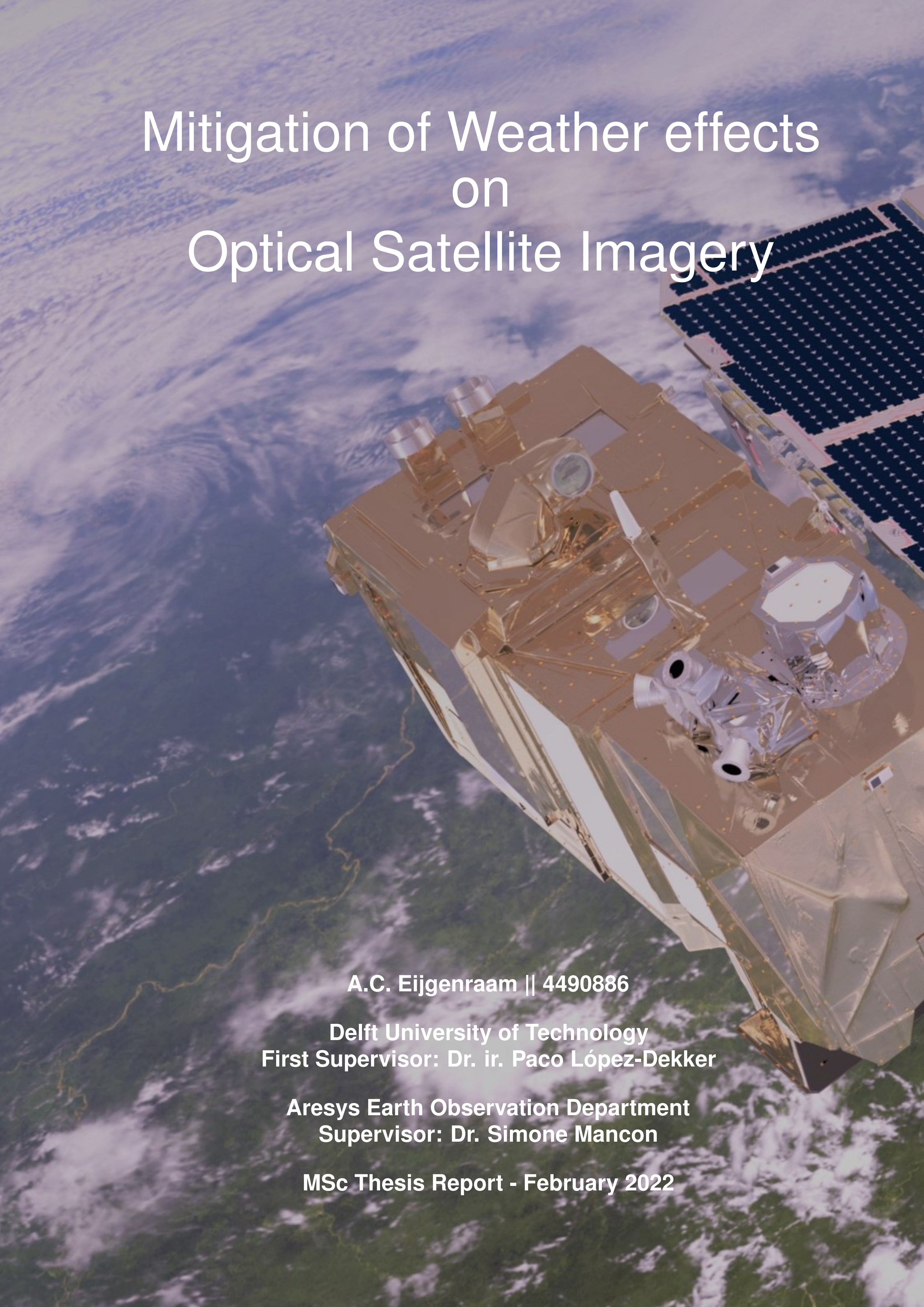


Mitigation of Weather effects on Optical Satellite Imagery

A.C. Eijgenraam || 4490886

Delft University of Technology
First Supervisor: Dr. ir. Paco López-Dekker

Aresys Earth Observation Department
Supervisor: Dr. Simone Mancon

MSc Thesis Report - February 2022

Mitigation of Weather effects on Optical Satellite Imagery

by

A.C. Eijgenraam

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Friday February 11, 2022 at 13:00 PM.

Student number:	4490886	
Thesis committee:	Dr. ir. Paco López-Dekker,	TU Delft, chair
	Dr. S. L. M. Lhermitte,	TU Delft
	Prof. dr. ir. R. F. Hanssen,	TU Delft

An electronic version of this thesis is available at `http://repository.tudelft.nl/`.
Cover photo by ESA-medialab (2015)

Abstract

How to deal with the presence of weather affected data is an unavoidable topic in the processing of optical imagery. Clouds and cloud shadows significantly alter the spectral signatures obtained from satellite data, which often leads to problems for any kind of scientific analysis. In this research there has been elaborated on two different kind of problems: The detection of clouds and cloud shadows and the mitigation of the effect caused by cloud shadows.

Most of existing operational cloud detection algorithms are so-called rule-based. Their performance is highly variable and they have their limitations. A new promising research was done by Mohajerani and Parvaneh (2019), where a convolutional neural network (CNN) named 'Cloud-Net' was developed. In this study we have elaborated on this CNN, by converting the analysis to Sentinel-2 data and making significant modifications on the model setup. The results have been compared to the ESA Scene Classification Map (SEN2COR algorithm). It was found that for the detection of clouds the overall CNN accuracy outperforms the ESA Scene Map (95.6% vs. 92.0% respectively). For the detection of cloud shadows the modified Cloud-Net model also gave better results (90.4% vs. 84.4%).

Previous work on cloud shadow correction algorithms show rather complex and inconvenient methods, where the only goal was to remove the effect of the shadow. If one is interested to also correct for illumination effects, to make it more aligned to a predetermined ground truth, new possibilities arise which allows for simpler and more direct methods. Two proposed methods have been investigated in this study. The first method, called 'decomposition of components', investigated the use of a single formula. The affected cloud shadow pixel is corrected based on the RGB difference with a ground truth image, and a single correction factor that was determined based under the assumption that cloud shadows cause a homogeneous alteration effect in a small area. The second method, called the 'CNN based method', presents a totally new idea by changing the Cloud-Net model to a regression model, in order to correctly alter cloud shadow affected pixels. The performance of both methods was quantified by the structural similarity index measure (SSIM). It was found that the decomposition of components method has the most potential, showing significant improvements on the correction of cloud shadow affected areas.

Preface

This Master Thesis covers the final chapter of my student life. It has been a great journey and I have learned a lot of things (or at least enough to graduate). My graduation project was not possible without the help of some people. First of all I want to thank my supervisor from Aresys, Simone Mancon. It was great to have a weekly call and your feedback was always very useful. I also had a great time in Milan and found it very nice that I was able to work from the office last summer. I also want to thank my supervisors from the TU Delft: Paco, Stef and Ramon. You really helped me on the documentation of my results and the way I should present my story.

Looking back on the past years, there have been a lot of cool projects I have participated in. The fieldwork in France was really fun and also the fieldwork trip to Iceland was very cool. The TU Delft offered me the opportunity to gather information from a wide field of disciplines. During my master I also did an internship at ING. Here I developed a new passion for machine learning. Because of this new knowledge I was able to combine CNN's with Remote Sensing for my thesis. A big thanks to Fabian Jansen for teaching me a lot on this subject.

Besides the technical support there was also the emotional support. I want to thank my family, friends and roommates for all their help and guidance. Because of you I had a really good time and I am happy to have you in my live. I am grateful for all the opportunities that were given to me and all the support I received over the years. My time in Delft has been great, and I am really looking forward to what the future will bring.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Cloud and cloud shadow detection

Clouds and cloud shadows are an unavoidable topic in the processing of optical imagery. To accurately detect these phenomena is a key component for scientific analysis. Clouds significantly alter the spectral signatures obtained from satellite data, which often leads to the false identification of land cover change (Tarrio et al., 2020). It is still not straightforward however how to develop good working cloud detection algorithms. This is due to their high spectral heterogeneity, and the spectral and temperature variability of the underlying surface. Most of the existing operational cloud detection algorithms, are so-called rule-based. This means that the corresponding algorithm depends on specific intensity thresholds, and rules that are carefully tailored for a given satellite platform and sensors. An approach like this is also sensitive to variations in atmospheric conditions, and to absolute surface reflectance values (Segal-Rozenhaimer et al., 2020). In the studies of Tarrio et al. (2020), Ipia et al. (2020) and Domnich et al. (2021) different cloud detection algorithms for Sentinel-2 imagery have been compared. In their comparison they all had three of these classical rule-based algorithms in common: FMask, MAJA and Sen2Cor. Three different labels were assigned to each scene: clear, cloud and cloud shadow. Tarrio et al. (2020) used a total of 6 Sentinel-2 scenes , distributed across the Eastern Hemisphere. Ipia et al. (2020) investigated 5 areas in the Amazon region, so an approach from a more regional point of view. Domnich et al. (2021) selected 21 scenes spread across Northern Europe. Table 1.1 shows a summary of the overall accuracies found in each study.

Table 1.1: Overall cloud and cloud shadow detection accuracies found in literature.

Algorithm	**Tarrio et al. (2020) acc.(%)**	**Ipia et al. (2020) acc.**	**Domnich et al. (2021) acc.**
FMASK	81.2	90	81
SEN2COR	77.8	79	82
MAJA	82.0	69	71

Comparing the results, the overall performance is quite different per paper. In some cases, a more than 10 percent difference is present. The most consistent algorithm is Sen2COR, varying in performance from 77.8 to 82 percent. The results of table 1.1 show that the performance of a rule-based algorithm is highly variable, depending on the type of scenes one chooses to evaluate the model on. These kind of rule-based models are designed in order to achieve the highest accuracy possible on a global scale. This means however that the accuracy can be highly variable from one region to another.

In most cases, one is interested in a limited amount of scenes to perform a cloud and cloud shadow detection on. It is possible to try out all of the rule-based models, and then select the one that works best. The problem here however, is that the accuracy that can be achieved is still limited. Also, if one selects multiple regions it is likely that multiple algorithms need to be used, since the suitability of each algorithm varies from one region to another. It is desired to have a cloud detection algorithm that is more robust than the rule-based algorithms, with a high accuracy and a low bias for the regions of interest.

A promising research has been done by Mohajerani and Parvaneh (2019), where a convolutional network called ’Cloud-Net’ was developed. An overall accuracy of 96.48% was achieved. This surpassed the

FMASK algorithm, which achieved an accuracy of 94.89%, on the same dataset. The idea is to elaborate on the work that has been done by the authors of this paper. A new version of Cloud-Net will be presented, and the results will be compared to the SEN2COR algorithm, or the so-called ESA Scene Map. From the literature, this model gave the most consistent results, and another advantage is that it is the most accessible one. In our study, the CNN has been used for the training of two binary classification models. One for thick clouds and one for cloud shadows. Thick clouds are defined as clouds where it is impossible to see any of the surface below. Although the main architecture of the network has remained intact, there are still some major other components that have been altered in this study.

The main difference, as already mentioned, is the training of two binary classification models. This has been done instead of the original setup, where one model was used for the classification of seven different classes. The second difference is that the training has been done on Sentinel-2 data instead of Landsat, which allows for a higher resolution analysis. Thirdly, different input bands have been used compared to the original model. In the study of Mohajerani and Parvaneh (2019), only the RGB-NIR band combination was tried. In this project, two different band combinations are used: RGB-NIR for cloud detection and RGB-NIR-SWIR for cloud shadow detection. The motivation will be elaborated on in chapter 3.3. Lastly, some main network hyperparameters have been changed like loss function, number of iterations, learning rate etc. The characteristics of the machine learning model, and the main differences with the classical rule-based algorithms, will be elaborated in chapter 2. After that a description of the data and the full method will be explained. A summary of the results will then be presented, and compared to the results of the ESA Scene Classification Map (SEN2COR algorithm) on the same dataset.

1.2 Cloud shadow mitigation

When an area is covered by thick clouds, the only option is to discard these clouds from the dataset. However, when there are thin or so-called cirrus clouds present, as well as cloud shadows, it is in some cases still possible to extract features from the data. In order to do this, a correction needs to be applied on the masked areas. In this project the mitigation of cirrus clouds will not be covered, but the cloud shadow issue has been investigated. The subject of cloud shadow mitigation on optical imagery is something that has been investigated before, by Nagare et al. (2018) or Shahtahmassebi et al. (2013) for example. Although The corrected images look better, some improvements can still be done. However, one big problem lies in the complexity of these algorithms. In the research of Nagare et al. (2018) for example, the method detects clouds in an image and derives their transmittance. For each cloud pixel, the method then locates the corresponding shadow pixel in the image plane. Next, it derives the Attenuation Factor for the direct Solar Irradiance (AFSI) value from the cloud transmittance values, and corrects the shadow pixel. Besides its complexity, it is also required that a cloud and the shadow coexist in an input image, which is not always the case. The reason behind this complexity is to make sure that the only correction that is applied, is the removal of the cloud shadow effect. However, for the end goal that is set for this project we are also interested in the modification of illumination effects, which means that there might be easier ways to overcome this cloud shadow issue.

This project has been done in collaboration with Aresys, on which some more information will be given in the next section. For one of their projects it is important to keep track of a certain commodity like coal or iron ore, and it is desired to monitor the structural changes over time. The commodity tracking algorithm is aligned to a certain set of conditions, so in order to achieve its highest potential not only the shadow effect needs to be removed, but also the illumination effects have to be modified, in order to make it fit the ground truth where the model is based on. Performing a RGB modification that does not only remove the shadow effect, but also corrects for the illumination difference with the clear ground truth day, allows us to formulate more simple and direct ways to deal with the described problem. Two different methods have been proposed, that have been investigated in this study.

The first approach is based on the decomposition of components that together form a cloud shadow ('decomposition of components method'). With decomposition we mean that a cloud shadow pixel can be reconstructed by a summation of different components. These components include the pixel values of the same area on a clear day, the illumination effect, and the cloud shadow effect itself. A RGB correction is carried out on a cloud shadow area, where this correction is based on the difference with a so-called ground truth image. This ground truth image is formed by taking the average RGB reflectances, from a combination of five clear days. The images on which the corrections are applied are relatively small (2.5 by 2.5 km), and therefore the assumption is made that the shadow effect in this area is the same for each pixel labelled as cloud shadow. From the median reflectance values for these pixels, a correction factor is retrieved. The final formula and setup of this experiment are described in chapter 3.7.

The second approach is a machine learning based method ('CNN based method'). With this method it is tried to overcome the requirement of a ground truth image, by performing a direct correction on the cloud shadow area. In order to achieve this, the Cloud-Net model has been changed from a classification to a regression model. For the training the ground truth values are needed, but we aim that after training the model knows how to correct cloud shadow affected areas, without having these reference values. Since Cloud-Net was originally designed for classification, there are probably other models that might work better. Due to the limited amount of time however, the same architecture has been used for this task, and it should be considered to be a first step in a new approach.

1.3 Collaboration with Aresys

This Master Thesis has been carried out in collaboration with the Earth Observation Department of Aresys, a company specialised in Remote Sensing located in Milan. They participate in system engineering and data analytics, for a wide range of geospatial applications, and have clients like ESA and Airbus.

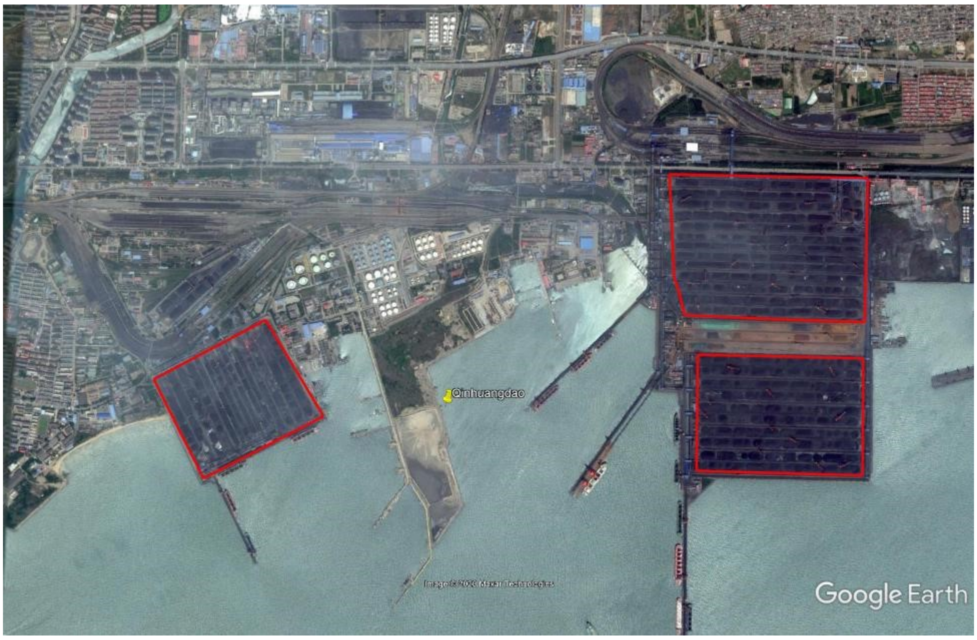


Figure 1.1: Google Earth image of Qinhuangdao port, China. Coal storages indicated within the red lines.

As already mentioned in the previous section, one of the projects they are currently involved in is the monitoring of stockpiles, like coal and iron ore. These stockpiles are monitored with the use of Sentinel-2 data, and they are stored in harbors like the Qinhuangdao port in China, as indicated in figure 1.1. The main goal here is to create accurate time series of the amount of a certain commodity. The benefits are clear, for replacing today's survey based methodology with a much faster method based on satellite observations. This will then allow stakeholders to get a more transparent and up-to-date view of the market.

The project is still in an early stage, but what is desired to have for Aresys is a model that can mask clouds and cloud shadows. Another useful tool is a possible correction model for the cloud shadows. This will be an important preprocessing step in order to obtain only clear, non-weather affected datasets. This is also where the theoretical and practical parts come together. From a scientific view it is interesting to improve on the detection of clouds and cloud shadows, on a local level. Also the new cloud shadow correction methods are something that can have added value. From a practical view it is useful, since the outcomes of this research can potentially improve on real world applications, like commodity tracking.

1.4 Research objectives

The main objective of this project is to find out if a CNN is able to outperform the SEN2COR algorithm on a local scale, and to define to what degree it is possible to correct the effect of cloud shadows. More specifically, this corresponds to the following goals:

- Make an adapted version of the Cloud-Net model that outperforms the SEN2COR algorithm by at least 1 percent, on the identification of clouds and cloud shadows on a Sentinel-2 dataset.
- Make an investigation of the suitability of different cloud shadow correction methods. Then define to what degree a correction is possible, and which method has the most potential.

The following research questions are proposed:

- Is it possible for a Convolutional Neural Network, to locally outperform the SEN2COR algorithm on the classification of clouds?
- Is it possible for a Convolutional Neural Network, to locally outperform the SEN2COR algorithm on the classification of cloud shadows?
- To what degree is it possible to correct for the effect of cloud shadows, and which method has the most potential?

1.5 Thesis Outline

As already described in the introduction, this thesis consist of two major parts: The identification of clouds and cloud shadows and the correction of cloud shadows. To begin with in chapter 2, some background information will be provided on the ESA rule-based classification algorithm. Then some general information on convolutional neural networks will be discussed, followed by a description of the Cloud-Net network that has been used in this research. Then in chapter 3 the complete methodology, for the cloud and cloud shadow classification and the cloud shadow mitigation, will be discussed. A description of the data will also be included. The results of the Cloud-Net algorithm and the cloud shadow corrections are presented in chapter 4. Chapter 5 consists of a discussion, followed by the conclusions in chapter 6.

2 Problem Background

2.1 SEN2COR: a rule-based classification algorithm

The Sentinel-2 Level-2A processing provides Level-2 products, from which one product is the so-called Scene Classification (SC). SEN2COR aims at providing a pixel classification map. Part of this SC map are the labels ’cloud shadows’ and ’cloud high probability’, which are the two targets we will also aim for with the Cloud-Net CNN. The SEN2COR algorithm is based on a series of threshold tests, that use as input top-of-atmosphere reflectance from the Sentinel-2 spectral bands. In addition, thresholds are applied on band ratios and indexes. The algorithm uses the reflective properties of scene features to establish the presence or absence of clouds in a scene. (ESA, 2013)

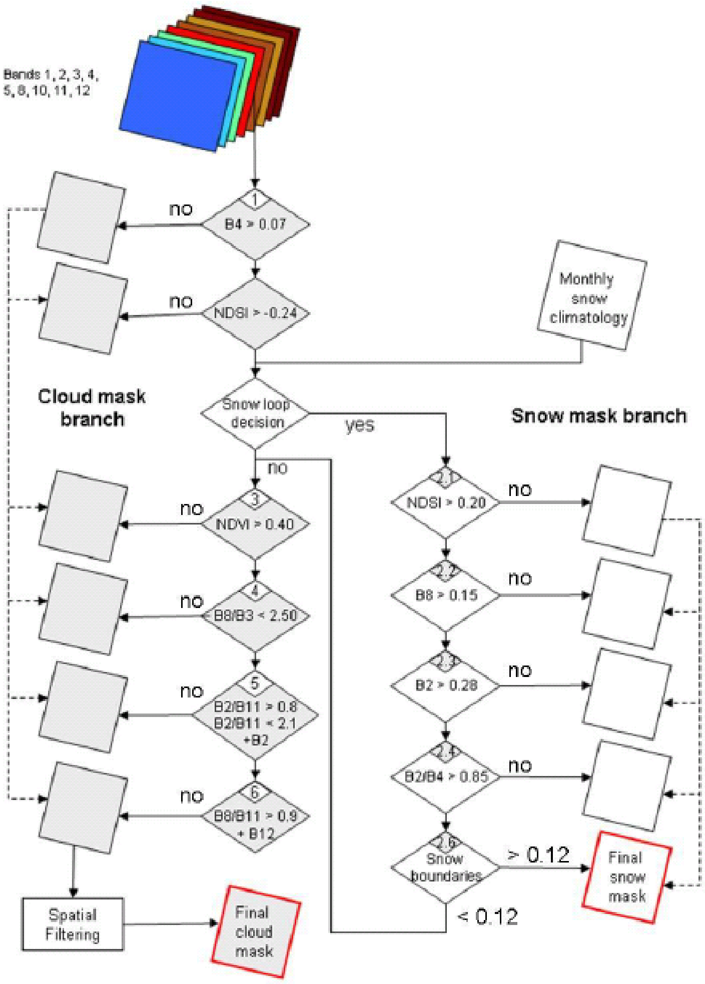


Figure 2.1: Level 2 Cloud/Snow Detection Algorithm Sequence (ESA, 2021).

Figure 2.1 shows the steps that are taken, in order to obtain a final cloud mask image. The thresholds are stated in the rhombus shapes. The empty boxes, like indicated on the left side, indicate that a pixel is considered cloud/snow free. In the literature it is stated that when a pixel is considered to be a potential cloud, and passes rhombus 6, it moves on to the spatial filtering box, after which the final cloud mask is created. Besides the detection of thick clouds, we are also interested in the detection of cloud shadows. Not all the details of the model will be discussed here, but the cloud shadow mask is constructed using "geometrically probable" cloud shadows. These are derived from the final cloud mask, sun position and cloud height distribution, and "radiometrically probable" cloud shadows derived from the radiometric properties of the pixels (Louis, 2021).

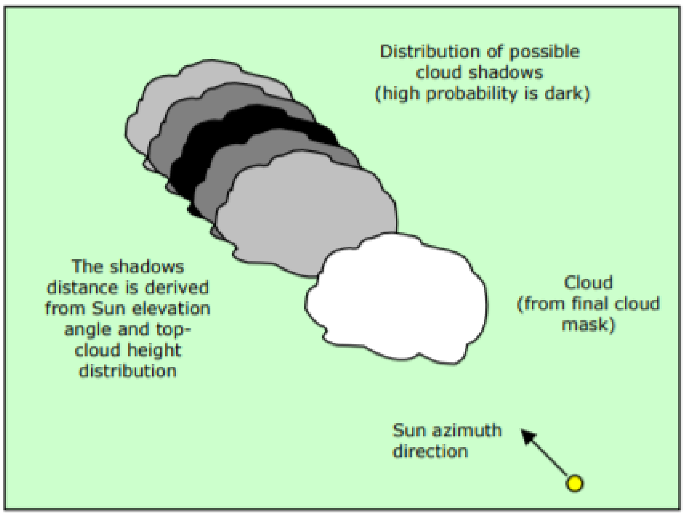

Distribution of possible cloud shadows (high probability is dark)
The shadows distance is derived from Sun elevation angle and top-cloud height distribution
Cloud (from final cloud mask)
Sun azimuth direction

Figure 2.2: Illustration of geometrically cloud shadow probability model (Louis, 2021).

Figure2.2 illustrates the idea behind the construction of the geometrically probable cloud shadows. Based on the sun's azimuth direction and the distance from the cloud, a cloud shadow probability is defined. The light grey clouds indicate a low cloud shadow probability, the black clouds a high cloud shadow probability.

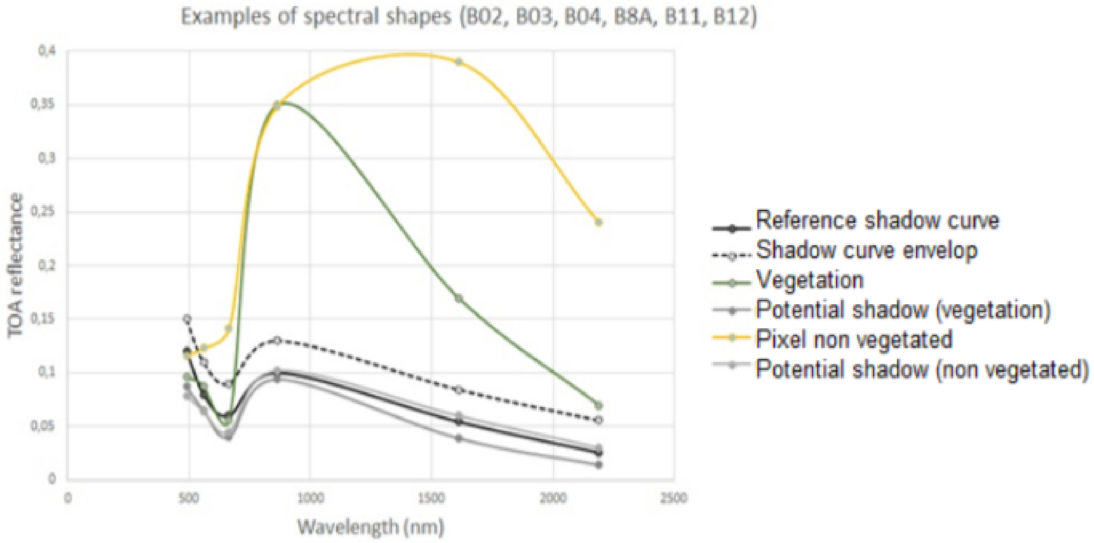


Figure 2.3: Illustration of radiometric cloud shadow probability model (Louis, 2021).

The idea behind the radiometric cloud shadow probability model is shown in 2.3. The radiometric input is obtained by identifying potential cloud shadows or "dark areas", based on a reference shadow spectral shape defined with bands B02, B03, B04, B08A, B11 and B12. This reference dark spectrum was built from a large range of cloud shadows examples on different type of land covers. Pixels with a spectrum close or darker than the reference shadow spectral shape, are considered as potential cloud shadows.

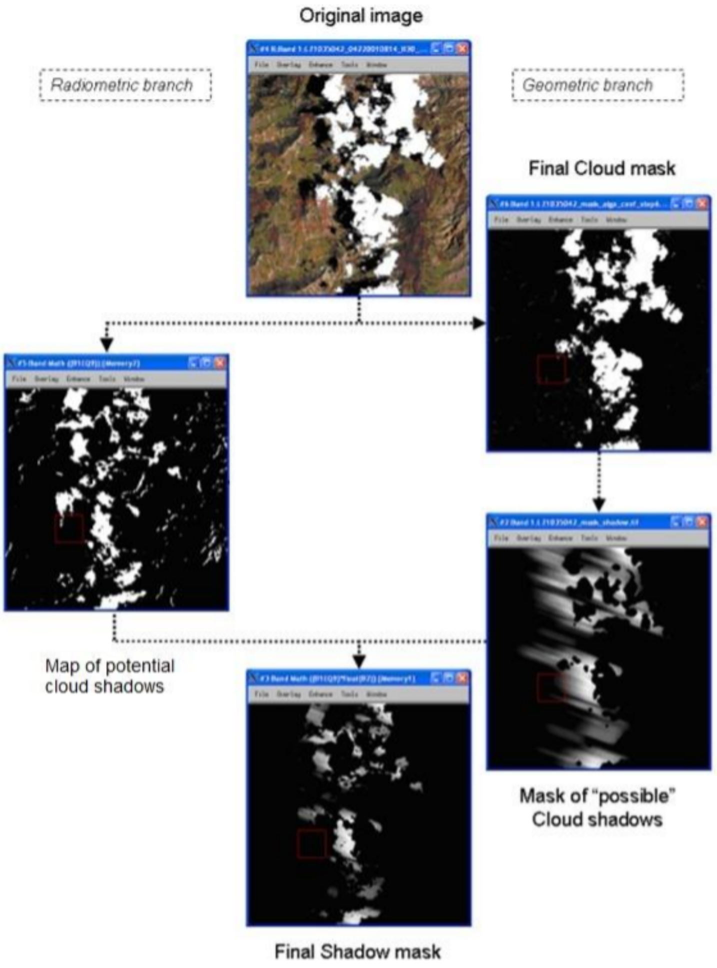


Figure 2.4: Schematic view of the algorithm for cloud shadow mask generation (Louis, 2021).

The sequence of processing steps to generate the final cloud shadow mask is shown in Figure 2.4. The final cloud shadow mask is obtained by multiplying the result of the radiometric branch (left side) by the result of the geometric branch (right side).

2.2 Cloud-Net: Convolutional Neural Network for the detection of clouds and cloud shadows

Cloud-Net is a convolutional neural network (CNN), developed by Mohajerani and Parvaneh (2019). It was designed to perform pixel classification on satellite images, originally trained and applied on Landsat data. To completely cover an explanation on the working of a CNN, is too much to cover for this project, but some basic principles are good to know. A basic explanation is covered in the next sections, followed by a more in deep explanation of some Cloud-Net properties.

A CNN is composed of two major parts:

- Feature extraction
- Classification

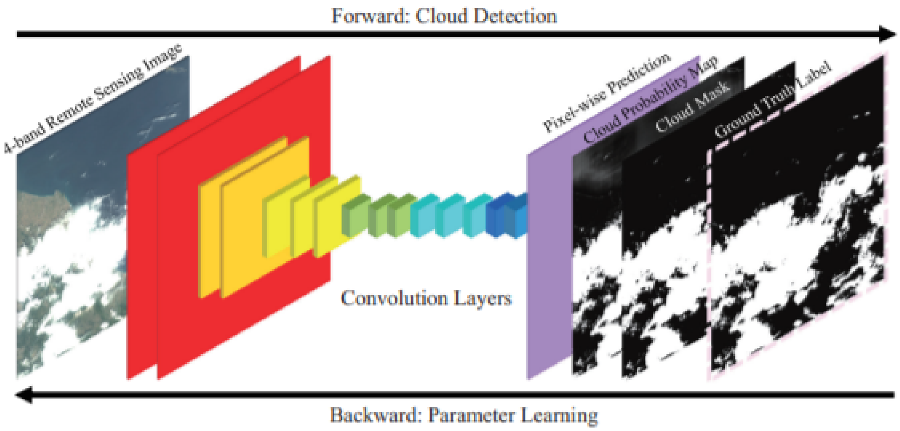


Figure 2.5: Example of a CNN for the detection of clouds (Wu & Shi, 2018).

Feature extraction

From this part of the architecture the network derives its name. Convolution in a CNN is performed on an input image, using a filter or a kernel. This filter, which is a small matrix of values, performs a mathematical operation on the input image sliding from the left top to the right bottom. This results in a so-called feature map. In a single network multiple feature maps are generated, as can be seen in figure 2.5. On each feature map, an activation function and a possible pooling layer (not further discussed here) are then applied. The output will then be used as input for the next layer. The different feature maps try to capture the different properties that describe the label the model is trying to predict. In figure 2.5 for example, the model tries to predict for each pixel whether it is a cloud or not. The different filters will try to capture the different features. These filters, together with a corresponding bias, are random for the first iteration. They are then updated after one full cycle of computations throughout the network. These filter values, together with the biases are called the weights.

Classification

The final output layer gives the output probabilities for each class. In case of the cloud detection example shown in figure 2.5, this will range from 0 to 1. Since weights are randomly assigned for the first training example, output probabilities are also random. The probabilities that are predicted by the networked are then compared to the true values, or the so-called targets that the network should predict. These often involves a lot of manual labelling, so assigning the corrected labels to each image that one wants to use as input for the training. The error of the CNN prediction is then computed, for which different formula's or so-called loss functions can be used. A common loss function is the mean squared error, which gives the following formula:

$$\text{Total error} = \sum \sqrt{(\text{target value} - \text{output probability})^2} \tag{2.1}$$

After the error is computed, backpropagation is used in order to calculate the gradients of the error, with respect to all the weights in the network. Then gradient descent is used to update all filter values and biases, to minimize the output error. The weights are adjusted in proportion to their contribution to the total error. If the network is capable of learning to predict the correct values associated with certain input data, the output error should keep decreasing and is expected to converge to a minimum. How many iterations are needed in order to reach a minimum is highly variable per case, and is depending on many factors, like the quality of the network, quality of the data, correlation between the features and the target, etc.

A process that is common for CNN's is called hyperparameter tuning. This is a process, where the main network parameters are optimized. These parameters include batch size, learning rate, number of iterations etc. There are different ways to perform hyperparameter tuning, one of the most common ones is called grid search. An elaboration on this grid search and the implementation in this study is described in chapter 3.3.

Cloud-Net architecture

Now that the general properties of a CNN have been discussed, some specifics of the Cloud-Net model will be stated in this section. As described earlier, the Cloud-Net model is able to perform end-to-end pixel classification on satellite imagery. A total of 12 hidden layers is present, as visualized in figure 2.6.

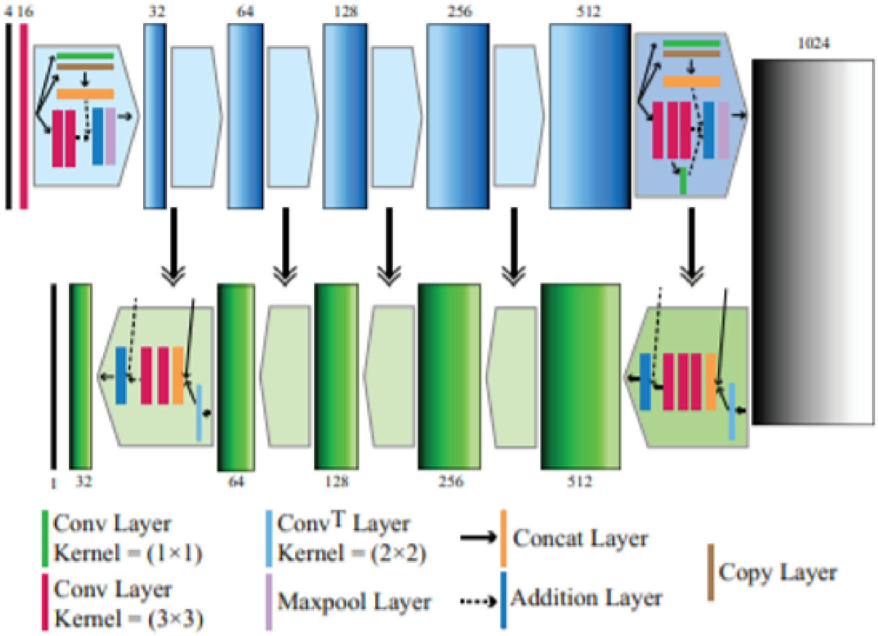


Figure 2.6: Architecture of the Cloud-Net model (Mohajerani & Parvaneh, 2019).

This is a so-called U-Net architecture. The design was originally proposed in Ronneberger et al. (2015), and was used for biomedical image segmentation. U-Net has proven to give high performance results, and over the years a lot of variations have been proposed for a wide range of applications. One of the advantages of using a U-Net structure is that it works with very few training samples, and it gives high performance for segmentation tasks. Secondly, an end-to-end pipeline processes the entire image in the forward pass, and directly produces segmentation maps. Because of this U-Net preserves the full context of the input images, which is a major advantage when compared to patch-based segmentation approaches (Alom et al., 2018).

For the input it is required to select a combination of x different bands, in the original paper 4 was used. From this 16 output channels are created after the first convolution, which is achieved by the use of 16 filters, each filter having a x dimensional shape. As can be seen in the top part the schematic overview, the number of output channels in the next hidden layer is doubled and this process is continued to a total of 1024. This part is called the contracting arm and is responsible for extracting features and producing deep low-level features of the input image. The bottom part of the architecture is the other arm and is called the expanding arm. This is designed to utilize those features and retrieve cloud and cloud shadow attributes, recover them, and finally generate an output mask in the last layer.

Figure 2.6 shows down pointing arrows drawn from the contracting arm to the expanding arm. These are so-called short cut connections. Before moving to the next layer in the contracting arm, the feature map from the expanding arm is added (the shapes are the same which makes this possible). This helps the network to preserve and utilize the learned contexts from the earlier layers. As a result, the network is capable of capturing more cloud features. Another effect of the short cut connections is that the training process is accelerated, by preventing the network from experiencing the vanishing gradient phenomenon during backpropagation. The vanishing gradient phenomenon occurs, when many components of the gradient of the loss function get very close to zero. This causes a stall in the updates of the parameters associated with these layers, since the algorithm uses the gradient to calculate its next step. The problem also tends to get worse as the number of hidden layers increases (Roodschild et al., 2020).

Another important property of the Cloud-Net model is the drop-out function that is applied on the output of hidden layer 7, the step moving from the expanding arm to the contracting arm. The idea behind this dropout function is to prevent the network from over-fitting. Over-fitting is a common issue in networks that have many hidden layers and therefore a lot of parameters. Dropout is a technique for addressing this issue. This technique was successfully applied with several types of neural networks and it shows significant improvements (Srivastava et al., 2014). In Cloud-Net a 2D dropout function is used, which means that instead of single elements, entire channels are zeroed out. The motivation behind this is described in Tompson et al. (2015). Since our network is fully convolutional and natural images exhibit strong spatial correlation, the feature maps are also strongly correlated, and in this setting standard dropout fails. In Cloud-Net, the probability p that a channel is set to zero, is set to 0.15. This value is preserved in this study.

Selection of Sentinel-2 input bands

The original Cloud-Net model uses a four channel input from the Landsat 8 satellite: The blue,green,red and near infrared (NIR) bands. It was assumed by the authors that this combination would work best. One can try different band combinations however, and also the number of input bands is variable. Table 2.1 shows an overview of all the Sentinel-2 bands present.

Table 2.1: Overview of available Sentinel-2 bands.

Sentinel-2 bands	Central wavelength (µm)	Resolution (m)
Band 1- Coastal aerosol	0.443	60
Band 2- Blue	0.490	10
Band 3- Green	0.560	10
Band 4- Red	0.665	10
Band 5- Vegetation Red Age	0.705	20
Band 6-Vegetation Red Age	0.740	20
Band 7- Vegetation Red Age	0.783	20
Band 8- NIR	0.842	10
Band 8A- Vegetation Red Edge	0.865	20
Band 9- Water vapour	0.945	60
Band 10- SIR-Cirrus	1.375	60
Band 11- SWIR	1.610	20
Band 12- SWIR	2.190	20

Table 2.1 shows an overview of all the Sentinel-2 bands, with their corresponding central wavelength and resolution. A high resolution is desired to have. Band 2,3,4 and 8 (RGB+NIR) are all 10m resolution bands, which would therefore be suitable input bands for the Cloud-Net model. Band 5,6,and 7 have a 20m resolution, but are focused on vegetation monitoring. Bands 9 and 10 have a resolution of 60m, which is considered too low. Bands 11 and 12 are two short wave infrared (SWIR) bands with a resolution of also 20m, and are therefore also considered to be potential useful bands for the CNN. Based on this information, a total of 6 bands is considered potentially suitable: B02, B03, B04, B08, B11 and B12.

The first band selection that needs to be defined is for the cloud detection model. It is well known that clouds are clearly distinguishable in the visible spectrum, therefore the RGB channels will be included. In Tan and Tong (2016) it is argued that cloud regions usually have higher NIR values than that of non-cloud regions. Therefore also the NIR band is included. In Baetens et al. (2019) it is explained that the SWIR bands are useful for separating clouds from snow, where clouds will show higher SWIR values than snow. In this research, there is no snow present in the data. Therefore these bands are not considered useful for cloud detection here. In the literature, there were no other functions found for the SWIR bands, regarding cloud detection. Therefore the final bands that will be used are: B02, B03, B04 and B08.

The second band selection, concerns the cloud shadow detection model. Just like clouds, cloud shadows are also clearly distinguishable in the visible spectrum. In Zhu and Helmer (2018) it is stated that for cloud shadows, direct solar radiation is blocked by clouds, so the shadow pixels are illuminated by scattered light. Because the atmospheric scattering is weaker at longer wavelengths, the NIR and SWIR bands of shadow pixels are much darker than surrounding clear pixels. Therefore it is assumed, that the NIR and SWIR bands contain useful information, in order to detect cloud shadows. This means that a total of six bands will be used for the CNN model: B02, B03, B04, B08, B11 and B12.

3 Methodology and data

3.1 Cloud and cloud shadow detection flowchart

In this chapter, the methodology of the training of a CNN that was followed will be discussed, as well as the correction of the cloud shadows. The process of cloud and cloud shadow detection is visualized in figure 3.1.

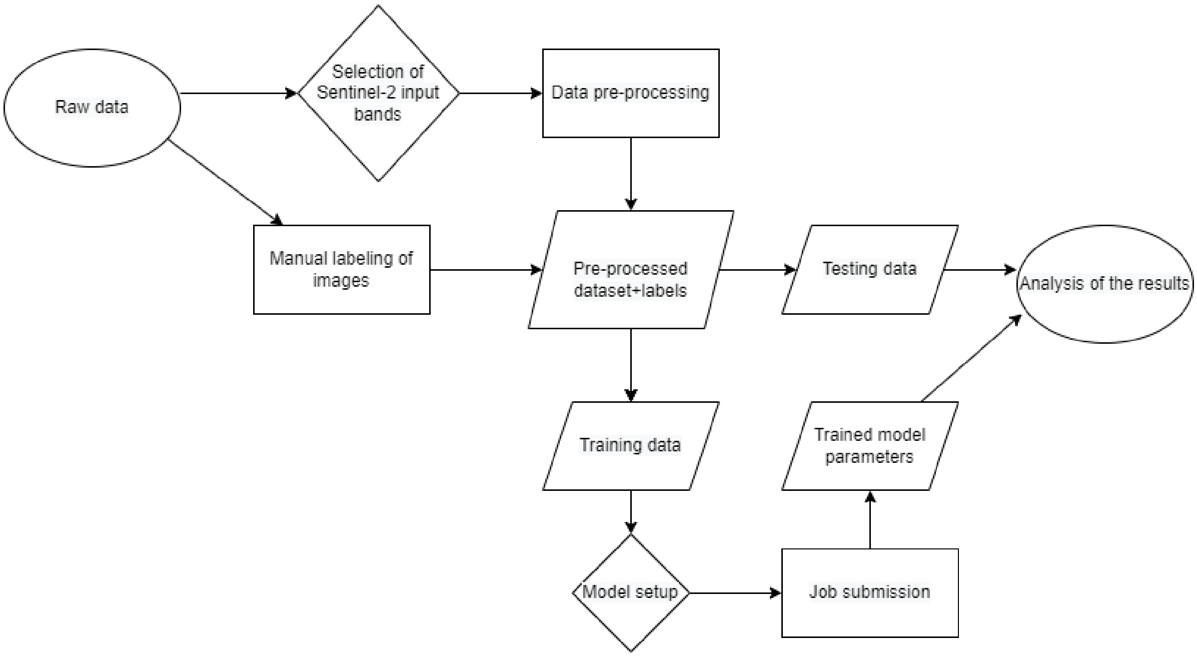


Figure 3.1: Flowchart for the cloud and cloud shadow detection analysis.

The flowchart follows the traditional approach of a supervised machine learning model. After obtaining the raw data, the first that needs to be done is labelling the images and pre-processing the dataset. Then the pre-processed data is split into training and testing data. After defining the model setup, training can be done. Then the trained parameters are applied on the testing data, which will allow for a final analysis on the performance of the model. Some of the steps stated in the flowchart, will be elaborated on in the next sections.

3.2 Cloud and cloud shadow detection data

1) Data collection

The Sentinel-2 data that has been used in this research is downloaded from Sentinel Hub EO Browser (Sentinel-Hub, 2021). The highest quality products have been selected, and these are the level 2A products. Six different locations have been chosen, shown in figure 3.2.

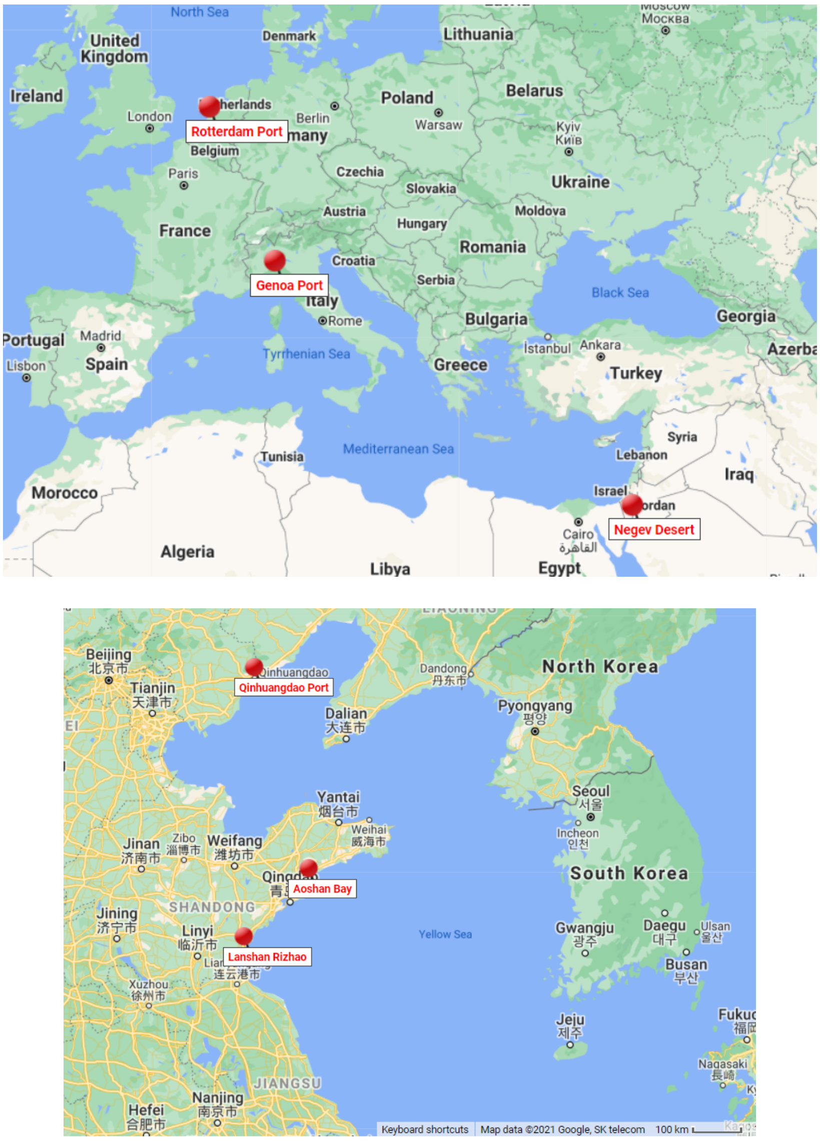


Figure 3.2: Areas of interest, for both training and testing for the detection of clouds and cloud shadows. 6 locations in total: Rotterdam Port (Netherlands), Geonoa Port (Italy), Negev desert (Israel) Qinhuangdao Port (China), Aoshan Bay (China) and Lanshan Rizhao (China).

Multiple locations and dates have been selected. In this study, we also try to perform a cloud shadow mitigation experiment, and for this an area with stable targets is needed (the details behind this part will be explained in chapter 3.6). These stable targets were found in the Negev desert. The other selected areas are ports or areas close to a port.

2) Preprocessing

Before feeding the data to the network, we need to preprocess the data. The first thing we would like to do is to apply a normalization. For neural networks it is convenient to have input values between 0 and 1 (Brownlee, 2020). A way to achieve is to use the following normalization:

$$z = \frac{x - \min(x)}{\max(x) - \min(x)}. \tag{3.1}$$

A second step is to divide the images into smaller sub images. This in order to stack the different images together, forming the final so-called tensor. In order to form this tensor the dimensions of the different images need to agree. Each image has been divided into sub images of 256 by 256 pixels. This is approximately 2.5 by 2.5 km.

3) Manual labeling of images

An important step in the cycle of the CNN training, is to acquire labelled data. For this project we have decided to do manual labeling. This allows for more freedom on how to classify certain features, and since there are some areas that are of greater interest for Aresys, it leaves also wider choice on the selection of which datasets will be used for training. The annotation tool that has been used is called CVAT (Computer Vision Annotation Tool), and is open source and free to use (Sekachev et al., 2020). Figure 3.3 shows an example of how the annotation is done.

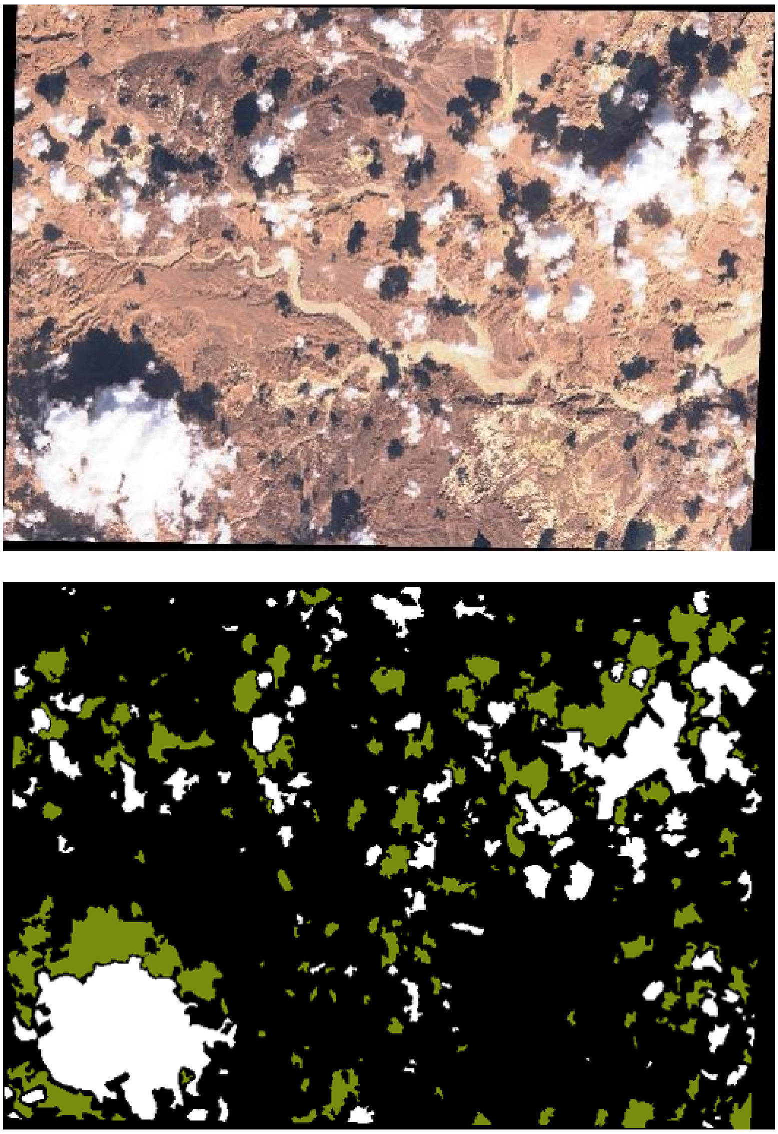


Figure 3.3: Example of label annotation input (top) and output (bottom) in CVAT, corresponding to a Negev desert scene. In the bottom image white areas indicate annotated clouds, green colored areas indicate cloud shadows.

4) Training and testing data

The dataset needs to be split in two parts, with one part for training and one for testing. This will be done two times, so two different training/testing sets will be constructed, with one for the cloud detection and one for the cloud shadow detection. This has been done because the features (clouds and cloud shadows) are not distributed evenly over the entire dataset. There is a total of 236 individual tensors or images, that can be chosen from to use for either training or testing.

Table 3.1: Overview of all the scenes, used for the classification of clouds and cloud shadows. The number of training (tr) and testing images (tst) are indicated for each scene.

Location	Date	Left top coordinate	Cloud tr	cloud tst	shadow tr	shadow tst
Qinhuangdao	2017-08-13	119.6717,39.9443	0	1	0	1
Qinhuangdao	2018-06-14	119.6724,39.9592	2	0	2	0
Qinhuangdao	2019-03-31	119.6737,39.9427	-	-	0	1
Qinhuangdao	2019-08-30	119.6519,39.9576	3	0	2	0
Qinhuangdao	2019-07-14	119.6717,39.9443	-	-	1	0
Qinhuangdao	2021-05-01	119.0475,39.8755	9	2	-	-
Qinhuangdao	2021-05-11	119.6724,39.9592	-	-	1	0
Aoshan Bay	2020-09-28	120.6652,36.5542	13	2	14	1
Aoshan Bay	2019-12-19	120.7517,36.4979	-	-	0	2
Genoa	2020-09-28	8.9288,44.4309	0	3	0	3
Genoa	2020-12-07	8.9288,44.4305	0	1	3	0
Genoa	2021-03-19	9.5821,44.4269	28	0	27	3
Lanshan Rizhao	2019-11-03	119.3094,35.3033	6	0	2	0
Lanshan Rizhao	2020-07-10	119.3496,35.1355	0	2	-	-
Lanshan Rizhao	2020-07-20	119.3496,35.1355	-	-	3	0
Lanshan Rizhao	2021-02-15	119.3312,35.1600	1	0	0	1
Lanshan Rizhao	2021-04-16	119.1824,35.3014	-	-	6	0
Negev Desert	2021-01-20	34.7399,30.4571	21	0	26	0
Negev Desert	2021-02-24	35.1854,30.3379	0	8	0	9
Negev Desert	2021-04-10	35.5567,30.4420	-	-	4	0
Rotterdam	2021-03-26	4.2516,51.8848	2	0	-	-
Rotterdam	2021-05-30	4.2516,51.8848	0	1	0	1
Total	-	-	85	20	91	22

Table 3.1 gives an overview of the entire dataset, and how the selection for training and testing data has been made. The procedure that was followed, was to individually analyze each image and add it to the cloud or cloud shadow training/testing set, when around 20 percent of image was covered by either one of these features (with the exception of a few images), and the label annotation was done correctly. These constrains resulted in 105 useful images for cloud detection and 113 for cloud shadows. A common training/testing ratio is 80-20 (Kalkman, 2021) which has also been used in this study.

Table 3.2: Number of pixels labelled as cloud and cloud shadow.

	pixels labelled as cloud (%)	pixels labelled as cloud shadow (%)
Training	27	21
Testing	21	21

The supercomputer that has been used is called Lisa (SURFsara, 2021). The Lisa system is a cluster computer consisting of several hundreds of multi-core nodes, running the Linux operating system. The system is installed and maintained by SURFsara. The Lisa system is used for the SURFsara service Research Capacity Computing Services (RCCS).

3.3 Selection of model parameters

Some main features of the original Cloud-Net model have been preserved. These include the number of hidden layers, pooling functions, activation functions, etc. There are other model parameters however that one can change, and this has also been done for the training of this network. An overview of these so-called hyperparameters are given in table 3.3.

Table 3.3: Neural network parameter settings.

batch size	5
Loss function	Binary cross entropy loss
Number of iterations	7000
Learning rate	0.0001
Optimizer	Adam

Grid search is a very computationally expensive process, and the budget available on the cluster computer is limited. Still some different combinations regarding batch size, number of iterations and learning rate have been tried. For the batch size 3,5 and 8 have been tried. For the number of iterations 5000,7000 and 9000. For the learning rate 0.001, 0.0005 and 0.0001. The options that worked best are given in table 3.3. An important note to make, is that the original Cloud-Net model was designed to classify 6 classes. In this project two binary classification models are generated, which makes it more logic to switch the loss function to binary cross entropy loss. The last parameter stated above is the optimizer. The Adam optimizer has been selected, which is an extension to stochastic gradient descent. It has recently seen broader adoption for deep learning applications in computer vision, and natural language processing (Brownlee, 2021).

3.4 Analysis of the results

After the output files have been downloaded from the Lisa server, the results can be analyzed. These results are also compared to the output given by the SEN2COR algorithm, or the so-called ESA Scene Classification Map. The weights from the trained CNN model are applied on the testing set. In combination with the labels, this then allows us to define an overall accuracy of the model

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}\ , \tag{3.2}$$

where TP is the number of true positive pixels, TN is the number of true negative pixels, FN is the number of false negative pixels and FP corresponds to the number of false positive pixels. These classes will also be represented in a confusion matrix.

Besides accuracy, there are also other quality indicators that will be used: precision, recall and F1 scores. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}\ . \tag{3.3}$$

Recall is the ratio of correctly predicted positive observations to the all observations in actual class - true:

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}\ . \tag{3.4}$$

Then finally, the F1 scores will be computed. This is a so-called harmonic mean of precision and recall:

$$\text{F1 score} = \frac{2 * (\text{recall} * \text{precision})}{\text{recall} + \text{precision}}\ . \tag{3.5}$$

3.5 Workflow cloud shadow mitigation

For the cloud shadow mitigation, there are two proposed methods. A flowchart with both the decomposition of components and CNN-based method, are given in figure 3.4.

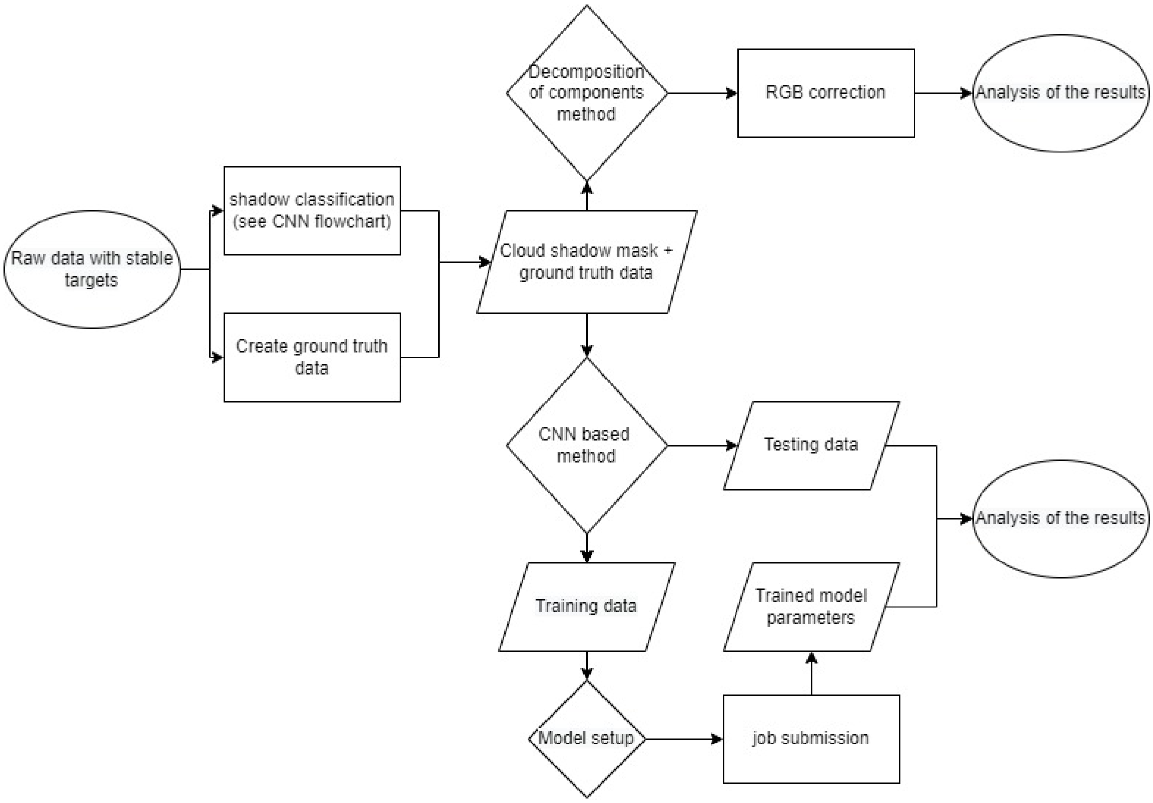


Figure 3.4: Flowchart for the mitigation of cloud shadows.

The first step is to find stable targets and to collect data. From this raw dataset a cloud shadow mask needs to be performed, and a corresponding ground truth image needs to be created. The masking of cloud shadows will be done with the trained CNN model.

The decomposition of components method is shown in the top arm. The formula that was used will be explained in section 3.7. The bottom arm corresponds to the CNN based method. Here the steps are similar to the classification flowchart. The data is split into training and testing data, and after training has been done, the trained parameters combined with the testing data allow us to analyze the results.

3.6 Cloud shadow mitigation data

1) Motivation for area of interest

As described in the introduction, it is desired to have stable targets that do not change over time. It was also already discussed that these targets were found in the Negev desert, but the motivation will be given here. The main reason for selecting this area, is the remote location where no human intervention is present. Within this large area, smaller areas have to be selected. The choice was made to select regions where stable rocky mountains were present.

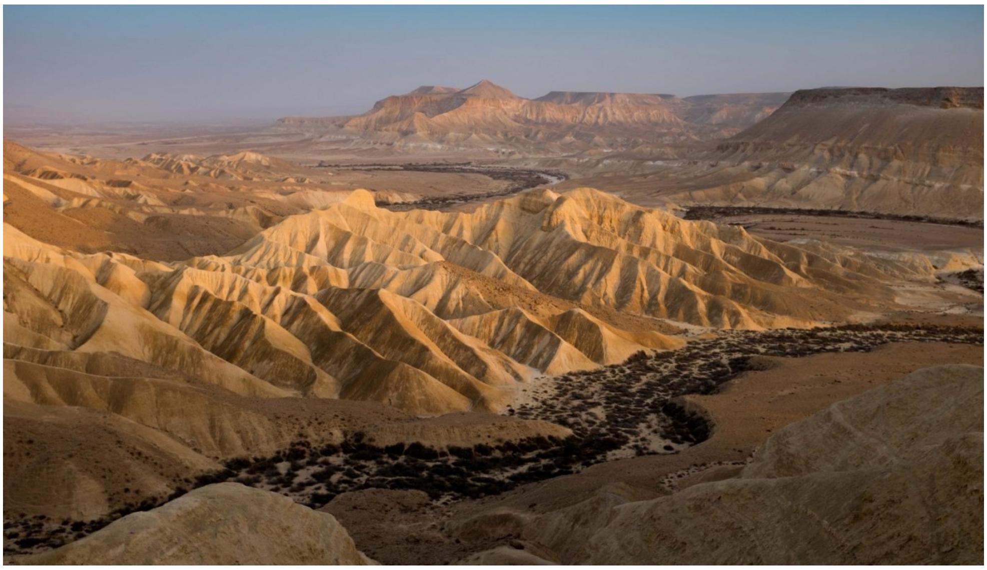


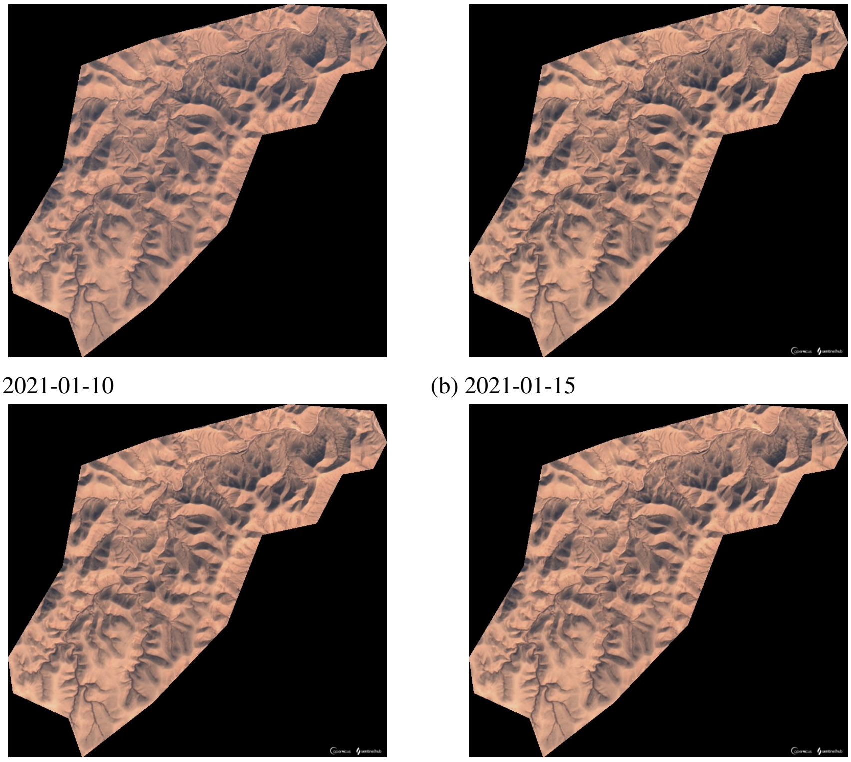
Figure 3.5: Impression of the Negev Desert, Israel (Giamberini & Provenzale, 2018).

Figure 3.5 gives an impression of these rocky, dusty mountains that are located in the Negev desert. As can be seen from this figure, the geometry of these mountains show strong similarities with the geometry of a commodity storage location. One important difference however is that these Negev mountains are stable over time. In order to verify this, a preliminary analysis has been carried out where the RGB spectrum of 4 different clear days have been compared.

The method that has been selected, in order to compare the similarity between two images, is called the structural similarity index measure or 'SSIM'. The measure between two windows x and y of common size N×N is (Mamun, 2019):

$$\mathrm{SSIM(x,y)} = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 +_y^2 + c_2)} \,. \tag{3.6}$$

Where μ_x is the average of x, μ_y the average of y, σ_x^2 the variance of x, σ_y^2 the variance of y, σ_{xy} the covariance of x and y, c_1 and c_2 are two variables to stabilize the division with weak denominator. This method is a common quantitative way of measuring the similarity of two images. The full background of this formula will not be discussed, but the main idea is that the outcome of this formula is a product of luminance, contrast and structure. The outcome is a unitless number between -1 and 1, where 1 means that two images are identical, and -1 that the match is perfectly imperfect (Datta, 2020). Consider the results of one scene where four different dates have been selected.



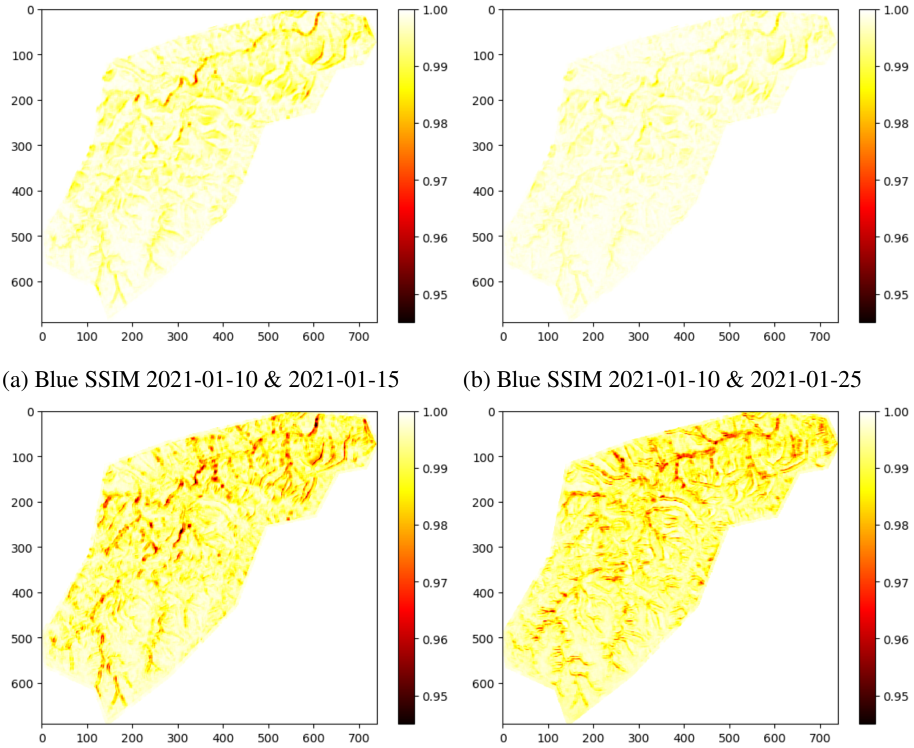
(a) 2021-01-10

(b) 2021-01-15

(c) 2021-01-25

(d) 2021-01-30

Figure 3.6: Sentinel-2 true color images taken on 4 different dates of a roughly 7 by 3 km area in the Negev desert.



(a) Blue SSIM 2021-01-10 & 2021-01-15

(b) Blue SSIM 2021-01-10 & 2021-01-25

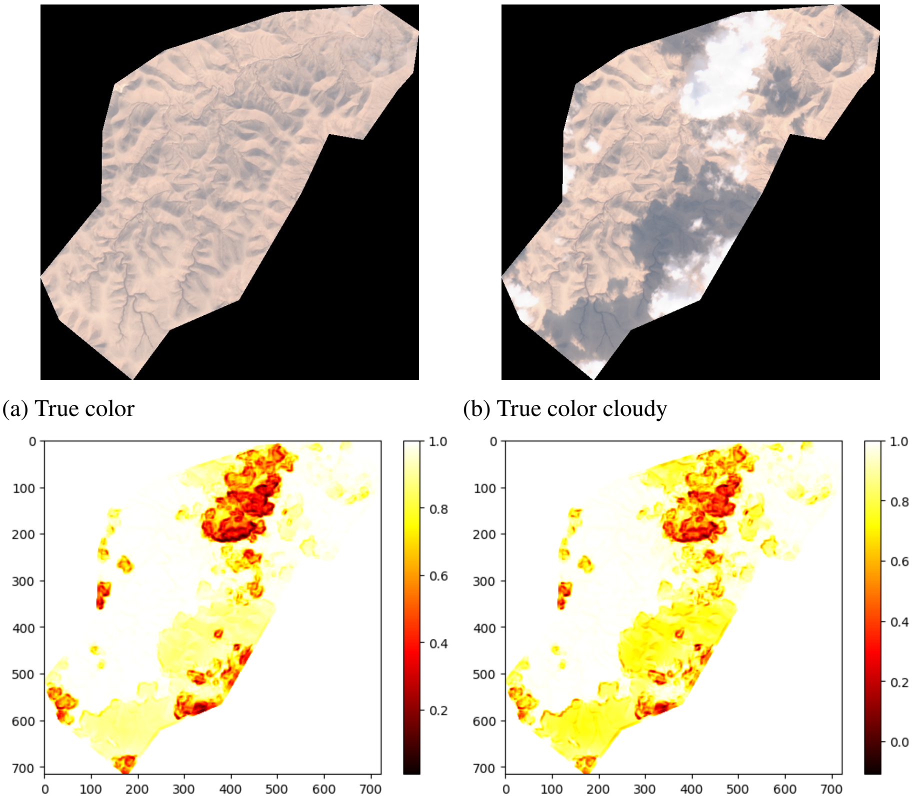
(c) Red SSIM 2021-01-10 & 2021-01-15

(d) Red SSIM 2021-01-10 & 2021-01-30

Figure 3.7: Structural similarity index measure of 4 different scenes at the Negev desert. Top 2 images display a blue SSIM analysis and the bottom two a red SSIM analysis.

Figure 3.7 shows 4 sub figures of the Negev desert, where the structural similarity index measure (SSIM) has been examined on 4 different scenes, shown in figure 3.6. The top two images show two blue SSIM analyses, and the two bottom images contain two red SSIM analyses. All four images show SSIM values close two 1. From this it can be concluded that the target is stable and can therefore be used for the cloud shadow mitigation experiments.

Next to the fact that it can be demonstrated when two images are similar, we also like to prove when two images are clearly not identical. Let's therefore analyze the same scene, but with one cloudy and one clear day:



(a) True color

(b) True color cloudy

(c) Blue SSIM

(d) Green SSIM

Figure 3.8: Structural similarity index measure of a cloudy versus a clear day. A blue and a green SSIM analysis are presented.

Figure 3.8 shows the blue and green SSIM outcomes corresponding to the cloudy vs. non-cloudy day. From these images it is easy to indicate at which areas a cloud or cloud shadow is present. From this an important second conclusion can be stated, that whenever a cloud shadow is present, the SSIM analysis is able to easily capture this feature.

2) Data collection & ground truth creation

In order to perform the cloud shadow mitigation experiment, the next step is to collect all the data. The first scene that will be used is the example area of the previous section, including the black spots. A day with quite some cloud shadows present was found on April 1, 2019. The second scene is the region displayed in figure 3.3 that was taken on January 20,2020.

The ground truth is then computed by finding 5 clear days of both scenes and taking the average RGB values. Both scenes have thereafter been divided into smaller images of 512 by 512 pixels. The first scene then generates 1 image and the second one 12 images.

3.7 Decomposition of components method for cloud shadow mitigation

The decomposition of components method aims to perform a RGB correction, by the use of a single computation, on areas that have been labelled as cloud shadow. The motivation behind the used formula and a description of the process, will be given in this section. A cloud shadow affected pixel can be described by the following formula:

$$\text{RGB affected} = \text{RGB clear} + \Delta\text{RGB shadow} + \Delta\text{RGB illumination} + \Delta\text{RGB structural} \quad (3.7)$$

The composition of this formula is quite straightforward. Whenever a cloud shadow pixel is present, this pixel can be reconstructed by a combination of the clear non- weather affected RGB value that can be found in a clear image, the alteration caused by the shadow, the illumination difference with the clear day and any other structural changes of a target. Structural changes can be any change that causes a permanent alteration of a pixel. When there is a load of coal stockpiles present in a harbor for example, a pixel will be permanently altered, whenever a new load is added or when some of it is removed. For this experiment, it is desired to make this term negligible. This can be done by finding a stable target that doesn't change over time, like explained in the previous section. That a target can change is something that needs to be taken into account however, or more concrete: how to make sure that a RGB correction on a cloud shadow pixel is a correction taken only the shadow and illumination effect into account? This is something that can be achieved by taking the median cloud shadow value of an image into account. In many images the cloud shadow looks like a homogeneous feature. Therefore the assumption is that the shadow effect in an image can be considered the same. Since each image is only a 2.5 squared kilometer area this assumption can hold ground. The next step is to formulate this assumption into a formula that can be applied to actually perform the correction. The Δ RGB shadow term is something that cannot be found directly but we can take the median RGB values of each pixel of an image indicated as cloud shadow however. This will then give the following:

$$\text{RGB affected} + \text{RGB Clear}_{\text{B}} = \text{RGB clear}_{\text{A}} + (\text{RGB clear}_{\text{B}} + \Delta\text{shadow median} + \Delta\text{illumination})$$

This can then be rewritten:

$$\text{RGB Clear}_{\text{B}} = -\text{RGB affected} + \text{RGB clear}_{\text{A}} + (\text{RGB clear}_{\text{B}} + \Delta\text{shadow median} + \Delta\text{illumination}) \quad (3.8)$$

3.8 CNN based method for cloud shadow mitigation

The main idea of the CNN based cloud shadow mitigation model is to use a modified version of the Cloud-Net model. The main change that needs to be made here is to switch from a binary classification model to regression. With the classification models it was only desired to know whether a certain pixel was a cloud or a cloud shadow. Now we do not only want to know the label, but also how to correct the altered RGB values. Therefore the labels have been changed from binary numbers to a range of values, which indicate the difference between the reflectance value on a cloud shadow compared to a clear day. When there is no cloud shadow present, the label value will be zero. After the reflectance differences between the cloud shadow and clear ground truth value have been taken, the numbers are normalized between 0 and 1 by using formula 3.1. Some parameters were changed like the loss function that was changed to 'mean squared error' (MSE) and learning rate that was changed to 0.00001, which was found to give better performances.

For the training of the data, 5 different scenes with cloud shadows from the Negev desert have been selected. For each date a ground truth was again created by taking the average RGB reflectance values from corresponding clear non-cloudy days. One change that has been made is to divide all the images into smaller tiles of only 128 by 128 pixels. Then from visual inspection each image was analyzed in order to decide whether at least around 20% cloud shadow was present. This gave a total of 44 images that will be used for the training. In total there are three different models to be trained: One for the blue, one for the green and one for the red reflectance correction.

3.9 Cloud shadow mitigation analysis of results

After the correction has been applied, the final step is to analyze the results. The quality of the correction will be determined, by using the SSIM comparison. The formula is stated in a previous section in equation 3.6. The outcome gives each pixel a value between -1 and 1, where 1 indicates a perfect match. The average SSIM value per color channel for all 13 images will be computed, for the pixels where a RGB correction is applied on. The average SSIM values before and after the correction, will then be compared to each other.

A second analysis on the cloud shadow corrected areas, will also be performed. This involves the use of histograms, where three different features per histogram will be plotted: The reflectance values of the cloud shadows, the reflectance values of the clear ground truth scene (on the same locations where a cloud shadow in the other image is present), and thirdly the reflectance values of the corrected cloud shadows. The reflectance overlap with the ground truth, before and after correction, will be computed (255 bins, since normal color intensities range from 0 to 255). An important note to make here, is that this analysis is done in order to show the effect of the correction, but one should be careful with conclusions about the performance. If a correction is carried out correctly, it is expected that the new reflectance values, show a overlap close to 100% with the ground truth reflectances. When a correction is not carried out correctly however, there can still be a relatively high overlap. A small offset will be visible, meaning that the wrong pixels overlap. Therefore, if one wants to make conclusions regarding the performance of the correction, it is wise to also look at the histograms from visual inspection, and only consider a correction to be good when the overlap is very high. In this study the SSIM analysis will be used as the main performance metric.

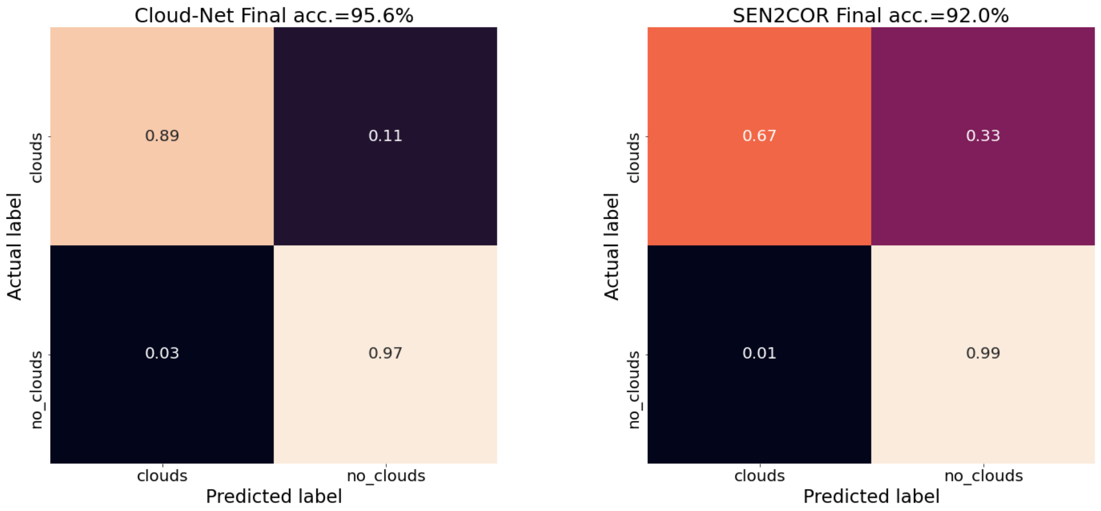
4 Results

4.1 Cloud classification

For the testing phase, regarding the cloud detection, a total of 20 images was used. The 6 locations that were chosen are: Rotterdam Port (Netherlands), Geonoa Port (Italy), Negev desert (Israel) Qinhuangdao Port (China), Aoshan Bay (China) and Lanshan Rizhao (China). An overview of all the scenes is given in the data section, table 3.1. The predictions were compared to the same manual labelled image, which is considered to be the ground truth. Table 4.2 summarises the results, figure 4.1 shows the corresponding confusion matrices.

Table 4.1: Test results of the Cloud-Net model and the SEN2COR algorithm, for the classification of clouds.

Classification model	**Overall accuracy(%)**	**Precision (%)**	**Recall (%)**	**F1 score (%)**
Cloud-Net	95.6	89.1	90.0	89.5
SEN2COR	92.0	67.3	93.1	78.1

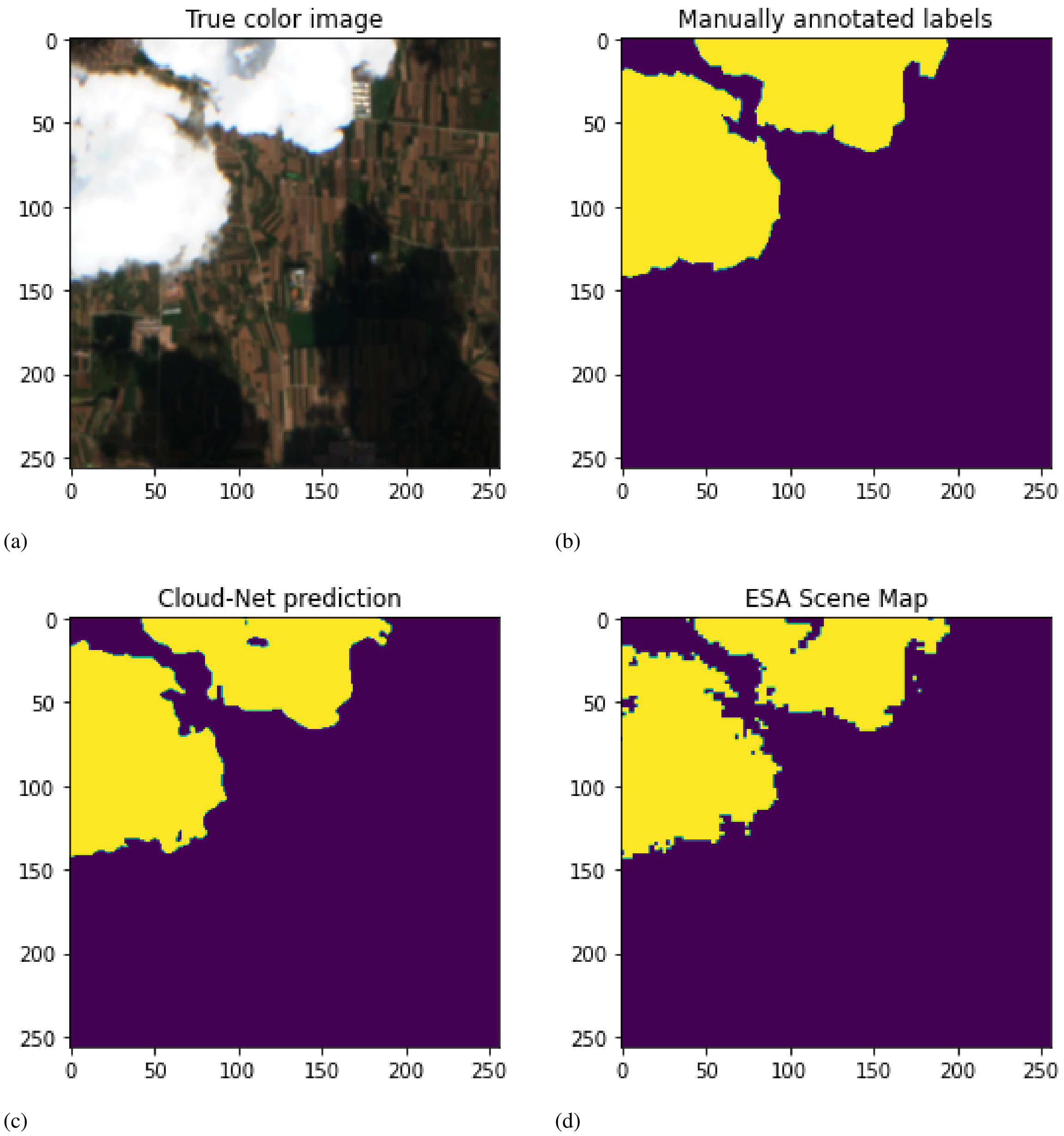


(a)

(b)

Figure 4.1: Confusion matrices for the Cloud-Net model on the left and the SEN2COR results on the right.

It can be concluded that Cloud-Net outperforms the ESA Scene Map, or so-called SEN2COR algorithm. Consider table 4.1, where all the scores except the recall are higher for Cloud-Net. The overall accuracy is 3.6% higher (95.6% versus 92.0% respectively), but especially the precision score is much higher (89.1% versus 67.3%). An analysis of one of the individual scenes is presented in figure 4.2.



(a)

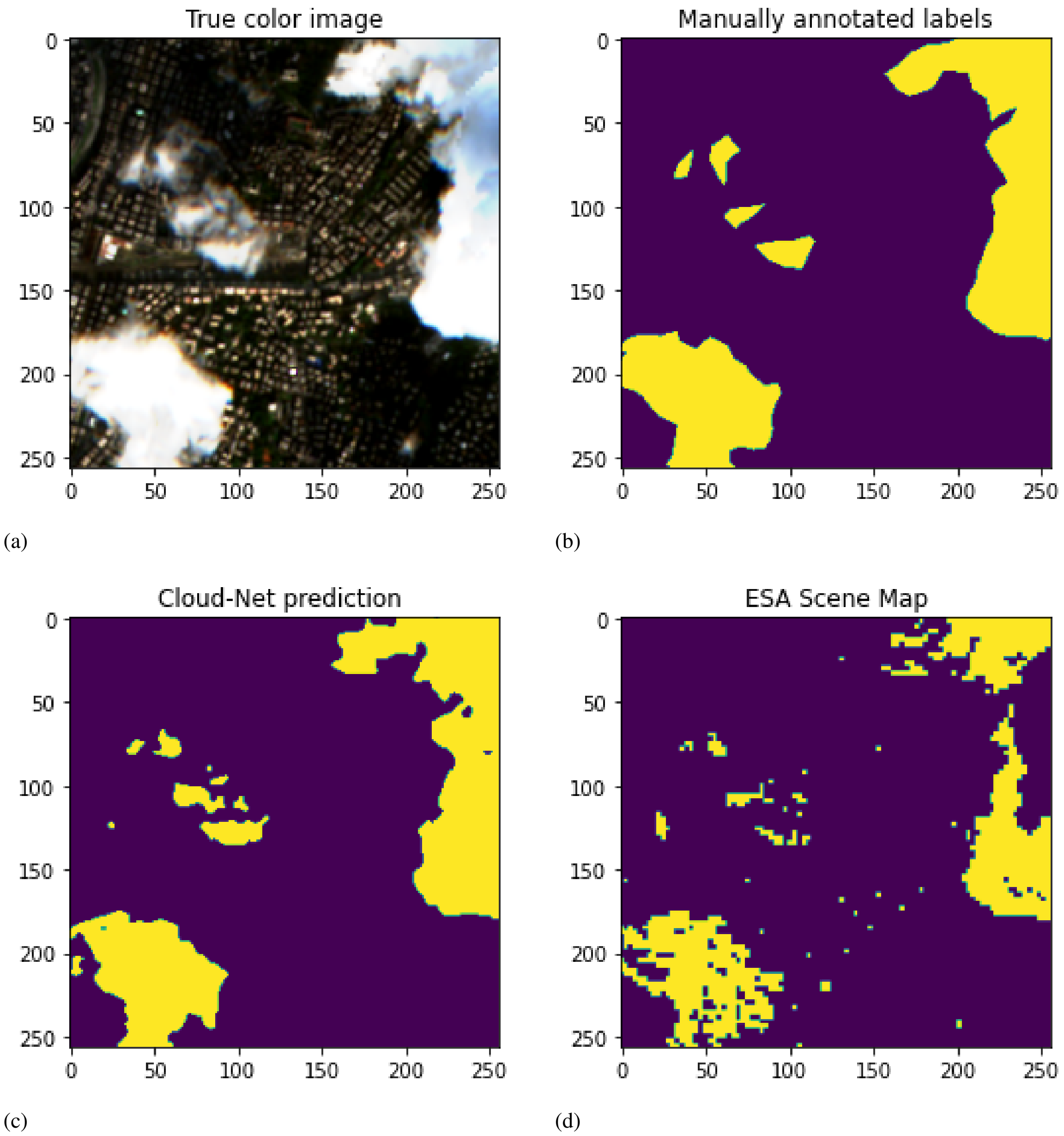
(b)

(c)

(d)

Figure 4.2: Thick clouds classification results, example of one of the testing images. Left top: True color image. Right top: Manually annotated labels (ground truth). Left bottom: Cloud-Net prediction. Right bottom: ESA Scene Map (SEN2COR).

The cloud detection results of figure 4.2 were taken at Genoa. The overall CNN accuracy of 98.2% is slightly better than the output of the SEN2COR algorithm of 97.6%, but the difference is quite small. There are some spots that SEN2COR seemed to have missed, but in general it can be stated that both results are of equal quality. Let's also consider a scene where a clear difference in performance is present.



(a)

(b)

(c)

(d)

Figure 4.3: Thick clouds classification results example of one of the testing images. Left top: True color image. Right top: Manually annotated labels (ground truth). Left bottom: Cloud-Net prediction. Right bottom: ESA Scene Map (SEN2COR).

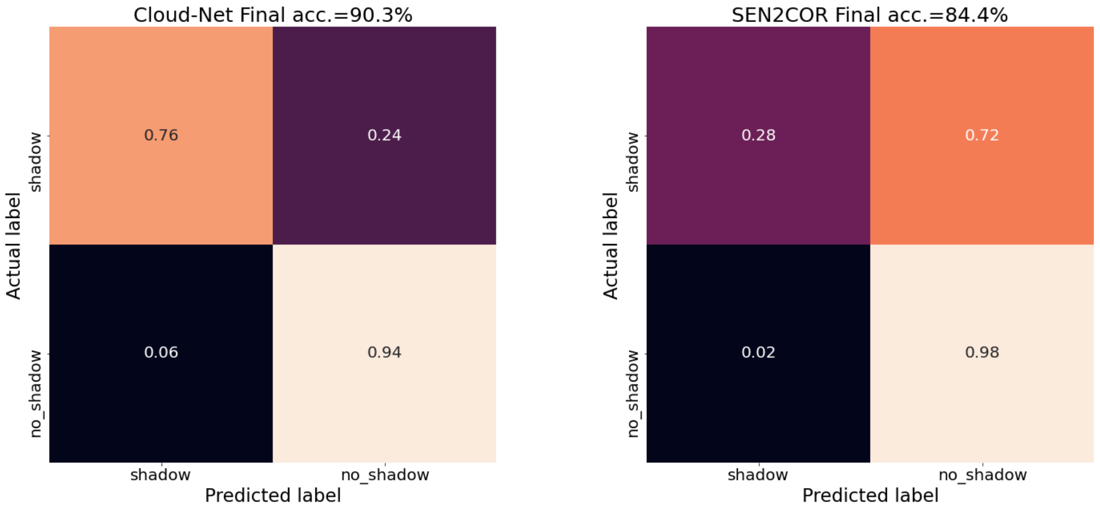
In figure 4.3, the cloud classification results of another scene are presented. It can be seen that the SEN2COR algorithm now significantly performs worse. The CNN has a higher overall accuracy of 96.6% versus 90.7% respectively. Especially the precision of 91.1% versus 65% is much higher for the Cloud-Net model. The results for all individual scenes are given in appendix A. For some scenes, there is a significant difference in performance, for others the differences are relatively small. Before diving into a possible explanation of this, the results of the cloud shadow detection will be discussed in the next section.

4.2 Cloud shadow classification

For the testing phase, regarding the cloud shadow detection, a total of 22 images was used. Table 4.2 summarises the results, including the corresponding confusion matrices in figure 4.4.

Table 4.2: Test results of the Cloud-Net model and the SEN2COR algorithm, for the classification of cloud shadows.

Classification model	**Overall accuracy(%)**	**Precision (%)**	**Recall (%)**	**F1 score (%)**
Cloud-Net	90.4	76.2	75.7	76.0
SEN2COR	84.4	28.2	81.6	41.9

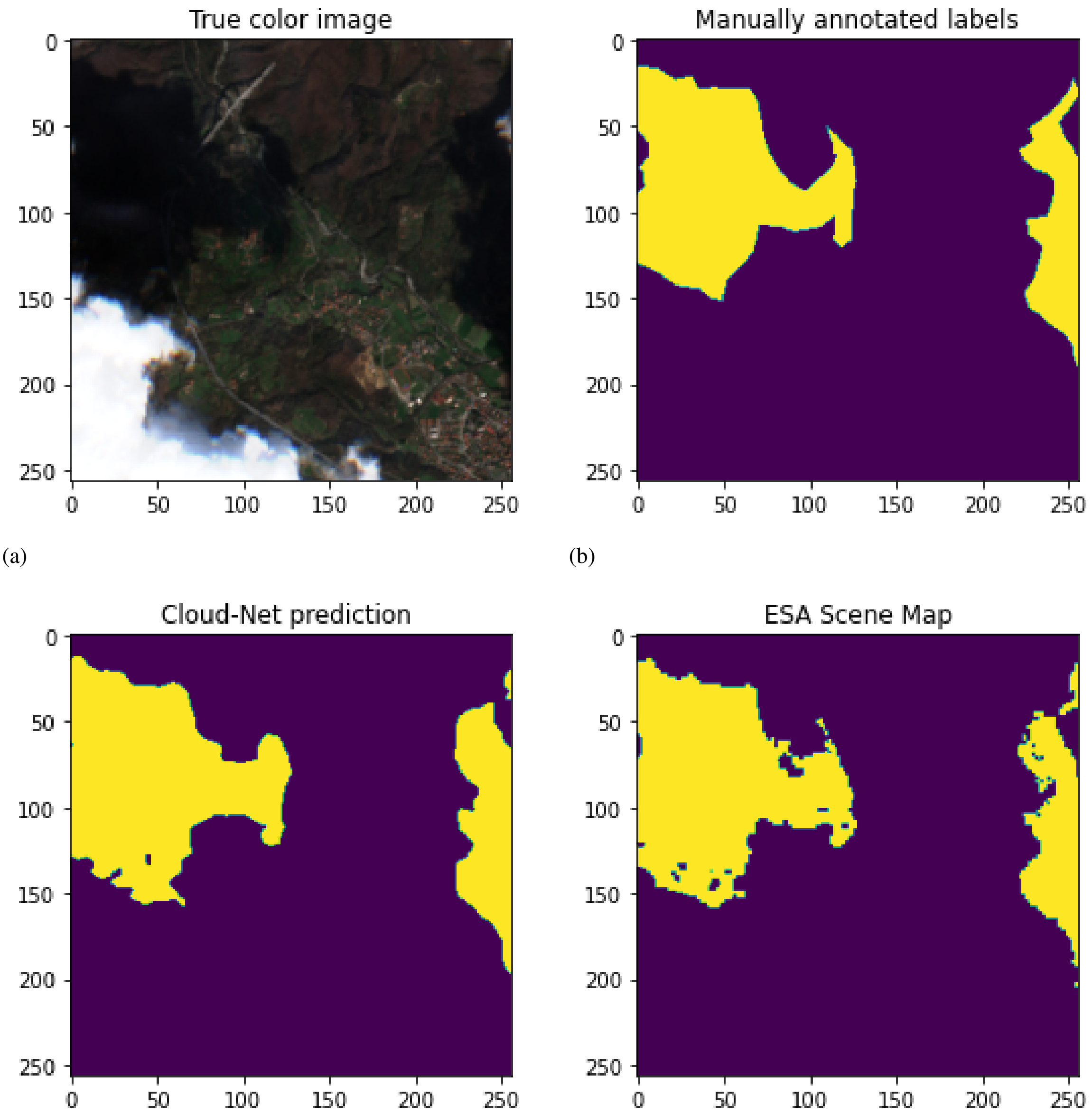


(a)

(b)

Figure 4.4: Confusion matrices for the Cloud-Net model on the left and the SEN2COR results on the right.

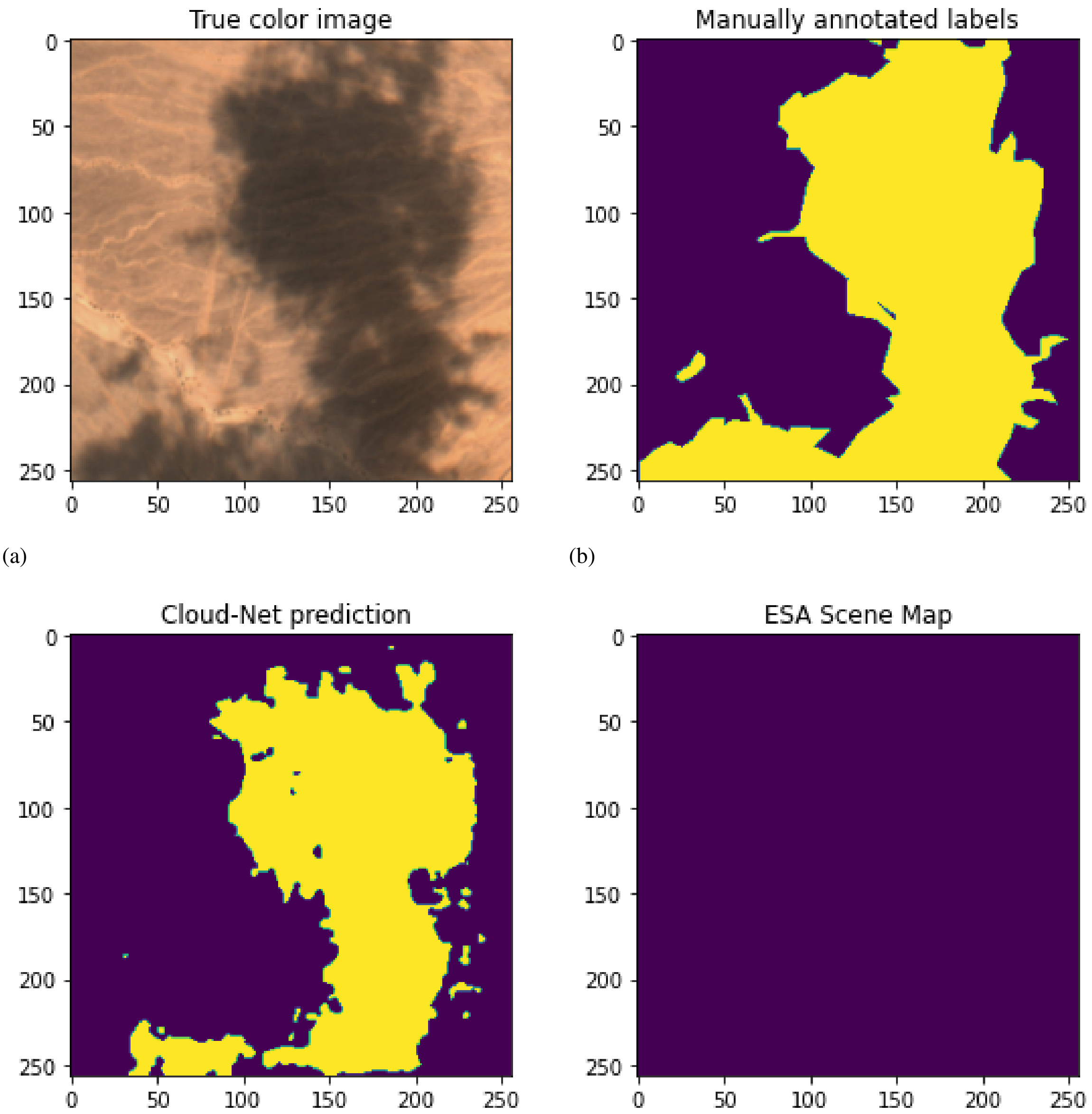
Comparing the different statistical analyses, it can be concluded that also in this case the Cloud-Net model outperforms the SEN2COR algorithm. The overall CNN accuracy of 90.4% is significantly higher than the 84.4% of the SEN2COR algorithm. What is most striking however, is the huge difference in precision performance. The Cloud-Net model has a 76.2 % precision score, versus only 28.2% for SEN2COR. This indicates, that whenever a cloud shadow pixel is present, the SEN2COR algorithm has difficulties with detecting it. The only score where SEN2COR is better than Cloud-Net, is the recall (81.6 % versus 75.7% respectively). An analysis of some individual scenes is presented in the next section.



(a) (b) (c) (d)

Figure 4.5: Cloud shadow classification results example of one of the testing images. Left top: True color image. Right top: Manually annotated labels (ground truth). Left bottom: Cloud-Net prediction. Right bottom: ESA Scene Map (SEN2COR).

Figure 4.5 illustrates the cloud shadow classification results for one of the testing images. The Cloud-Net model achieved here an overall accuracy of 96.5%, and the SEN2COR an accuracy of 96.2%. In this scene, there are a few spots that the SEN2COR algorithm missed, but in general the performance is similar to Cloud-Net. Let's also consider another scene, where there is a big difference visible.



(a)

(b)

(c)

(d)

Figure 4.6: Cloud shadow classification results example of one of the testing images. Left top: True color image. Right top: Manually annotated labels (ground truth). Left bottom: Cloud-Net prediction. Right bottom: ESA Scene Map (SEN2COR)

In Figure 4.7, it can be seen that the SEN2COR algorithm was not able to capture any of the cloud shadows. The Cloud-Net prediction is also not perfect, but is able to capture at least most of the cloud shadow pixels. The individual scores per scene are summarised in appendix B. From this it becomes clear, that the SEN2COR algorithm has the same problem with the other desert scenes.

What is the reason that the Cloud-Net model outperforms the ESA Scene Map? The setup of the two models is totally different, as described in chapter 2. One of the most fundamental differences between the Cloud-Net model and the SEN2COR algorithm, is that Cloud-Net takes information from adjacent pixels into account when determining if a pixel is a cloud shadow or not. This is not the case for the SEN2COR algorithm, where each pixel is analysed individually. It is therefore interesting to investigate what happens when the spatial context is not taken into account.

This can be done by modifying the Cloud-Net model, and change the kernel size to 1. This means that each pixel is analysed individually and the spatial context is ignored. The model structure and setup are remained intact. An overview of the different performances is given in table 4.3. The new CNN predictions, for the same two scenes shown before, can be found in figure 4.7.

Table 4.3: Test results of the modified Cloud-Net model and the SEN2COR algorithm, for the classification of cloud shadows.

Classification model	**Overall accuracy(%)**	**Precision (%)**	**Recall (%)**	**F1 score (%)**
Cloud-Net, modified	81.9	40.5	56.5	47.2
SEN2COR	84.4	28.2	81.6	41.9

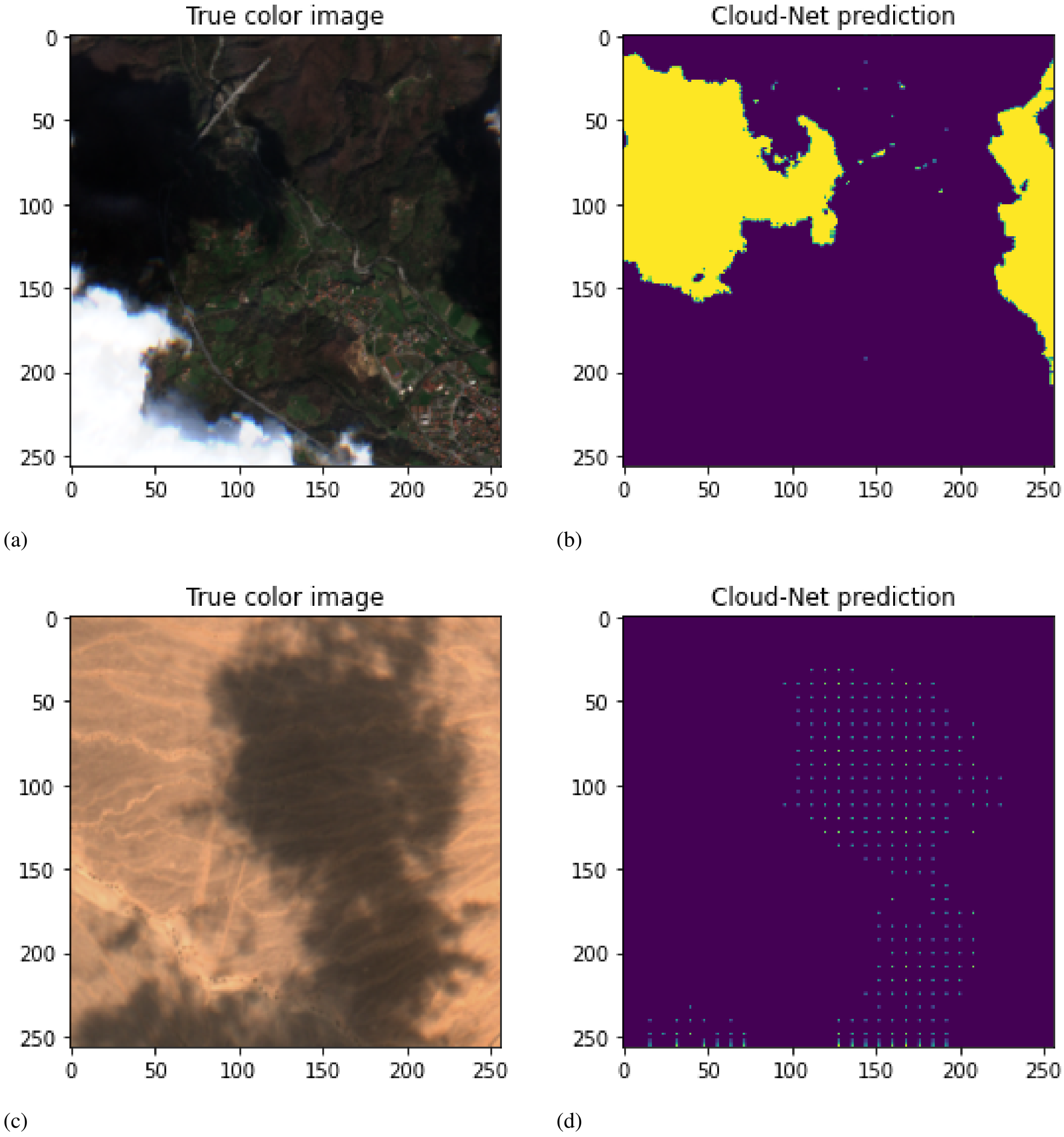


Figure 4.7: Cloud shadow classification results for the modified Cloud-Net model. Left top: True color image scene 1. Right top: CNN prediction scene 1. Left bottom: True color image scene 2. Right bottom: CNN prediction scene 2

What is interesting to see, is that the modified Cloud-Net model model is still able to make good predictions for scene 1, but totally fails on scene 2. The same behavior was found for the SEN2COR algorithm, and this also holds for the other desert scenes. From this it it can be concluded, that the performance for cloud shadow classification in desert scenes is increased, when the information of the surrounding pixels is taken into account. The results for the individual scenes are given in appendix C.

4.3 Cloud shadow mitigation

Decomposition of components method

The experiment was carried out on 13 images, all of 512 by 512 pixels, as described in the previous chapter. A summary of the results for the average SSIM scores, and the overlap of the histograms with the ground truth, are presented in table 4.4 and table 4.5 respectively.

Table 4.4: Overall SSIM scores (decomposition of components method).

SSIM score	**before correction**	**after correction**
Blue reflectances	0.17	0.59
Green reflectances	0.18	0.68
Red reflectances	0.18	0.67

Table 4.5: Overall histogram overlap scores (decomposition of components method, 255 bins).

Histogram ground truth overlap	**before correction (%)**	**after correction (%)**
Blue reflectances	6.7	78.2
Green reflectances	8.3	87.9
Red reflectances	10.6	88.4

From the tables above, it can be stated that the RGB modification has resulted in a significant improvement. The overall blue SSIM score (unitless range between -1 and 1) has improved from 0.17 to 0.59, and the green/red reflectances have improved from 0.18 to 0.68/0.67. The overlap between the histograms of the corrected and ground truth image has also increased. On average increasing from around 8% overlap to almost 85%. Consider the results of one of the individual scenes, presented in figure 4.9.

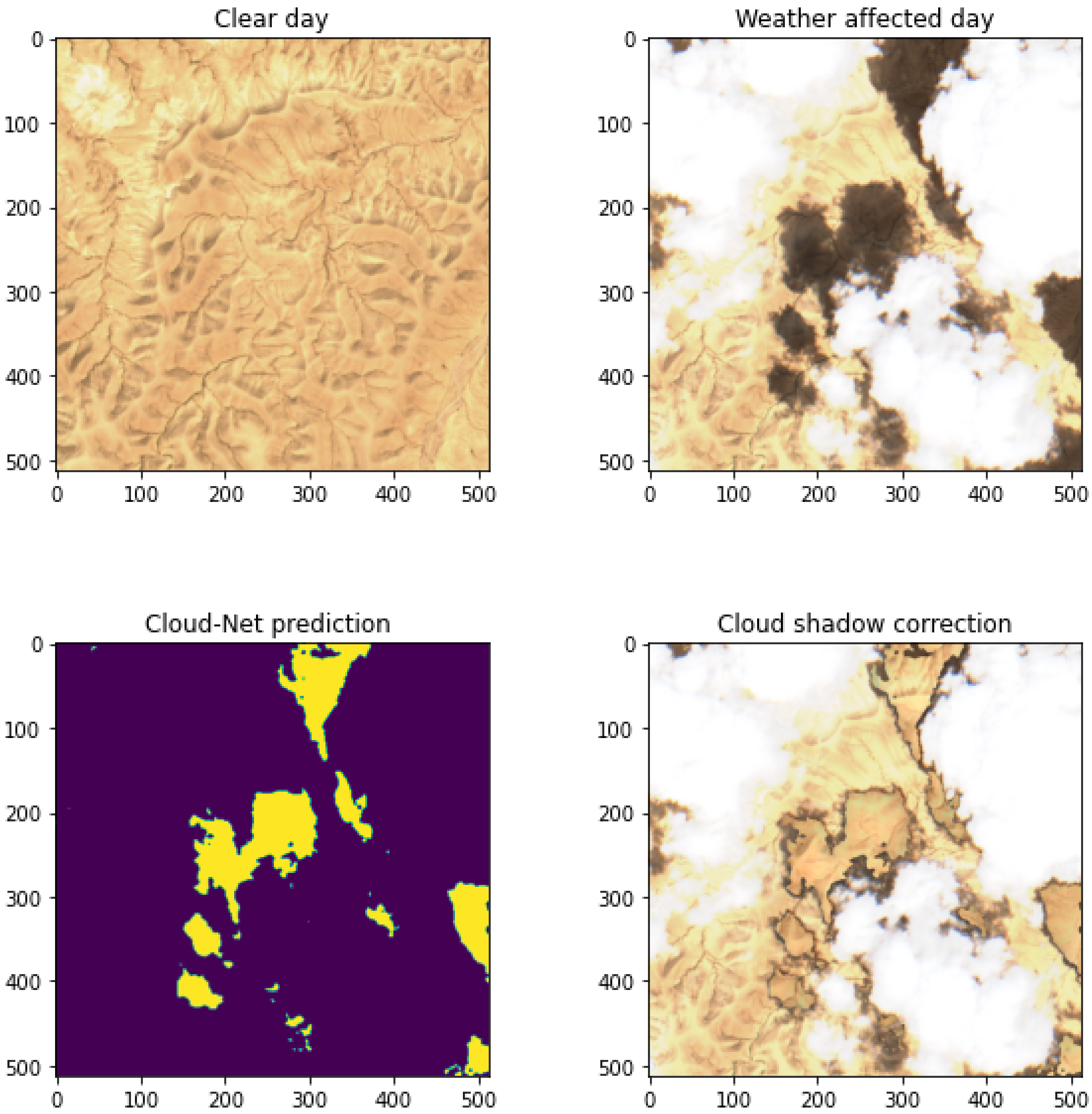
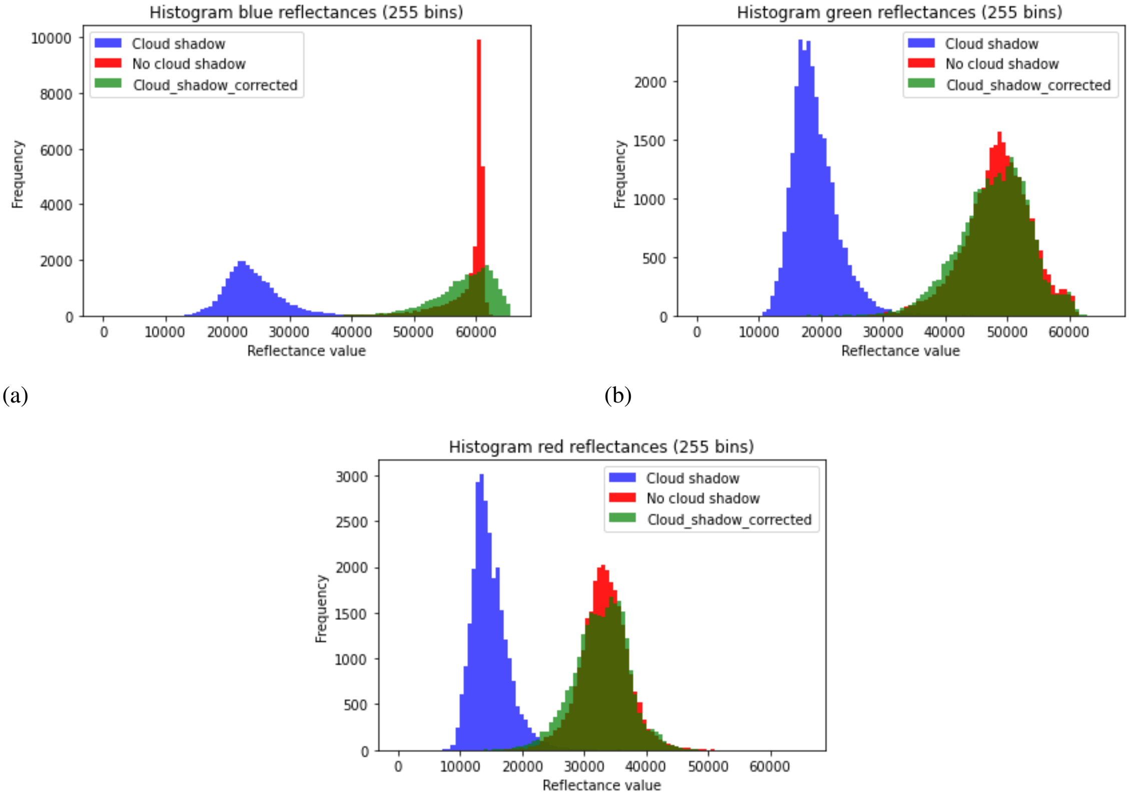


Figure 4.9: Clouds shadow mitigation results. Left top: Clear day ground truth. Right top: Weather affected image. Left bottom: Cloud-Net prediction. Right bottom: Cloud shadow mitigation results.

Figure 4.9 displays the cloud shadow mitigation results on one of the scenes. The first step is to verify that the CNN prediction on the shadow locations are accurate. From visual inspection it can be concluded that this is the case, except for some shadows on the edges. Then the formula on the cloud shadow correction is applied, giving the right bottom image as final result. The areas that used to be dark spots are now much lighter and are now more aligned with the ground truth image. Also consider the histograms corresponding to this scene.



(a)

(b)

(c)

Figure 4.10: Blue, green and red reflectance histograms, showing the difference between the cloud shadow affected area and the clear ground truth, as well as the effect of performing the cloud shadow correction.

Figure 4.10 shows the RGB histograms, where the effect of the RGB modification becomes visible. The data that was downloaded from Sentinel Hub EO Browser was in 16 bit format, meaning that the RGB reflectances range from 0 to 65535. The cloud shadow areas are represented in blue and the clear ground truth values in red. There is a clear difference visible in each histogram. What is desired to see is that after the cloud shadow correction the histograms become more aligned. This correction is captured in the green colored reflectance values. What can be seen is that at the right top and bottom histograms (green and red reflectances respectively), corrected and clear day reflectances do now very well overlap each other. At the left top histogram however, (blue reflectances) there is a clear peak in the clear day reflectances, that is not very well captured in the correction. In general it can be said that the correction is quite good, with an average increase in overlap from 1.5% to 78.2%. The results of the other scene will not be described here, but can all be found in appendix D.

CNN based method

The testing of the CNN based method has been done on the same 13 images. The results are summarized in table 4.6 and table 4.7.

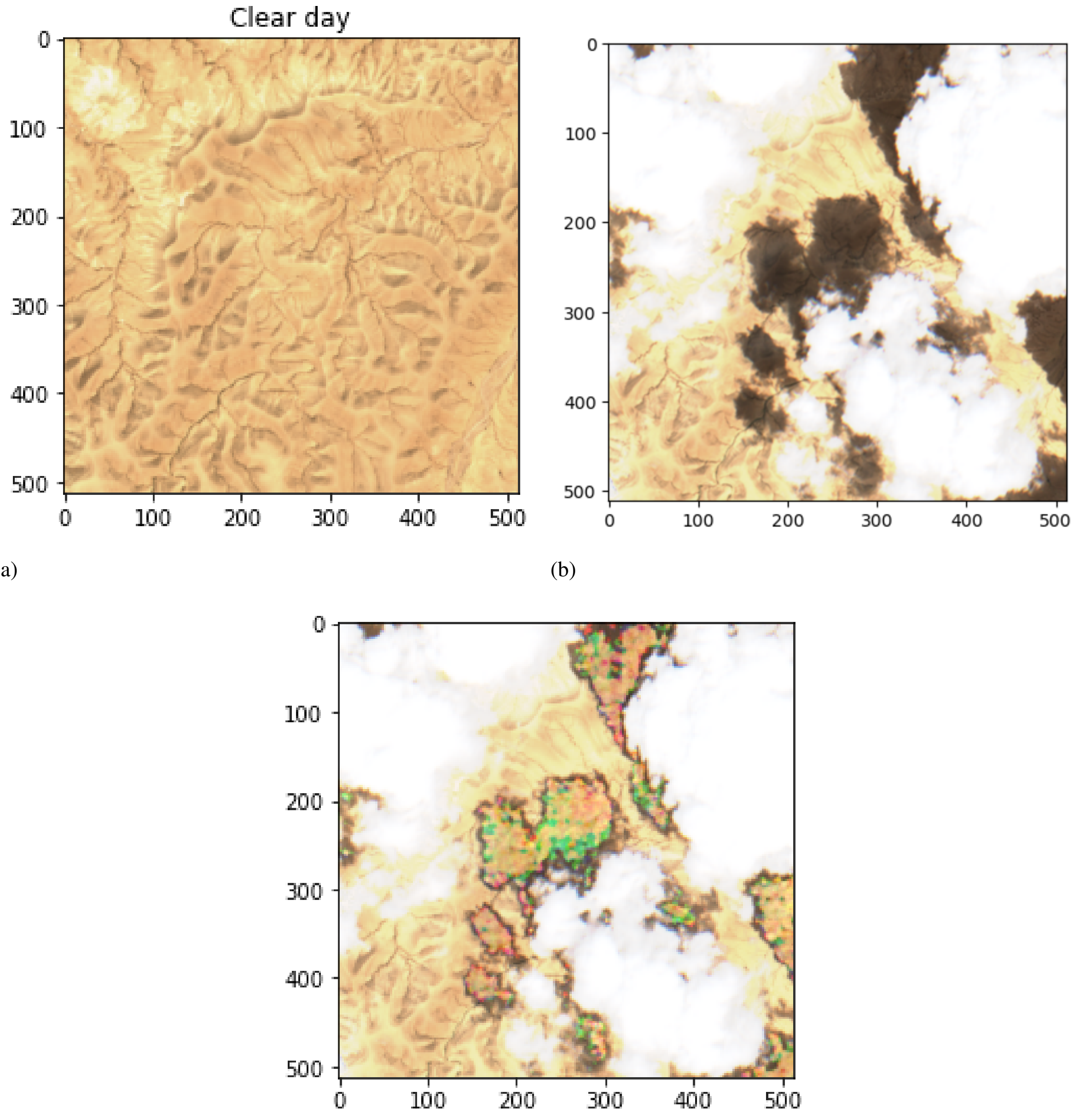
Table 4.6: Overall SSIM scores (CNN based method).

SSIM score	**before correction**	**after correction**
Blue reflectances	0.17	0.20
Green reflectances	0.18	0.20
Red reflectances	0.18	0.24

Table 4.7: Overall histogram overlap scores (CNN based method, 255 bins).

Histogram ground truth overlap	**before correction (%)**	**after correction (%)**
Blue reflectances	6.7	55.3
Green reflectances	8.3	72.3
Red reflectances	10.6	71.6

Comparing the two tables, there is a clear difference visible. Regarding the SSIM scores, there is only a small improvement visible, with on average only a 0.04 improvement after correction. This indicates that the RGB modification only had a small effect on the mitigation of the shadow effect. The histogram overlap however, shows a significant increase, on average around to around 66%. In the previous chapter it was stated that a very high overlap is needed, in order to conclude that a correction has been carried out correctly. The results of one of the individual scenes is presented in figure 4.11.



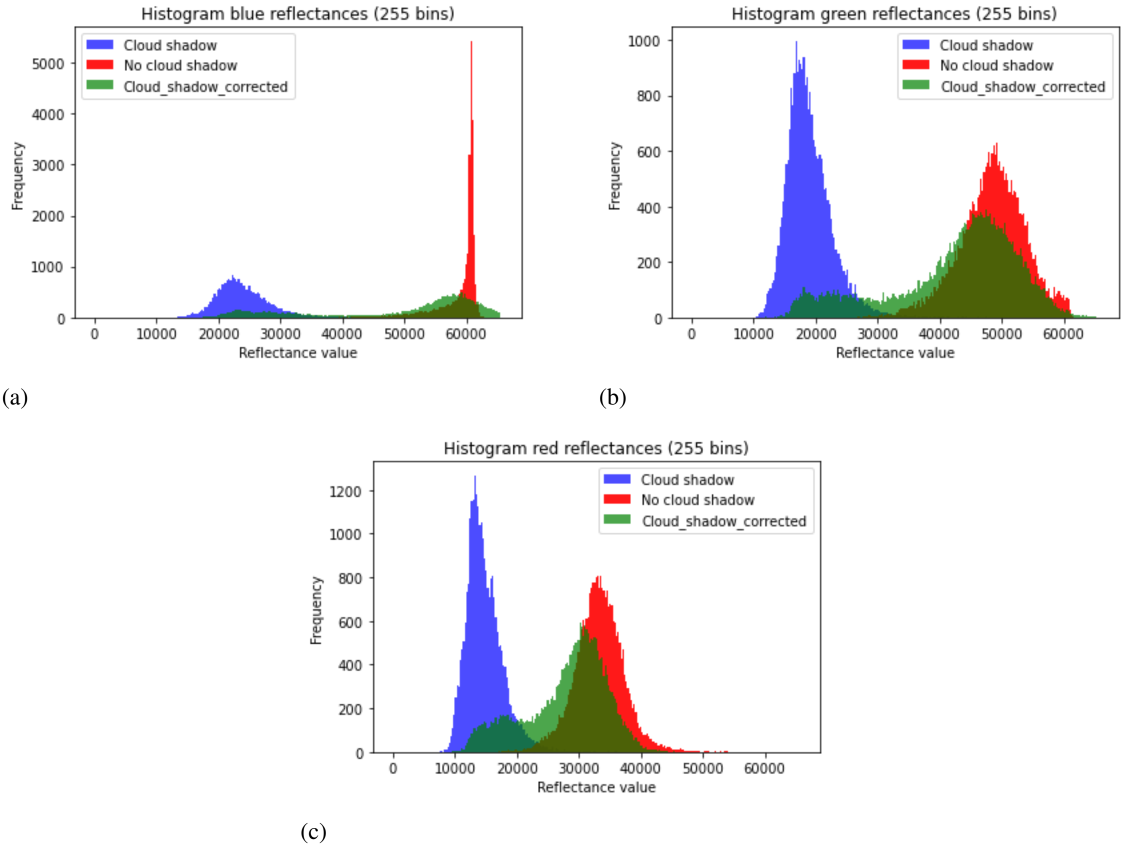
(a)

(b)

(c)

Figure 4.11: Clouds shadow mitigation results. Left top: Ground truth image Right top : Weather affected image. Bottom: Cloud shadow mitigation results

From visual inspection it can be seen that the location of the cloud shadows is determined quite accurately but that the quality of the correction is lower compared to the decomposition of components method. Here the SSIM score is only 0.1 while the decomposition of components method gave a score of 0.44. The color of the corrected areas do not very well match the ground truth image and there are also some light blue spots visible that should not be there. We also consider the corresponding histograms.



(a)

(b)

(c)

Figure 4.12: Blue, green and red reflectance histograms showing the difference between the cloud shadow affected area and the clear ground truth, as well as the effect of performing the cloud shadow correction.

Here the problem of the histogram overlap quality indicator becomes clear. It can be seen that the RGB values have been shifted towards the ground truth values, but the shift has not been carried out correctly. There is a clear increase in overlap visible (on average from 1.5% to 59%), but the wrong pixels are overlapping. This explains why the SSIM score has improved only a little bit while the histogram overlap has increased a lot.

The results for the other Negev desert scene can be found in appendix E. Overall it can be concluded that the decomposition of components method gives better results. The average SSIM value after correction of 0.65 clearly outperforms the 0.21 score of the CNN based method. The advantage of the CNN based method would be, that once the model is trained, there is no ground truth image needed to perform a correction. In order to make that work however, the performance of the model should be increased a lot.

5 Discussion

The results indicate that the adapted version of Cloud-Net is able to locally outperform the SEN2COR algorithm, on the classification of both clouds and cloud shadows. On the classification of clouds, the overall accuracy is 3.6% higher (95.6% vs. 92.0%), and for the classification of cloud shadows, the overall accuracy is 6% higher (90.4% vs.84.4%). Regarding the cloud shadow mitigation in the Negev desert area, a decomposition of components and CNN based method were proposed. It was found that the decomposition of components method gave better results. The SSIM score here resulted in an average improvement from 0.18 to 0.65 for the RGB color channels. The CNN based method only gave a very small improvement, from 0.18 to 0.21.

5.1 Performance difference Cloud-Net and SEN2COR

One of the most fundamental differences between the Cloud-Net model and the SEN2COR algorithm is that Cloud-Net takes the spatial context into account, for the classification of clouds and cloud shadows. It was demonstrated that without using the information of adjacent pixels, the overall accuracy for cloud shadow detection, dropped from 90.4% to 81.9%, a significant difference. This finding agrees with the original motivation behind the construction of CNN's, that was proposed by LeCun et al. (1999), one of the first authors on neural networks: "Images have a strong 2D local structure: variables (pixels) that are spatially nearby are highly correlated. Local correlations are the reasons for the well known advantages of extracting and combining local features before recognizing spatial or temporal objects, because configurations of neighboring variables can be classified into a small number of relevant categories (e.g. edges, corners...).

Another important difference is that a CNN learns to distinguish features directly from the data, while with rule-based algorithms we need to define ourselves what information is important and what not. With a lot of trial and error we are usually able to construct a model that is a reasonable approximation of the truth, but there is still a lot of information that we miss. A good example of this is demonstrated by Tshitoyan et al. (2019). The authors developed a deep learning algorithm that analyzed text from old material science papers. Without any explicit insertion of chemical knowledge, the algorithm captured complex materials science concepts such as the underlying structure of the periodic table and structure–property relationships in materials. Furthermore it was demonstrated that the model can recommend materials for functional applications several years before their discovery. This suggests that latent knowledge regarding future discoveries is to a large extent embedded in past publications. From this it can also be concluded that deep learning algorithms are very good in filtering relevant information, and how to interpret it.

5.2 Cloud and cloud shadow detection in other areas

This study focused on local performance of the Cloud-Net and SEN2COR algorithm, but what are the limitations if we consider other scenes?

The SEN2COR algorithm is already calibrated to perform cloud and cloud shadow detection on a wide variety of scenes on a global level. This makes it very easy to use from a practical point of view. As it was demonstrated in the results however, the SEN2COR algorithm sometimes completely misses the complete presence of certain features. Compared to other rule-based methods like FMASK and MAJA, SEN2COR is the most consistent one, as was demonstrated in table 1.1. Still this study showed that the performance can vary a lot. When you encounter such an issue for a particular scene, there is nothing to do about it. The parameters of the SEN2COR algorithm are fixed. In other words: The SEN2COR algorithm is easy to use and widely applicable, but when any issues are encountered for a particular scene, there is no flexibility to adapt the model.

In order to use the Cloud-Net model for a particular scene, the CNN needs to be trained first with the proper training data. The final model in this study is adapted to a variety of scenes, summarised in table 3.1. This model can be used but chances are that the result is not very good. If one wants to take maximal advantage of the CNN, the model must be trained first with the right data. This procedure of course also includes the manual labelling, preprocessing etc. This can be time consuming but will result in a model that is highly likely to outperform the SEN2COR algorithm on the scenes of interest. The pro's and con's of the CNN are opposite to that of the SEN2COR algorithm: Cloud-Net is not easy to use and the process is time consuming, but there is high flexibility on the setup and the results are much likely to outperform the SEN2COR algorithm.

5.3 Evaluation of cloud shadow mitigation quality indicator

One of the issues that was encountered in this study, was how to evaluate the performance of a cloud shadow mitigation model. One can always use visual inspection, but a more quantitative analysis is preferred. In Nagare et al. (2018) one of the quality indicators was a correlation coefficient with a reference image. There was no elaboration on the exact formula however. In Richter and Muller (2005), a shadow correction method was proposed, but the quality was only examined from visual inspection. There is also not a lot of literature available in general on this topic. Another important point is that there are two possible options for the desired output: Only correct for the effect of shadows and make it match the rest or the image, or correct for the effect of shadows and illumination effects and make it match a reference image. The second option was chosen in this study.

The most important quality indicator in this study is the SSIM method. As far as we know this method has not been used before for the evaluation of cloud shadow correction models. This model for image comparison was originally proposed by Wang et al. (2004), to make an quantitative analysis on digital image distortion. Here it was argued that there are already some convenient and easy to use quality metrics like mean squared error (MSE) or peak signal-to-noise ratio (PSNR), but that they are not very well matched to perceived visual quality. The SSIM method overcomes that issue, and that is why we believe that this method has strong potential for the evaluation of cloud shadow corrections. Most papers on this subject still rely heavily on visual inspection, and the SSIM method can be very good in capturing the perceived quality of a correction. This is also in line with the results presented in this study, where a properly corrected cloud shadow area gave higher SSIM values and also looked better from visual inspection. In the work of Wang and Li (2010) , a distortion of an image was evaluated with both the absolute error and the SSIM output. It was concluded that the SSIM output better reflects the spatial variations of perceived image quality.

To evaluate cloud shadow corrections based on the histogram overlap is to our knowledge something completely new. It also turned out not to be the most convenient metric. As it was demonstrated, only a very high histogram overlap indicates a good correction. However, the amount of overlap needed to consider a correction good, has not been quantified. From visual inspection it is interesting to see how the color intensities have shifted, but it is hard to define a good quality metric from the resulting overlap.

5.4 Possible improvements on CNN based method for cloud shadow correction

It was demonstrated that the CNN based method gave little improvement on the correction of cloud shadows. The SSIM values increased on average only from 0.18 to 0.21. There are multiple possibilities why this corrections were so poorly. A few options will be discussed.

The first reason is that machine learning was never a good idea to tackle the problem. Nowadays there is a lot of data available and different machine learning techniques can be used to make sense of this data. That doesn't mean however that these models are always working. When there is simply no correlation between the input data and the labels or the correlation is too small there is nothing that a CNN or other deep learning model can do. In the case of this project the CNN should learn how to correct cloud shadow affected pixels, based on the reflectance value. It is not unlikely that this is something the CNN can simply not learn. From an intuitive point of view it makes sense that it is quite advanced not only to identify a cloud shadow affected pixel, but also to learn how much to alter the RGB reflectance value in order to remove this shadow effect.

A second option is that the architecture of the CNN is not fitting the problem. The model was originally designed to perform well on classification problems and it has now been changed to a regression model. Perhaps the results would be better if the number of layers is changed, other activation functions are used, different filter sizes are tried etc. This is something one can spend a lot of time on to find out the best set-up of a particular model and there was not enough time in this project to do that. It would be interesting to investigate different CNN architectures to find out if there are better options.
A third and final option is the amount of training data that was used. More data to train the model will usually give you better results on the testing data. The amount of data needed in order to get a well performing model is highly varying from one project to another. As was already discussed before, the correction of cloud shadows is a more complicated task than just the detection. Therefore, it is a plausible theory that a larger dataset is needed in order to get the model working well.

6 Conclusions and recommendations

This study can be divided into two major parts. The first part aims to use a modified CNN in order to improve on the classification of clouds and cloud shadows. The second part explores the mitigation of the effect of cloud shadows, where two methods have been proposed. The outcome of each research question will be summarised.

1) Is it possible for a Convolutional Neural Network, to locally outperform the SEN2COR algorithm on the classification of clouds?

For the training of the CNN on the classification of clouds, 85 images (each image 256x256 pixels) have been used, with a B02-B03-B04-B08 band combination. 20 images were selected for testing and the results were compared to that of the SEN2COR algorithm. It was found that the Cloud-Net model outperforms the SEN2COR algorithm. The overall accuracy was higher (95.6% vs. 92.0%), surpassing the goal of outperforming the SEN2COR algorithm by 1%. The precision was also higher (89.1% vs. 67.3%) and also the F1 score was better (89.5% vs. 78.1%). Only the recall score of the SEN2COR algorithm outperformed the Cloud-Net model (93.1% vs. 90.0%).

2) Is it possible for a Convolutional Neural Network, to locally outperform the SEN2COR algorithm on the classification of cloud shadows?

91 images were used for the training of cloud shadows, with a B02-B03-B04-B08-B11-B12 band combination. 22 images were selected for testing. It was again found that the Cloud-Net model outperformed the SEN2COR algorithm. This time the differences were even bigger. The overall accuracy of the CNN was much higher (90.4% vs. 84.4%). This means that also in this case the goal of outperforming the SEN2COR algorithm by at least 1%, is achieved. Better results were also found in the precision score (76.2% vs. 28.2%) and F1 score (76.0% vs. 41.9%). Again, only the recall score of the SEN2COR algorithm was higher than Cloud-Net (81.6% vs. 75.7%). An investigation was also done on why the CNN performs better. It was shown that when the kernel size was reduced to 1, the results significantly became worse. The overall accuracy dropped to 81.9%.

3) To what degree is it possible to correct for the effect of cloud shadows, and which method has the most potential?

Two proposed cloud shadow mitigation methods have been tested on 13 different images of 512 by 512 pixels. The names given to the correction models were 'decomposition of components' and 'CNN based method'. The structural similarity index measure (SSIM) was used as the main quality indicator. It was demonstrated that with the decomposition of components method the blue SSIM values increase from 0.17 to 0.59. The green and red SSIM values both increase from 0.18 to 0.68. From visual inspection this significant improvement was also visible, with colors more aligned with the reference image.
The CNN based method significantly performed worse. The blue and green reflectances only increased from 0.18 to 0.20, and the red reflectances from 0.18 to 0.24. From visual inspection it also became clear that the colors did not match the ground truth image. Therefore, it can be concluded that the decomposition of components method has the most potential for the mitigation of cloud shadows.

4) Recommendations for future work

- Investigate if it is possible for a CNN to outperform the rule-based algorithms on a global scale. This means that a very large amount of images need to be labelled in order to represent as many different scenes as possible. If proven successful, one can skip the training phase and perform a direct classification on the scene of interest.
- Elaborate on the architecture of Cloud-Net. In this study some hyperparameters have been tweaked, but there are still many options in the setup that can be changed. For example, the number of hidden layers can be altered, the dropout value can be changed, kernel size etc. This is a very computationally expensive process, but can result in better performances.
- Continue on the exploration of different performance metrics to evaluate the effect of a cloud shadow mitigation method. In this study the SSIM method was introduced, but it is interesting to see if there are other metrics that are also able to accurately capture the quality of a correction.
- Apply the decomposition of components method on a non-stable target. For most applications the target is not stable and will change over time. A logical step in future work would therefore be to also investigate this shadow mitigation method on non static scenes.
- Investigate if it is possible to improve on the CNN based method for the correction of cloud shadows. As a first step we would recommend to add more training data and see if this gives better results. A second step would be to change the setup of the model. We would recommend to look into the main architecture of the the CNN and make some major changes, like the number of hidden layers and the kernel size.

A

Table A1: Cloud classification results presented for each individual test image.

Index	**CNN acc**	**ESA acc**	**CNN prec**	**ESA prec**	**CNN rec**	**ESA rec**	**CNN F1**	**ESA F1**
1	98.2	97.6	93.5	91.0	98.7	98.6	96.0	94.6
2	99.0	98.8	89.0	85.8	95.8	96.8	92.2	91.0
3	92.6	91.3	89.1	87.2	94.6	93.6	91.8	90.3
4	92.1	89.5	78.3	71.4	93.0	90.0	85.0	79.7
5	97.3	97.1	78.1	66.4	78.3	84.6	78.2	74.4
6	96.6	90.7	91.1	65.0	93.0	88.3	92.0	74.9
7	97.5	96.9	63.9	44.3	84.9	95.8	72.9	60.5
8	98.7	98.4	34.0	20.7	80.7	64.6	47.8	31.3
9	91.2	78.8	72.1	24.4	95.4	99.6	82.1	39.2
10	94.2	90.2	90.9	67.8	89.1	97.0	90.0	79.8
11	97.2	96.7	97.7	51.0	70.3	94.8	81.8	66.3
12	100	100	100	100	100	100	100	100
13	93.9	83.1	70.3	5.9	92.3	94.3	79.8	11.8
14	96.9	96.9	95.9	83.4	85.2	95.2	90.3	88.9
15	93.9	92.2	94.3	63.0	79.7	98.2	86.4	76.8
16	93.9	75.1	94.7	27.4	88.2	99.9	91.4	43.0
17	89.5	80.3	86.6	45.8	83.2	92.8	84.9	61.3
18	95.4	95.4	99.4	99.0	95.2	95.5	97.3	97.2
19	98.0	97.8	90.4	89.2	80.2	78.8	85.0	83.7
20	96.0	93.8	88.1	81.5	76.3	65.6	81.8	72.7

B

Table B1: Cloud shadow classification results presented for each individual test image.

Index	CNN acc	ESA acc	CNN prec	ESA prec	CNN rec	ESA rec	CNN F1	ESA F1
1	89.9	90.2	42.5	18.2	57.8	83.4	48.3	29.9
2	76.0	86.7	91.0	5.2	36.2	68.9	52.0	9.6
3	93.6	91.5	83.6	71.6	88.1	88.7	85.4	79.3
4	92.5	94.2	82.2	38.4	57.3	99.5	67.8	55.4
5	86.5	89.6	89.1	0.7	42.8	71.8	57.9	14.7
6	93.2	96.7	79.9	86.9	60.2	80.6	68.8	83.7
7	96.9	98.6	80.7	84.1	57.2	81.1	66.6	82.6
8	95.7	96.9	63.8	84.5	66.6	72.2	65.4	77.0
9	93.3	90.9	85.4	86.4	83.5	75.0	84.1	80.3
10	96.5	96.2	96.2	93.7	87.6	87.8	91.7	90.7
11	95.8	89.5	91.4	63.4	92.1	94.2	92.0	75.8
12	81.8	90.2	73.7	94.3	57.2	71.5	63.7	81.4
13	90.2	68.4	78.7	0	88.9	-	83.0	-
14	78.2	57.2	55.6	0	90.2	-	68.6	-
15	88.8	65.5	76.3	0	89.5	-	82.3	-
16	99.2	96.7	74.8	0	99.8	-	86.3	-
17	88.0	54.3	75.6	0	97.1	-	85.0	-
18	81.7	51.7	64.9	0	96.3	-	76.5	-
19	96.5	90.6	64.7	0	96.9	-	77.9	-
20	93.1	72.2	94.3	2.6	83.6	100	88.2	50.1
21	93.5	95.0	90.8	0	43.1	-	58.4	-
22	88.7	94.6	69.2	68.5	51.5	81.7	58.7	74.5

C

Table C1: Cloud shadow classification results for Cloud-Net model with kernel size 1.

Index	Accuracy	Precision	Recall	F1 score
1	87.1	92.7	46.9	62.3
2	34.6	99.9	17.3	29.5
3	94.0	87.6	86.0	86.8
4	95.1	62.9	80.4	70.6
5	83.7	95.3	38.7	55.0
6	96.2	74.9	83.8	79.1
7	98.3	84.0	77.1	80.4
8	96.4	71.0	73.0	71.9
9	93.3	86.6	82.9	84.7
10	96.1	99.4	84.0	91.1
11	95.8	90.2	93.3	91.7
12	88.1	89.7	68.0	77.3
13	68.8	1.0	96.6	2.2
14	57.3	0.3	98.9	0.7
15	65.7	0.5	100	1.0
16	96.7	2.0	100	4.1
17	54.8	1.0	99.7	2.1
18	52.0	0.7	97.2	1.2
19	90.6	0.5	100	1.0
20	80.3	31.0	99.8	47.4
21	95.4	9.1	91.4	16.6
22	82.2	81.0	37.5	51.2

D

Table D1: SSIM values for each individual scene before and after correction, for the decomposition of components method.

Index	Blue before	Blue after	Green before	Green after	Red before	Red after
1	0.29	0.17	0.27	0.57	0.28	0.57
2	0.15	0.60	0.17	0.68	0.18	0.69
3	0.14	0.74	0.16	0.75	0.16	0.73
4	0.15	0.64	0.17	0.64	0.17	0.61
5	0.17	0.50	0.16	0.71	0.16	0.75
6	0.16	0.64	0.17	0.70	0.18	0.70
7	0.17	0.50	0.19	0.56	0.19	0.54
8	0.14	0.65	0.16	0.70	0.17	0.69
9	0.18	0.62	0.18	0.70	0.18	0.68
10	0.08	0.82	0.08	0.84	0.08	0.82
11	0.22	0.70	0.24	0.72	0.24	0.70
12	0.16	0.54	0.18	0.62	0.20	0.61
13	0.16	0.57	0.18	0.66	0.19	0.66

E

Table E1: SSIM values for each individual scene before and after correction, for the CNN based method.

Index	**Blue before**	**Blue after**	**Green before**	**Green after**	**Red before**	**Red after**
1	0.29	0.09	0.27	0.10	0.28	0.12
2	0.15	0.21	0.17	0.22	0.18	0.28
3	0.14	0.25	0.16	0.23	0.16	0.29
4	0.15	0.24	0.17	0.23	0.17	0.28
5	0.17	0.19	0.16	0.22	0.16	0.23
6	0.16	0.20	0.17	0.18	0.18	0.24
7	0.17	0.23	0.19	0.20	0.19	0.27
8	0.14	0.21	0.16	0.19	0.17	0.23
9	0.18	0.16	0.18	0.15	0.18	0.19
10	0.08	0.09	0.08	0.10	0.08	0.11
11	0.22	0.26	0.24	0.26	0.24	0.30
12	0.16	0.25	0.18	0.23	0.20	0.29
13	0.16	0.25	0.18	0.26	0.19	0.32

References

Alom, M. Z., Hasan, M., Yakopcic, C., Taha, T., & Asari, V. (2018, 02). Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation.

Baetens, L., Desjardins, C., & Hagolle, O. (2019). Validation of copernicus sentinel-2 cloud masks obtained from maja, sen2cor, and fmask processors using reference cloud masks generated with a supervised active learning procedure. *Remote Sens. 11, no. 4: 433.*. doi: https://doi.org/10.3390/rs11040433

Brownlee, J. (2020). *How to use data scaling improve deep learning model stability and performance.* Retrieved 2021-03-16, from `https://machinelearningmastery.com/how-to-improve-neural-network-stability-and-modeling-performance-with-data-scaling/`

Brownlee, J. (2021). *Gentle introduction to the adam optimization algorithm for deep learning.* Retrieved from `https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/`

Datta, P. (2020). *All about structural similarity index (ssim): Theory + code in pytorch.* Retrieved 2021-06-12, from `https://medium.com/srm-mic/all-about-structural-similarity-index-ssim-theory-code-in-pytorch-6551b455541e`

Domnich, M., Sünter, I., Trofimov, H., Wold, O., Harun, F., Kostiukhin, A., … Boccia, V. (2021, 10). Kappamask: Ai-based cloudmask processor for sentinel-2. *Remote Sensing*, *13*. doi: 10.3390/rs13204100

ESA. (2013). *Sentinel-2 user handbook.*

ESA. (2021). *Level-2a algorithm overview.* Retrieved 2021-03-10, from `https://dragon3.esa.int/web/sentinel/technical-guides/sentinel-2-msi/level-2a/algorithm`

ESA-medialab. (2015). *Sentinel-2: monitoring changing lands.* Retrieved from `https://www.esa.int/ESA_Multimedia/Images/2015/06/Sentinel-2_monitoring_changing_lands#.YdySe6PLHKM.link`

Giamberini, M., & Provenzale, A. (2018). *The birth of a new ocean.* Retrieved from `https://blogs.egu.eu/geolog/2018/02/19/imaggeo-on-mondays-the-birth-of-a-new-ocean/`

Ipia, A., Picoli, M., Câmara, G., Andrade, P., Chaves, M., Lechler, S., … Queiroz, G. (2020, 04). remote sensing comparison of cloud cover detection algorithms on sentinel-2 images of the amazon tropical forest. *Remote Sensing*, *12*. doi: 10.3390/rs12081284

Kalkman, P. (2021). *Increase the accuracy of your cnn by following these 5 tips i learned from the kaggle community.* Retrieved from `https://towardsdatascience.com/increase-the-accuracy-of-your-cnn-by-following-these-5-tips-i-learned-from-the-kaggle-community-27227ad39554`

LeCun, Y., Haffner, P., Bottou, L., & Bengio, Y. (1999). Object recognition with gradient-based learning. In *Shape, contour and grouping in computer vision* (pp. 319–345). Springer Berlin Heidelberg. doi: 10.1007/3-540-46805-6_19

Louis, J. (2021). *Level-2a algorithm theoretical basis document.*

Mamun, I. (2019). *Image classification using ssim.* Retrieved 2021-06-12, from `https://towardsdatascience.com/image-classification-using-ssim-34e549ec6e12`

Mohajerani, S., & Parvaneh, S. (2019). Cloud-net: An end-to-end cloud detection algorithm for landsat

8 imagery. *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 2019, pp. 1029-1032*. doi: 10.1109/IGARSS.2019.8898776.Arxive

Nagare, M., Kaneko, E., Toda, M., Aoki, H., & Tsukada, M. (2018, 07). Cloud shadow removal based on cloud transmittance estimation. , 4031-4034. doi: 10.1109/IGARSS.2018.8517580

Richter, R., & Muller, A. (2005, 08). De-shadowing of satellite/airborne imagery. *International Journal of Remote Sensing*, *26*, 3137-3148. doi: 10.1080/01431160500114664

Ronneberger, O., Fischer, P., & Brox, T. (2015, 10). U-net: Convolutional networks for biomedical image segmentation. *LNCS*, *9351*, 234-241. doi: 10.1007/978-3-319-24574-4_28

Roodschild, M., Gotay Sardiñas, J., & Will, A. (2020, 12). A new approach for the vanishing gradient problem on sigmoid activation. *Progress in Artificial Intelligence*, *9*, 351-360. doi: 10.1007/s13748-020-00218-y

Segal-Rozenhaimer, M., AlanLi, Das, K., & Chirayath, V. (2020). Cloud detection algorithm for multi-modal satellite imagery using convolutional neural-networks (cnn). *Remote Sensing of Environment. 237. 111446.*. doi: 10.1016/j.rse.2019.111446

Sekachev, B., Manovich, N., Zhiltsov, M., Zhavoronkov, A., Kalinin, D., Hoff, B., … Truong, T. (2020, August). *opencv/cvat: v1.1.0.* Zenodo. Retrieved from `https://doi.org/10.5281/zenodo.4009388` doi: 10.5281/zenodo.4009388

Sentinel-Hub. (2021). *Sentinel hub eo browser.* Retrieved from `https://www.sentinel-hub.com/`

Shahtahmassebi, A., Yang, N., Wang, K., Moore, N., & Shen, Z. (2013, 08). Review of shadow detection and de-shadowing methods in remote sensing. *Chinese Geographical Science*, *23*, 403-420. doi: 10.1007/s11769-013-0613-x

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014, 06). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*, 1929-1958.

SURFsara. (2021). *Lisa cluster computer.* Retrieved from `https://userinfo.surfsara.nl/systems/lisa`

Tan, K., & Tong, X. (2016, 11). Cloud extraction from chinese high resolution satellite imagery by probabilistic latent semantic analysis and object-based machine learning. *Remote Sensing*, *8*, 963. doi: 10.3390/rs8110963

Tarrio, K., Tang, X., Masek, J. G., Claverie, M., Ju, J., Qiu, S., … Woodcock, C. E. (2020). Comparison of cloud detection algorithms for sentinel-2 imagery. *Science of Remote Sensing. 2.*. doi: 10.1016/j.srs.2020.100010

Tompson, J., Goroshin, R., Jain, A., Lecun, Y., & Bregler, C. (2015, 06). Efficient object localization using convolutional networks. , 648-656. doi: 10.1109/CVPR.2015.7298664

Tshitoyan, V., Dagdelen, J., Weston, L., Dunn, A., Rong, Z., Kononova, O., … Jain, A. (2019, 07). Unsupervised word embeddings capture latent knowledge from materials science literature. *Nature*, *571*, 95-98. doi: 10.1038/s41586-019-1335-8

Wang, Z., Bovik, A., Sheikh, H., & Simoncelli, E. (2004, 05). Image quality assessment: From error visibility to structural similarity. *Image Processing, IEEE Transactions on*, *13*, 600 - 612. doi: 10.1109/TIP.2003.819861

Wang, Z., & Li, Q. (2010, 11). Li, q.: Information content weighting for perceptual image quality assessment. ieee image proc. 20(5), 1185-1198. *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, *20*, 1185-98. doi: 10.1109/TIP.2010.2092435

Wu, X., & Shi, Z. (2018, 11). Utilizing multilevel features for cloud detection on satellite imagery. *Remote Sensing*, *10*, 1853. doi: 10.3390/rs10111853

Zhu, X., & Helmer, E. (2018, 09). An automatic method for screening clouds and cloud shadows in optical satellite image time series in cloudy regions. *Remote Sensing of Environment*, *214*. doi: 10.1016/j.rse.2018.05.024