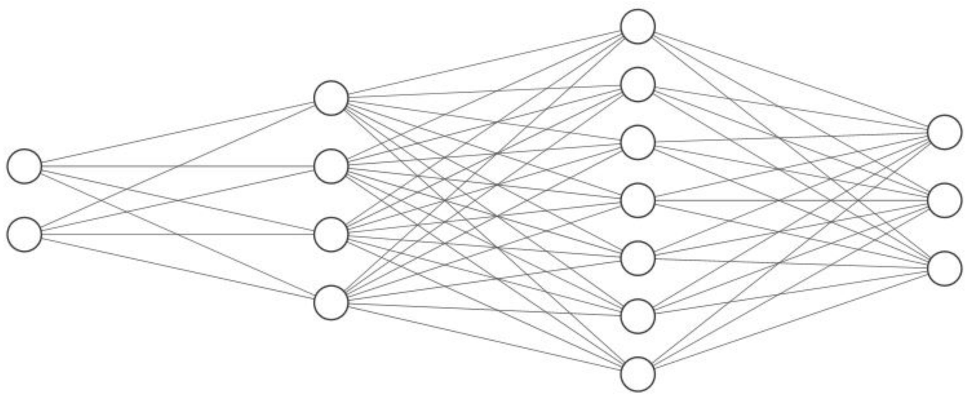

Figure 2.6: Example of the neural network layer

### 2.3.1 Activation Function

Another crucial basic architectures of ANN is activation functions, since they introduce non-linear properties to the network. An activation function sets the output behavior of each node, or "neuron" in an ANN. This output is then used as input for the next node and so on until a desired solution to the original problem is found. This allows the ANN to make sense of and learn from complicated, non-linear mappings between inputs and response variables. Without these functions, then nodal activation could only be a linear process, which would exponentially increase the processing power and time needed to solve problems[DA18].

**Rectified Linear Units (ReLU)** are used in our model as the activation function for the hidden layer. The equation function of ReLU is 2.1

$$f(x) = max(x, 0) \tag{2.1}$$

In other words, the activation is simply thresholded at zero (see image above on the left). There are several pros and cons to using the ReLUs:

• (+) It was found to greatly accelerate the convergence of stochastic gradient descent compared to the sigmoid/tanh functions. It is argued that this is due to its linear, non-saturating form.

• (+) Compared to tanh/sigmoid neurons that involve expensive operations (exponentials, etc.), the ReLU can be implemented by simply thresholding a matrix of activations at zero.

• ( - ) Unfortunately, ReLU units can be fragile during training and can "die". For example, a large gradient flowing through a ReLU neuron could cause the weights to update in such a way that the neuron will never activate on any datapoint again. If this happens, then the gradient flowing through the unit will forever be zero from that point on. That is, the ReLU units can irreversibly die during training since they can get knocked off the data manifold. For example, you may find that as much as 40% of your network can be "dead" (i.e. neurons that never activate across the entire training dataset) if the learning rate is set too high. With a proper setting of the learning rate this is less frequently an issue[cs215].

**Softmax** is used in our model as the activation function for the output layer. Mathematically the softmax function is shown in 2.2, where z is a vector of the inputs to the output layer (if you have 10 output units, then there are 10 elements in z). And again, j indexes the output units, so j = 1, 2, ..., K.

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \tag{2.2}$$

The softmax function squashes the outputs of each unit to be between 0 and 1, just like a sigmoid function. But it also divides each output such that the total sum of the outputs is equal to 1, as 2.7 shows. The output of the softmax function is equivalent to a categorical probability distribution, it tells you the probability that any of the classes are true.

The softmax can be used for any number of classes. It's also used for hundreds and thousands of classes, for example in object recognition problems where there are hundreds of different possible objects[cs215]. In our project, there are 3 different output classes.

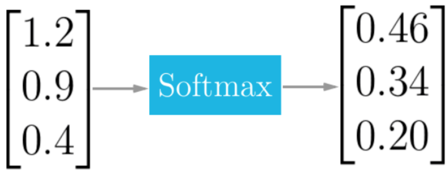


Figure 2.7: Example of Softmax

### 2.3.2 Keras

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. The advantage of Keras is as follows.

• User friendliness. Keras is an API designed for human beings, not machines. It puts user experience front and center. Keras follows best practices for reducing cognitive load: it offers consistent & simple APIs, it minimizes the number of user actions required for common use cases, and it provides clear and actionable feedback upon user error.

• Modularity. A model is understood as a sequence or a graph of standalone, fully configurable modules that can be plugged together with as few restrictions as possible. In particular, neural layers, cost functions, optimizers, initialization schemes, activation functions and regularization schemes are all standalone modules that you can combine to create new models.

• Easy extensibility. New modules are simple to add (as new classes and functions), and existing modules provide ample examples. To be able to easily create new modules allows for total expressiveness, making Keras suitable for advanced research.

• Work with Python. No separate models configuration files in a declarative format. Models are described in Python code, which is compact, easier to debug, and allows for ease of extensibility[Ker].

### 2.3.3 Loss Function and Optimizer Algorithms

Loss function is a method of evaluating how well your algorithm models your dataset. If your predictions are totally off, your loss function will output a higher number. If they are pretty good, it will output a lower number. As you change pieces of your algorithm to try and improve your model, your loss function will tell you if you are getting anywhere.

During the training process, we tweak and change the parameters (weights) of our model to try and minimize that loss function, and make our predictions as correct as possible. Optimizers shape and mold your model into its most accurate possible form by futzing with the weights. The loss function is the guide to the terrain, telling the optimizer when it is moving in the right or wrong direction. They tie together the loss function and model parameters by updating the model in response to the output of the loss function[Alg].

## 2.4 Model

A fully connected neural network is built as 2.6. The neural network is composed of 1 input layer, 2 hidden layer and 1 output layer. For the input layer, we have 55 dimensions, for we have 11 bands * 5 times. For the hidden layers, we put 128 neurons and 256 neurons separately. For the output layer, we put 3 neurons, which classify the input into 3 categories - Deciduous, Evergreen, Other. We randomized the weights by uniform distribution. The sparse categorical crossentropy as the lost function and RMSProp as the optimizer are being used.

In each class 20000 sets of point are chosen to form a training set, which contains 20% of validation set in order to verifying that any increase in accuracy over the training data set actually yields an increase in accuracy over a data set that has not been shown to the network before, or at least the network hasn't trained on it (i.e. validation data set). If the accuracy over the training data set increases, but the accuracy over the validation data set stays the same or decreases, then overfitting happens and the training should be stopped.

15000 sets of point are chosen to form a testing set. This data set is used only for testing the final solution in order to confirm the actual predictive power of the network. The crucial part of code is as follows.

**Algorithm Code**

```
model = Sequential()
model.add(Dense(128, input_dim=55, init='uniform',activation='relu'))
model.add(Dense(256,init='uniform',activation='relu'))
# output activation
model.add(Dense(3,init='uniform',activation='softmax'))

rmsprop = RMSprop(lr=0.001,rho=0.9,epsilon=1e-08,decay=0.0)
# build the model
model.compile(optimizer = rmsprop,loss = 'sparse_categorical_crossentropy',metrics=['accuracy'])
# train
model.fit(x_train,y_train, validation_split=0.2,epochs=2,batch_size=32,verbose=1)
# evaluate
loss,accuracy = model.evaluate(x_test,y_test)
print("loss:",loss)
print("accuracy",accuracy)
```

## 2.5 Results

The result is shown in 2.8. The result is extraordinary for final loss is about 0.006 and the accuracy is 0.998.

```
In [11]: runfile('/Users/apple/Desktop/Spacebel/Program/pixel_based_network.py', wdir='/Users/
apple/Desktop/Spacebel/Program')
89500
35600
827600
/Users/apple/Desktop/Spacebel/Program/pixel_based_network.py:62: UserWarning: Update your
`Dense` call to the Keras 2 API: `Dense(128, input_dim=55, activation="relu",
kernel_initializer="uniform")`
  rmsprop = RMSprop(lr=0.001,rho=0.9,epsilon=1e-08,decay=0.0)
/Users/apple/Desktop/Spacebel/Program/pixel_based_network.py:63: UserWarning: Update your
`Dense` call to the Keras 2 API: `Dense(256, activation="relu", kernel_initializer="uniform")`
  # build the model
/Users/apple/Desktop/Spacebel/Program/pixel_based_network.py:64: UserWarning: Update your
`Dense` call to the Keras 2 API: `Dense(3, activation="softmax",
kernel_initializer="uniform")`
  model.compile(optimizer = rmsprop,loss =
'aparse_categorical_crossentropy',metrics=['accuracy'])
Epoch 1/2
60000/60000 [==============================] - 2s 39us/step - loss: 0.0520 - acc: 0.9841
Epoch 2/2
60000/60000 [==============================] - 2s 35us/step - loss: 0.0357 - acc: 0.9929
45000/45000 [==============================] - 1s 16us/step
loss: 0.006490495928660837
accuracy 0.9983777777777778
```

Figure 2.8: Training result

The behavior of accuracy and loss of two epochs is shown in 2.9 and 2.10. These figures show that the influence of overfitting can be neglected.

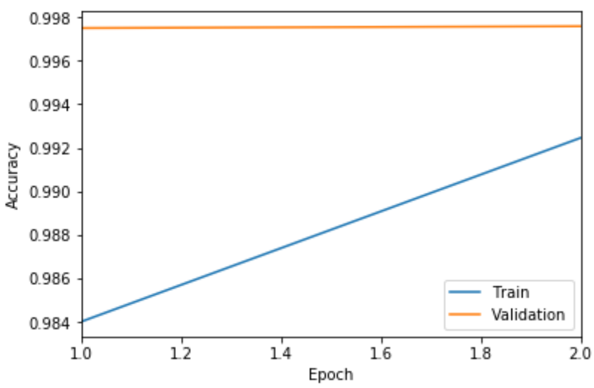


Figure 2.9: Model accuracy

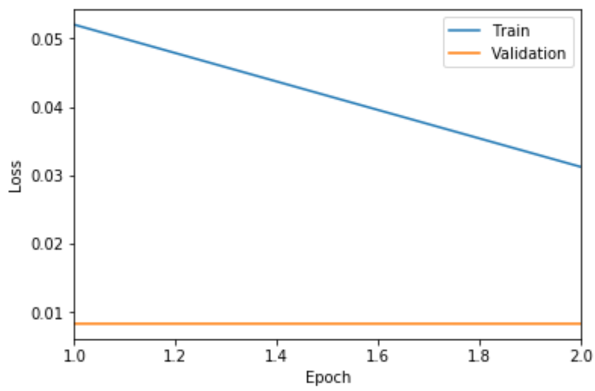


Figure 2.10: Model loss

In order to have an intuitive understanding of the result. An unlabeled area was chosen to be predicted and visualized. The chosen area has the size of 2745*5490 pixels. The TCI of the selected area is 2.11

We extracted the information from the image set and get the prediction of the image. When we got the prediction, we visualized them on Cytomine as the 2.12 shows.

The visualization is bizarre because visualization the prediction on Cytomine is a little complicated. We can pixel-wise upload the labels to the image but there will be too many annotation points and Cytomine cannot merge all the neighbor pixel with same labels. Instead they are just separated points with labels. In this way the shape of landscape will not show and the upload process will last a long time.

The way to upload the annotations is to upload the Polygon type in shapely.geometry model in Python. With the prediction, we find out the contours, but the contours do not contains the information of direction. So although we find the contours of each classes, it has no information on whether the inside or outside of the contours represent the class. What is more, there are lots of landscape surrounded by the other landscape. For example there are some Evergreen surrounded by Crops, or some Water surrounded by City Infrastructures, which explains that there are some overlapping in the annotation, which actually want to present the area in between.

For the reasons above, the visualization of the prediction is not satisfying and the labels are not right. But still we can see some boundaries between each classes in detail. T hey are perfectly distinguished. In order to get a better view of our prediction, another way is adopted.

This time, we made a mask by the predicted labels as the 2.13 shows. We merged the mask and the original image together and get 2.14.

Compare 2.11 with 2.14, we can see the prediction results are quite satisfying. We take a small part from the region and make some comparisons. As the 2.15 shows, the three categories are separated clearly with three colors. Almost all the categories are correctly classified apart from small regions.

One mis-prediction is most obvious in the image. There is a river running through the image from top to the bottom. The river is mis-classified to the Evergreen instead of Other. The main reason for that is we are lack of water annotations compared to the other annotations since in this area there are not enough water. In this case only few of Water are chosen to be the training set.

## 2.6 Discussion

In order to find out the importance of particular bands or time for the classification, 2 more experiment was carried out. We manually adjusted the test set and observe the classification result.

The first experiment is to find out the importance of bands information. There are 11 bands that we used for classification. And we made a combination of these 11 bands, so there are 2048 combinations in total. We drop the combination of choosing all the band because it means no input. Each of the combination is composed of different bands. We manually set the data, which is in the combinations, to zero in test set individually, and let this training set run through our model and observe the results, and let this training set run through our model and we observe the results.

There are 132 combinations of different bands which can reach more than 90% of accuracy. We counted the total number of each band in these 132 combinations and the table 2.2 is made. From the table we can see the number of band 1, 11 and 12 are much less than other bands, because they are less set to 0 in the 132 combinations. That is to say, in order to get the accuracy greater than 90%, it is more likely to use the information of these 3 bands. On the contrary, the number of band 2, 3 and 5 are quite large, more than 100 out of 132 combinations, which represents less importance in our model.

Table 2.2: The number of each band (set to zero) in 132 combinations which has the greater accuracy than 90%

| | | | | | | |
|---|---|---|---|---|---|---|
| Band | 1 | 2 | 3 | 4 | 5 | 6 |
| Number | 61 | 102 | 104 | 91 | 108 | 94 |
| Band | 7 | 8 | 9 | 11 | 12 | |
| Number | 89 | 99 | 97 | 63 | 74 | |

The second experiment is to find out the importance of time information using the similar way as above. We have 5 times a year in our model, so there will be 32 combinations in total. Similarly we drop the combination of choosing all the times. We set all the bands value in chosen time to 0 and observe the accuracy. There are only 8 combinations reaching more than 90% accuracy. The table 2.3 is made according to same statistic method above. In the table we can see that 27th March and 7th July have been only counted once, so statistically speaking they are more important.

Table 2.3: The number of each time (set to zero) in 8 combinations which has the greater accuracy than 90%

| Time | 27th March | 26th May | 7th July | 29th August | 15th October |
|---|---|---|---|---|---|
| Number | 1 | 4 | 1 | 5 | 4 |

To be clear, the result above are based on statistics. Although it shows in more situations where the information of some bands or particular time is more used than the others, it is not a guarantee to say the information of other bands or time is not important. Because there are also some other factors should be considered. For example, although some of the individual bands or time are of not much importance, the combination of different bands or time can play a significant role in our model. What is more, in the first experiment, we set the individual band to zero of all the time, so this experiment did not give us the information about the time. Because we have chosen 5 times in a year, maybe one band is more important in March of this year whereas the other bands are more important in July. Similarly, the second experiment did not offer us the information about the bands, since we set all the bands to 0 in selected time. After all, these two experiment only show us the statistic importance of individual time and band of the model.

Moreover, all the 2 experiments are based and worked only for our model. If the model has been changed, the importance analysis above will be meaningless and new experiment should be carried out.

Figure 2.11: To be predicted image


Figure 2.12: Visualization on Cytomine (part)

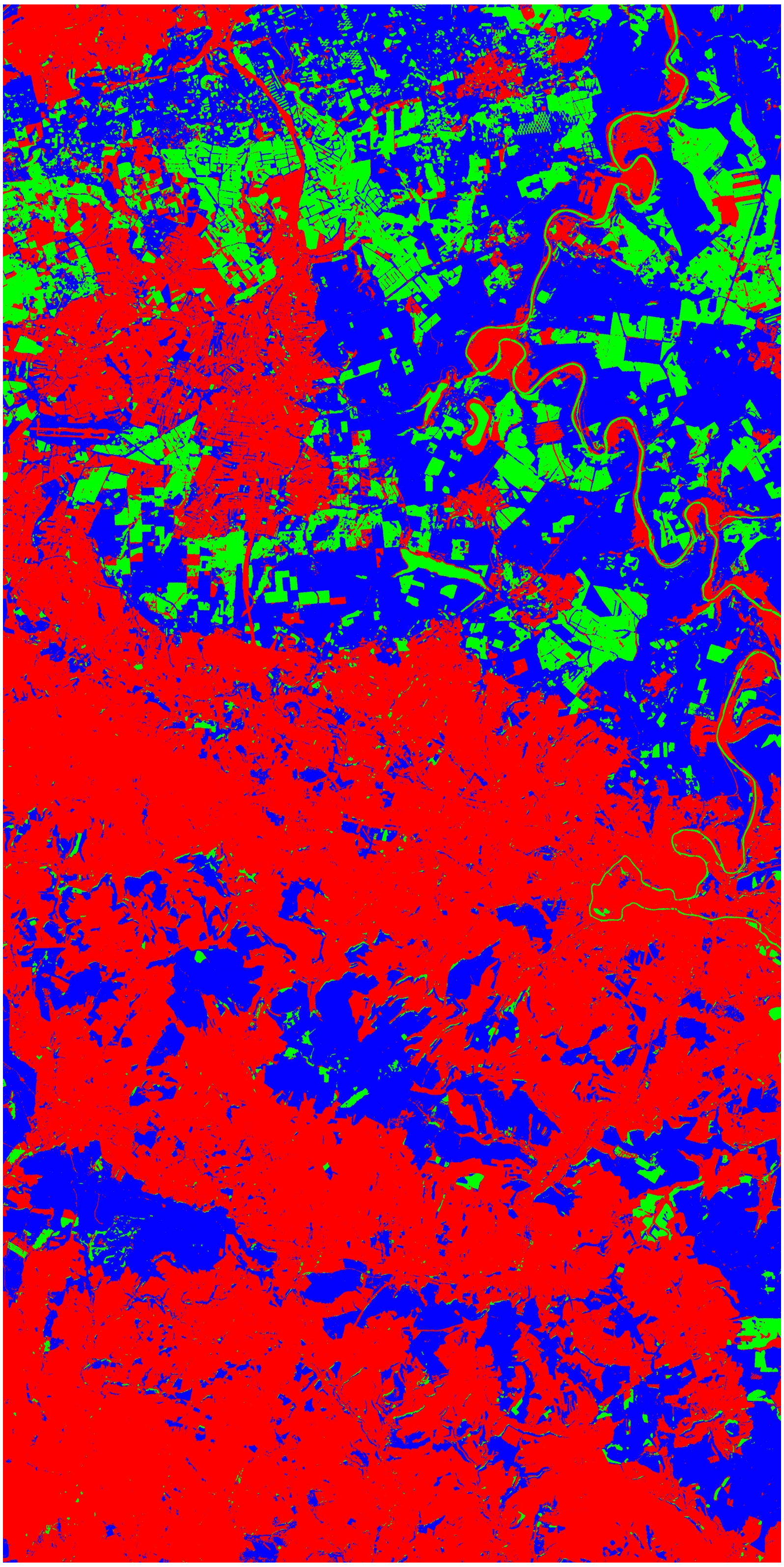

Figure 2.13: The predicted label: Blue = Deciduous, Red = Other, Green = Evergreen

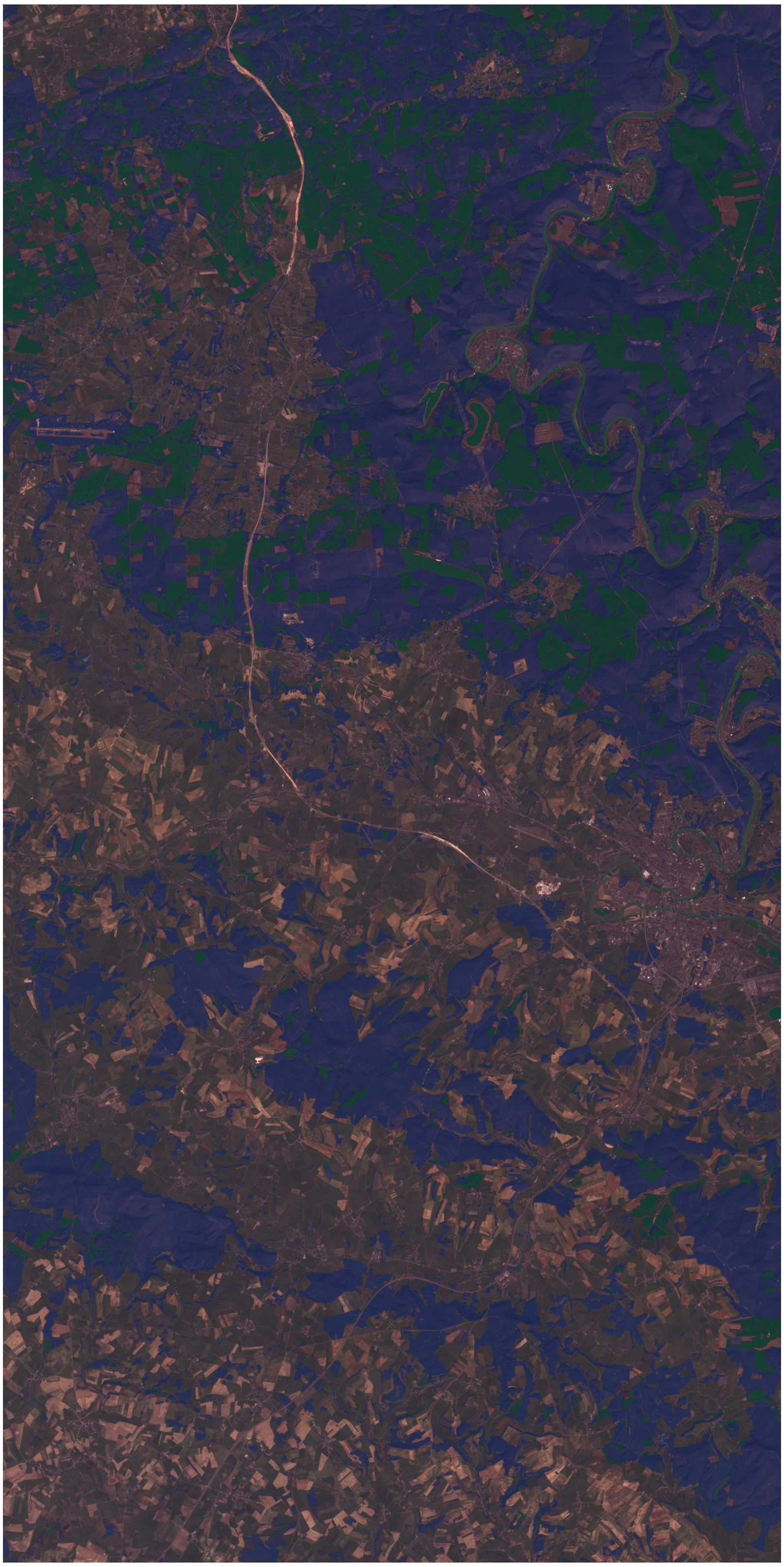

Figure 2.14: Merged predicted image

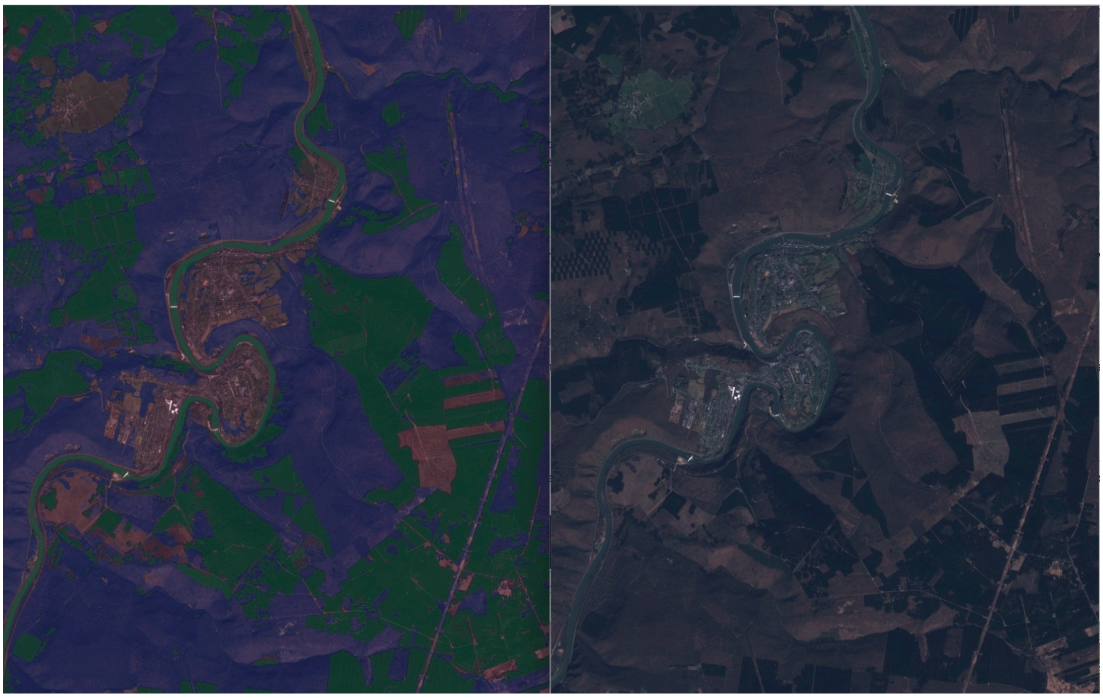

Figure 2.15: Comparision of the predicted and original images (Part); predicted: left, original: right

# Chapter 3

# Conclusions and Perspectives

## 3.1 Conclusions

In this work, we first focus on the data from Sentinel-2. We study the data itself and try to optimise the extraction of data. We indeed considered the extractor's speed as a key factor to allow the usage of a remote data storage system such as the system provided by Cytomine. Moreover, we made it possible to locally store the data fetched from the platform, which makes it easier to reuse. With the help of Cytomine, a proper training set and testing set were made.

Then we introduced the artificial neural network model using Keras. A 4 layer neural network was made to predict the categories of unknown area. In the test set the model performed well and reached the accuracy of 0.998. We managed to visualize the prediction of a larger area, in which the performance of the model was extraordinary.

Finally, 2 experiments are carried out to find out the input importance. We draw a conclusion that the information of band 1, 11 and 12 is more important than the other bands, and 27th of March and 7th of July are of more importance than other date in the statistical sense.

## 3.2 Perspectives

We have visualized the prediction locally, but a way must be found out to upload the prediction to Cytomine. If the prediction can be visualized on Cytomine, then we are able to review all the predictions and get rid of the mis-classification. Then we will have the ground truth to do the other researches.

Due to lack of training data for Water categories, we can add more data of water manually to the training set in order to get a better performance for the model.

We got the importance analysis through statistical method. A perspective study for the reason why these input are more important can be carried out. This can help a better understanding of the Sentinel-2 data and functions in the future.

After we get enough ground truth, we can develop more advanced algorithm to detect changes of forests in Wallonia region, which is our final destination.

Another important issue is cloud and snow. Due to the uncertainty of snow and clouds, we have selected only the cloud-free and snow-free images. But in reality, most of the images have cloud. An algorithm should be developed to avoid the influence of clouds and snow.

# List of Figures


2.1 Cytomine logo . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

2.2 Overview of multidisciplinary collaborative principles illustrated for tumor segmentation in H&E lung cancer whole tissue slides: (a) Images are uploaded using Cytomine-WebUI or remote clients. (b) Images and related data are stored by Cytomine-Core and Cytomine-Image Management System. (c) Once uploaded, multi-gigapixel images are de facto available to other distributed users according to access rights and referenced by URLs. (d) Remote, multidisciplinary individuals are collaboratively and semantically annotating regions of interest in images and each annotation is referenced by its URL. (e) Expert annotations can be filtered and sets of annotations can be displayed or retrieved through the API. (f) Distributed algorithms can exploit these annotations, here a segmentation recognition model is built by supervised learning based on expert training examples. (g) An algorithm or recognition model can be applied remotely on new multi-gigapixel images for automatic annotation. (h) Experts review other user and automatic annotations by using Cytomine-WebUI proofreading tools. (i) Reviewed annotations can eventually be reused to refine and re-apply the recognition model. (j) Once image annotations are validated by an expert, final quantification results of the reviewed layer are exported in standard formats[Mar17] . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

2.3 Overview of Cytomine-WebUI: (a) Zoomable multi-gigapixel image viewer (a la Google Maps) with overlaid annotations colored according to ontology terms (Original image size: 19960×25088 pixels). (b) Annotation drawing tools including various shapes and operations on polygons. (c) Gallery of bronchus annotations in current image. (d) Main menu including project listing, ontology editor, storage to upload images, user activity statistics, textual search engine. (e) Selected annotation panel with thumbnail, suggested terms (based on content-based image retrieval algorithm), textual description.(f) Project-specific, userdefined ontology for semantic annotation. (g) Activation of annotation layers of possibly distributed users and softwares. (h) Annotation properties (key-value pairs). (i) Proofreading tools to accept or edit annotations. (j) Job template panel to launch preconfigured processing routines on regions of interest. (k) Gigapixel image overview with current position. (l) Multidimensional image panel with selectors for channel, slice in a z-stack, and time point. (m) Image layer panel to apply on-the-fly tile image processing[Mar17] . . . 6
2.4 TCI sample image . . . 8
2.5 Part of annotations on Cytomine . . . 9
2.6 Example of the neural network layer . . . 10
2.7 Example of Softmax . . . 12
2.8 Training result . . . 14
2.9 Model accuracy . . . 14
2.10 Model loss . . . 14
2.11 To be predicted image . . . 18
2.12 Visualization on Cytomine (part) . . . 19
2.13 The predicted label: Blue = Deciduous, Red = Other, Green = Evergreen 20
2.14 Merged predicted image . . . 21
2.15 Comparision of the predicted and original images (Part); predicted: left, original: right . . . 22

# List of Tables

2.1 Spectral bands for the SENTINEL-2 sensors [Age18a] . . . . . . . . . . . . 7
2.2 The number of each band (set to zero) in 132 combinations which has the greater accuracy than 90% . . . . . . . . . . . . . . . . . . . . . . . . . 16
2.3 The number of each time (set to zero) in 8 combinations which has the greater accuracy than 90% . . . . . . . . . . . . . . . . . . . . . . . . . 16

# Bibliography


[Age18a] European Space Agency. MultiSpectral Instrument (MSI) Overview. https://earth.esa.int/web/sentinel/technical-guides/sentinel-2-msi/msi-instrument, 2018.

[Age18b] European Space Agency. User Guides Sentinel-2 MSI Definitions. https://sentinel.esa.int/web/sentinel/user-guides/sentinel-2-msi/definitions, 2018.

[AHS+19] Heikki Astola, Tuomas Häme, Laura Sirro, Matthieu Molinier, and Jorma Kilpi. Comparison of sentinel-2 and landsat 8 imagery for forest variable prediction in boreal region. *Remote Sensing of Environment*, 223:257–273, 2019.

[Alg] Algorithmia. Introduction to optimizers.

[BCM12] Jon Atli Benediktsson, Jocelyn Chanussot, and Wooil M Moon. Very high-resolution remote sensing: Challenges and opportunities [point of view]. *Proceedings of the IEEE*, 100(6):1907–1910, 2012.

[cs215] cs231n. CS231n Course Materials. http://cs231n.github.io/neural-networks-1/#nn, 2015.

[DA18] Inc. Deep AI. Activation Function. https://deepai.org/machine-learning-glossary-and-terms/activation-function, 2018.

[DDBC+12] M Drusch, U Del Bello, S Carlier, O Colin, V Fernandez, F Gascon, B Hoersch, C Isola, P Laberinti, P Martimort, et al. Sentinel-2: Esa's optical high-resolution mission for gmes operational services. *Remote Sensing of Environment*, 120:25–36, 2012.

[GLL90] J. S. Griswold, T. L. Lightle, and J. G. Lovelady. Hurricane Hugo: Effect On State Government Communications. *IEEE Communications Magazine*, 28(6):12–17, 1990.

[HHH+09] Simon S Haykin, Simon S Haykin, Simon S Haykin, Kanada Elektroingenieur, and Simon S Haykin. *Neural networks and learning machines*, volume 3. Pearson Upper Saddle River, 2009.

[Ker] Keras Documentation. https://keras.io/.

[Mar17] Raphaël Marée. Cytomine overview. https://doc.cytomine.be/display/MAINDOC/Cytomine+overview, 2017.

[MRS+16] Raphaël Marée, Loïc Rollus, Benjamin Stévens, Renaud Hoyoux, Gilles Louppe, Rémy Vandaele, Jean-Michel Begon, Philipp Kainz, Pierre Geurts, and Louis Wehenkel. Collaborative analysis of multi-gigapixel imaging data using cytomine. *Bioinformatics*, 32(9):1395–1401, 2016.

[Tea19a] Cytomine Cooperative Team. Cytomine. https://cytomine.be/, 2019.

[Tea19b] Cytomine Cooperative Team. Cytomine Source Code. https://github.com/cytomine, 2019.

[ZWW15] Zhe Zhu, Shixiong Wang, and Curtis E Woodcock. Improvement and expansion of the fmask algorithm: Cloud, cloud shadow, and snow detection for landsats 4–7, 8, and sentinel 2 images. *Remote Sensing of Environment*, 159:269–277, 2015.