CHALMERS
UNIVERSITY OF TECHNOLOGY

A Bayesian machine learning approach to geostationary infrared precipitation retrievals

Master thesis for Engineering mathematics and Computational science

GUSTAV TELLWE

Department of Space, Earth and Environment
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2020

Master's thesis 2020

A Bayesian machine learning approach to geostationary infrared precipitation retrievals

Gustav Tellwe

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Space, Earth and Environment
Chalmers University of Technology
Gothenburg, Sweden 2020

A Bayesian machine learning approach to geostationary infrared precipitation retrievals
Gustav Tellwe

Supervisor: Simon Pfreundschuh, Department of Space, Earth and Environment
Examiner: Patrick Eriksson, Department of Space, Earth and Environment

Master's Thesis 2020
Department of Space, Earth and Environment
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2020

A Bayesian Machine Learning approach to Geostationary infrared precipitation retrievals
GUSTAV TELLWE
Department of Space, Earth and Environment
Chalmers University of Technology

Abstract

This project uses geostationary satellite data to retrieve precipitation rates at surface level. It is achieved through the use of quantile regression neural networks (QRNN) calibrated against rain rates from the Global Precipitation Measurement (GPM) Core Observatory satellite. The area of exploration is located over the Amazon rainforest. The main difficulty of this problem is that geostationary data is not directly related to rain as it only perceives the cloud top temperatures. It does, however, have a high temporal and spatial resolution which makes it interesting for applications in remote areas of the Earth where ground-based radar equipment is unavailable. The result of the project is mainly a comparison between different neural network architectures such as multi-layer perceptron (MLP) and convolutional neural networks (CNN), but there is also a minor comparison to an adapted version of a Hydroestimator (HE) that is currently in use by the National Institute for Space Research (INPE) in Brazil. The best performing configuration, with regards to the loss function, in this study was a CNN. It performed significantly better than the adapted HE for a test conducted over two days in March. An unsuccessful attempt to improve the results using time-series was also conducted. Furthermore, a U-net architecture was also tested on rain rate data that has been resolution-enhanced through interpolation.

Keywords: Deep learning, Remote sensing, Quantile regression, Precipitation, Infrared, Bayesian, GOES, GPM, CNN, U-net

Acknowledgements

Fist, I would like to thank my supervisor Patrick Ericsson for the opportunity to work on his project idea and for his continuous guidance and support throughout these five months. I would also like to thank Simon Pfreundschuh for great support both in the theoretical and implementation part of this project. Without his persistent work on the QRNN this project would not have been possible. Furthermore I would like to direct a special thanks to Teodor Norrestad. This project is many ways a continuation of his work and his willingness to Share his experiences was most helpful. Finally I would like ti direct a special thanks to Daniel Alejandro Vila from the National Institute of space research in Brazil. He was most helpful in providing information and from their models on the same subject.

Gustav Tellwe, Gothenburg, June 2020

Contents

List of Figures

List of Tables

1
Introduction

Water is one of the essential requirements for life on earth. With the current rapid changes in our climate, monitoring and predicting both the global and local water cycles become increasingly important. The project focuses on measuring one of the key aspects of the water cycle - precipitation.
Measuring precipitation is vital for many societal functions. Managing hydropower operations, landslide warning systems and irrigation control are a few examples [21]. Unfortunately, there are some major difficulties. The land areas of interest are huge, and rainfall is in most cases irregular and rare. The physical processes behind precipitation are also highly diverse.

1.1 Current retrieval methods

Today's precipitation measurement methods can to a large extent be divided into three groups. The first one is ground-based equipment such as radar and rain gauges. The second one is orbiting satellite instruments, and the last group is geostationary satellite instruments. Orbital satellites continuously deliver data at a low temporal resolution, typically one or two measurements a day for a specific location. They pass the earth at a relatively low altitude and can thus use radar instruments to increase the quality of the retrieval. Radar is able to penetrate the cloud tops and measure rain directly. The geostationary satellites do not have this ability and instead measures the emitted energy from the earth, thus limiting the information to cloud top temperatures. The stationary position does however improve the temporal resolution to around 4 measurements per hour. They also cover large areas of the earth including the remote places that do not have the ground-based equipment. Because of the high temporal and spatial resolution, improving the quality of the geostationary retrievals is of high interest.

1.2 Geostationary satellite retrievals

The geostationary satellites produce, due to the high temporal and spatial resolution, a huge amount of data making deep learning algorithms suitable. Geostationary satellites usually have multiple instruments on board and the main one used in precipitation analysis is the passive radiometer that measures both energy that is emitted by the earth and reflected sunlight. The wavelengths used for precipitation is fist and foremost infrared. Examples of previous work that use deep learning for geostationary precipitation retrievals are the PERSIANN algorithms [10]. There are also products using other statistical methods. One

group of such algorithms are hydroestimators (HE) [22] that uses a non-linear power-law regression. The results of this project will be compared to an adapted version of a HE.
Infrared radiation capture the cloud top temperatures. The main idea for using this information to extract precipitation is that cold clouds should indicate a larger vertical cloud development which would then also implicate higher precipitation [21]. But this relationship is not clearly defined and other information such as cloud formations also play a role. Since deep learning can learn relationships in the data by itself, it becomes an interesting method for a problem like this one.

1.3 Aim and scope

This report aims to evaluate two new ideas in the realm of geostationary satellite precipitation retrievals. The first one is to use a network type developed by the Earth, Space and Environmental department at Chalmers called quantile regression neural networks (QRNN) [18]. This network type predicts quantiles of the probability distribution for a precipitation measurement. The second idea is to use the GPM satellite data as a reference for training. Radar or rain gauges are most common when calibrating the geostationary models but GPMs advantage of monitoring more remote areas of the earth makes it an interesting option. To limit the scope of the project, only a small area over the Amazon rain forest is being examined. The area is ranging from $-70\,^{\circ}$W to $-51\,^{\circ}$W and $-11\,^{\circ}$N to $2.5\,^{\circ}$N. Common neural network architectures can be applied within the QRNN framework and this project will explore Multi-Layer Perceptron (MLP), Convolutional Neural Network (CNN) and a more recent development of CNN called U-Net.
The results are limited to 15-minute temporal and $5\,\text{km} \times 5\,\text{km}$ spatial resolution. There will be evaluations of how lower spatial resolution affects the results. The best performing model will also be compared to an adapted version of a HE currently in use by the National Institute of Space Research in Brazil.

2
Neural Networks

This chapter provides the necessary background information in the field of artificial neural networks (ANN). First, there is a brief overview of what neural networks are and how they function. Then there are sections about the different architectures. After that, the QRNN framework is explained and finally, some methods of measuring the performance of QRNNs are presented.

2.1 Introduction

The smallest part of an ANN is called a neuron and it is a computational unit. The neuron takes inputs in the form of a vector and performs some type of computation to produce one output. The computation is usually done by multiplying the input, x, by a weight vector, w, adding a bias, b, and then finally a non-linear activation function, f, to produce the output. The computation is as follows:

$$\text{output} = f(\sum_i x_i w_i + b). \tag{2.1}$$

One example of an activation function is the Rectified linear unit (ReLu) and calculated as:

$$ReLu(x) = \begin{cases} x & x > 0 \\ 0 & x \leq 0. \end{cases} \tag{2.2}$$

Neurons are then assembled in different ways to form a network. Figure 2.1 shows one example of such a network. The idea behind an ANN is in a rough sense to copy our brain. Sensory input results in a chain reaction of our neurons causing some desired action. We learn by redirecting what neurons are being activated.

In the ANN, the connections are represented by weights. The weights determine where a neuron should send its information by either increasing or decreasing the signal strength. The activation function acts as a threshold for the neuron. Information is only passed on if the combined input signal strength crosses the threshold. The network learns by adjusting the weights and thus creating a model that responds in the desired way depending on the type of input data.

This study only includes feed-forward neural networks. These are networks that are limited to only sending information forward in the network and thus avoids forming cycles.

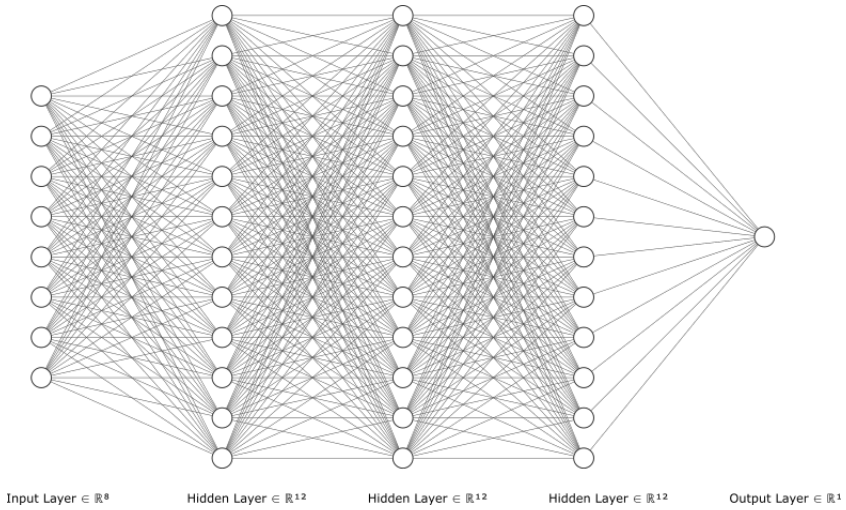


Figure 2.1: Example of an Artificial Neural Network [15]

2.2 Multi layer perception

The simplest of the feed-forward ANNs is the multi-layer perceptron (MLP). Each layer is defined by a set of neurons where each neuron takes input from each of the neurons in the previous layer. This is called fully connected. The deeper the network is, i.e the more layers it has, the more complex tasks can be handled. But the size of the network also makes it more susceptible to overfitting. That is when the model is not learning the underlying properties of the dataset but adjust to the noise of the dataset which makes it good to recognise the training data but bad at handling unseen data. Layers of fully connected neurons are also called dense layers. Figure 2.1 shows an example of an MLP.

2.3 Convolutional Neural Networks

Convolutional neural networks (CNN) where developed in the field of image recognition and have produced great results. Compared to MLP, CNN introduces some new layers starting with the convolutional layer [7]. This can be thought of as a window of weights, kernel, that is being placed over a part of the image and then slides across it. For each slide, it multiplies the weights in the window with the input from the image and produces one output. Figure 2.2 shows an illustration of this. In image recognition, this works as a feature detector. Each kernel learns to detect features in the image. For examples edges or dots. By making the network deeper and connecting more convolutional layers after one another, more complex features can be learned such as faces or cars.
The next new structure often used in convolutional neural networks is the pooling layers. These layers handle the issue of invariance in the network as a whole. The convolutions described above create features maps with the precise location still encoded. This causes small changes in the input to result in very different reactions throughout the network. Examples of solutions to this problem are to either increase the step that the kernels takes for each move or to add a pooling layer. This layer slides a window, similar to the CNN, but instead of calculating the matrix multiplication and activation, it does a pooling operation.

The most common is max-pooling which takes the maximum value on the field of view. The stride of this window is then equal to the window size resulting in a downsampled feature map.
Most common for a CNN is that the outputs of the convolutions are fed into a few dense layers to produce the desired output. Figure 2.3 shows an illustration of a complete CNN.

2.4 U-Net

U-net is an architecture first implemented for image segmentation in the biomedical field [19]. It is designed to produce an image of roughly the same size as the input. This is done through a fully convolutional neural network meaning there are no fully connected layers and only convolutional layers and pooling layers. The network structure used in this project can be found in the method chapter below. Figure 2.4 shows the networks structure from the original article [19]. The network can be split into two parts. Encoder and decoder, much like an autoencoder. The encoder consists of blocks of convolutional layers and pooling layers. This produces a downsampled feature representation of the input. The decoder part does the opposite. It upsamples the features, through transpose convolutional layers, to the original size. During the pooling in the encoder, some spatial information is lost. If the features extracted in the middle of the network was only upscaled, the decoder has a limited way of knowing where the features are located. To fix this, skip connections are introduced. A skip connection pass the activation's right before a pooling layer to the result of the corresponding upsampling layer. This will allow for spatial information to flow through the network.

2.5 Training

For the ANN to be able to learn, it must have some sort of measure of how it wants to optimise the output. In this case, the data is pared with a label providing the requirements for supervised learning. The output of the neural network and the label can now be compared through a loss function. The loss function describes how the output should relate to the

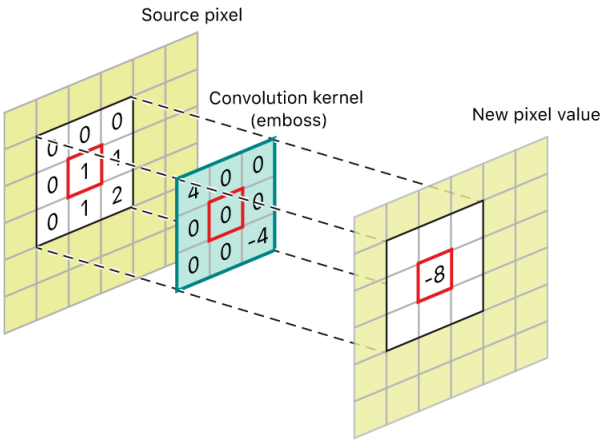


Figure 2.2: Illustration of how the kernels work in convolutional layers [4].

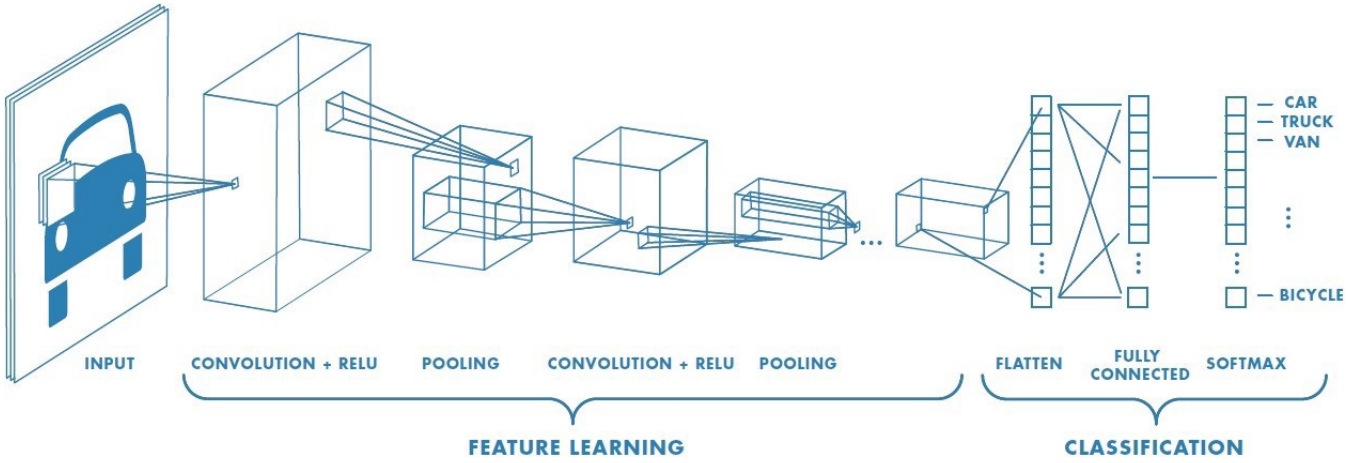


Figure 2.3: Illustration of a complete CNN structure [14].

labels. The most common example is the mean squared error. It computes the distance between the output and the label squared as described by the following equation:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2. \tag{2.3}$$

Here $\hat{y}_i$ is the predicted value and y_i is the label and N is the total samples. The network can now be viewed as one large function. The parameters of the function are the weights an biases. The task of training a network is the same as finding the best minima, i.e it is an optimisation problem. The problem cannot be solved in any analytical way so numerical methods are used instead. The first method created for this purpose is the stochastic gradient descent. It finds a better set of weights and biases by calculating the gradient and then takes a step in the negative direction of it. The gradient is found using an algorithm called backpropagation. The backpropagation algorithm calculates the gradients layer by layer, starting from the output and using the chain rule on the loss function [9].

2.6 Quantile regression neural networks

One way of producing outputs for rain measurements is that for every input $x \in R^m$ have one output value $y \in R$. This approach lacks uncertainty estimates on the measurement. An alternative way is to use a Bayesian framework. The task can be reformulated to not predict the single value but instead, a probability distribution $p(y|x)$. The QRNN does this by approximating the quantiles of the distribution function $p(y|x)$. This is achieved through a special loss function and it derives from the following equation:

$$L(\hat{y}_\tau, y) = \begin{cases} \tau|y - \hat{y}_\tau|, & \hat{y}_\tau < y \\ (1-\tau)|y - \hat{y}_\tau|, & \text{otherwise} \end{cases} \tag{2.4}$$

where y is the label, $\hat{y}_\tau$ is the prediction for the τ quantile [18]. Let g be a probability distribution and $F(\hat{x})$ be the corresponding cumulative distribution function. It can be shown that the τth quantile of the cumulative distribution function $F(\hat{x})$ minimises the

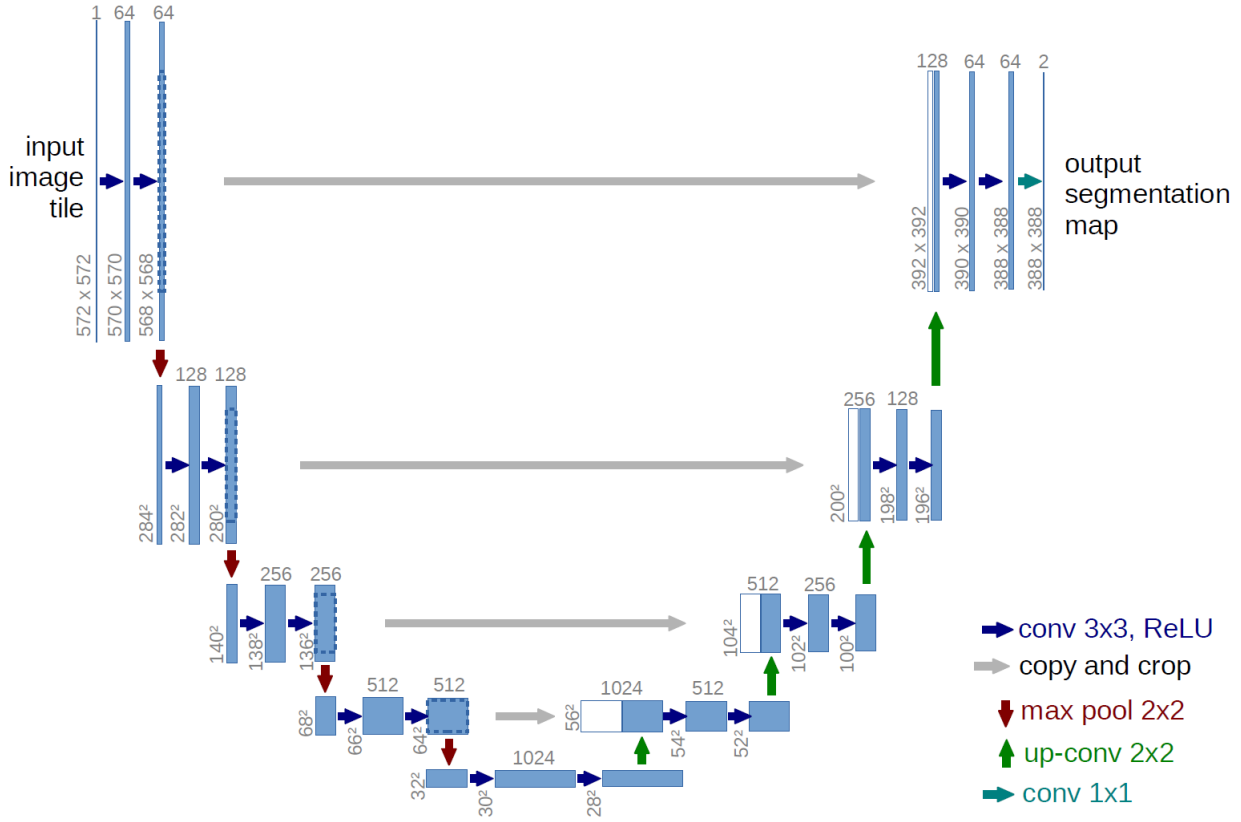


Figure 2.4: The original U-net architecture [19].

expectation value of $L(\hat{x}_\tau, \hat{x})$ where $\hat{x}_\tau$ is the τ quantile of $F(\hat{x})$ and it is defined as [13]

$$\hat{x}_\tau = \inf\{\hat{x} \ : \ F(\hat{x}) \geq \tau\}. \tag{2.5}$$

By letting the loss function be the mean of equation (2.4) over the training set, the QRNN learns to approximate the quantiles of the conditional distribution $p(y|x)$. Thus the network will have one output for every quantile that it wishes to predict and a probability distribution can then be approximated. An ilustration of a complete QRNN is shown in figure 2.5

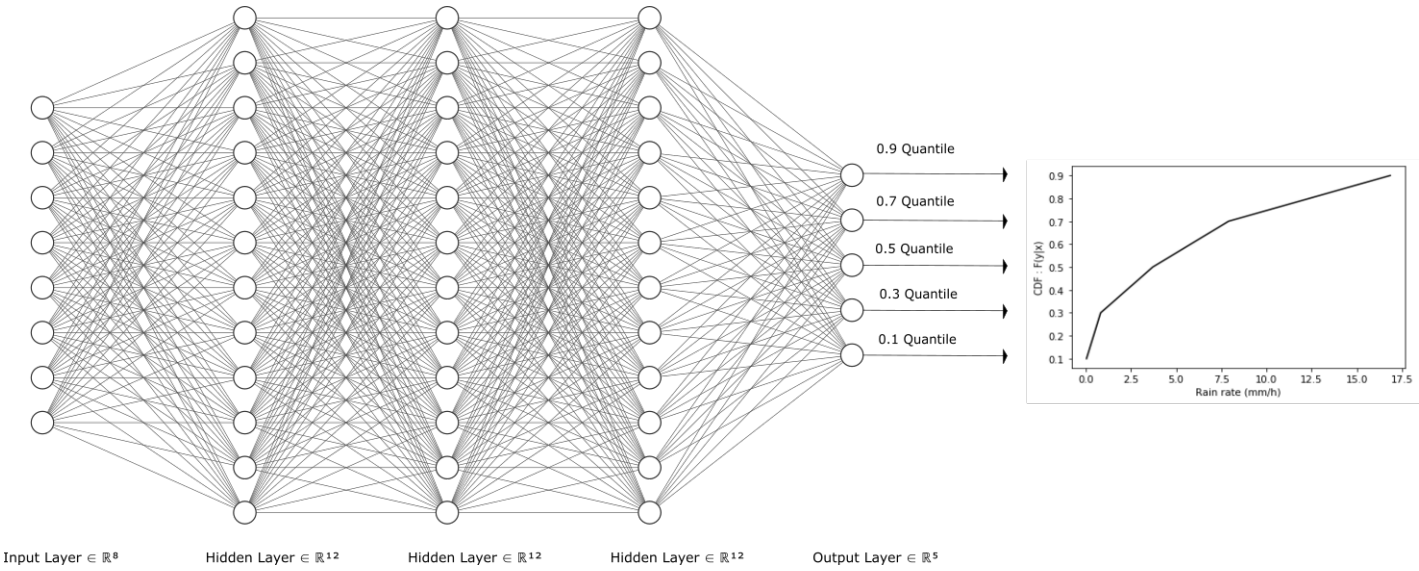


Figure 2.5: Illustration of a complete QRNN. The output values can be used to construct an approximation of the cumulative distribution function of the rain rate prediction. This is exemplified by the graph at the end of the network.

3
Data

This chapter provides background information regarding the data for this project. It starts with a brief overview of remote sensing and then moves on to the satellites and their respective instruments.

3.1 Remote sensing

Remote sensing is the measuring of a phenomenon without making physical contact with it. It can be divided into active and passive. Active is the gathering of information by sending and receiving different types of signals. One example is radar. By measuring the return signal, particles such as rain and snow can be detected at multiple layers in the cloud. Passive remote sensing, on the other hand, measures the energy that has been either reflected or emitted by the earth. The wavelength spectra of interest are from 0.3 µm to around 14 µm. Shorter wavelengths contain more reflected energy and longer wavelengths contain more of the thermally emitted energy. The breaking point of where the radiation is mainly emitted is at around 3 µm [20]. The emitted energy originates from the fact that every object with a temperature above 0 K emits thermal radiation due to kinetic energy. Figure 3.1 shows an illustration of passive and active remote sensing.

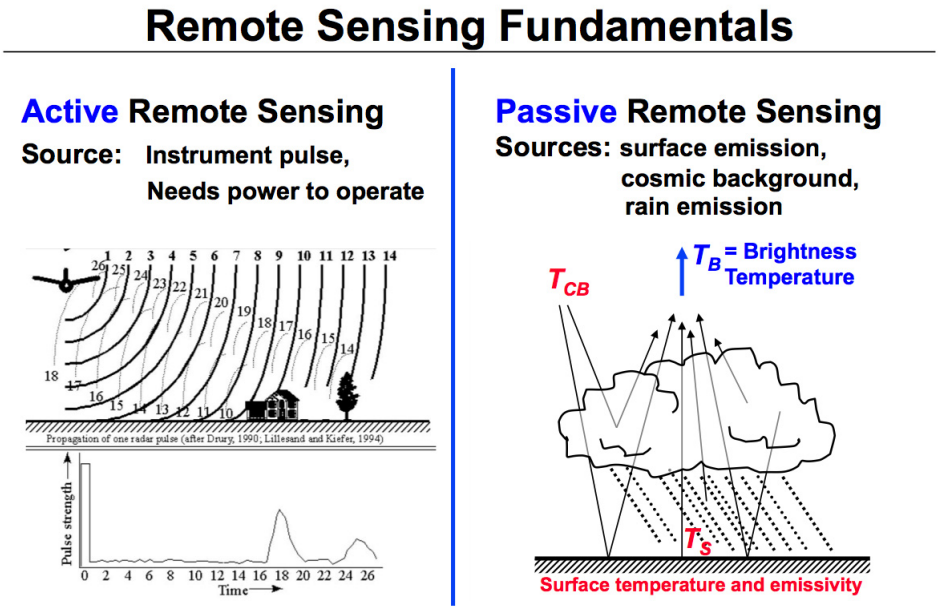


Figure 3.1: Illustration of active and passive remote sensing [16].

3.2 GOES

The input data for this project is created by the Advanced Baseline Imager (ABI) instrument aboard the Geostationary Operational Environmental Satellites (GOES) - R series operated by the National Aeronautics and Space Administration (NASA) and the National Oceanic and Atmospheric Administration (NOAA) [2]. The satellite was launched in November 2016 and is located over the western hemisphere. ABI is a passive imaging radiometer producing images from 16 different spectral bands ranging from 0.37 µm to 13.3 µm in wavelength. Table 3.1 shows the basic information about the different channels. The instrument has a

Table 3.1: ABI channels

ABI Band	Wavelength [µm]	Type	Nickname	Best spatial resolution [km]
1	0.47	Visible	Blue	1
2	0.64	Visible	Red	0.5
3	0.86	Near-infrared	Veggie	1
4	1.37	Near-infrared	Cirrus	2
5	1.6	Near-infrared	Snow/Ice	2
6	2.2	Near-infrared	Cloud particle size	2
7	3.9	Infrared	Shortwave window	2
8	6.2	Infrared	Upper-level water vapor	2
9	6.9	Infrared	Midlevel water vapor	2
10	7.3	Infrared	Lower-level water vapor	2
11	8.4	Infrared	Cloud-top phase	2
12	9.6	Infrared	Ozone	2
13	10.3	Infrared	"Clean" longwave window	2
14	11.2	Infrared	Longwave window	2
15	12.3	Infrared	"Dirty" longwave window	2
16	13.3	Infrared	CO_2 longwave	2

couple of different modes in which it changes where and how frequent different areas are scanned. The full disk image, see Fig. 3.2, takes about 5 minutes to create. One of the modes, mode 3, only produces these images. Other modes also create sub-domain images at a higher temporal resolution, up to 60-second intervals resulting in a 10-minute temporal resolution for the full disk image. This project only uses the full disk images since the area of interest is not included in the sub-domains created by the GOES-R satellite. The main part of the data is collected at 10-minute temporal resolution since the mode 3 is rarely active. Notable is also that the data is labelled with start and end time of the full disk scan and thus every point in the image is not labelled with a precise time.

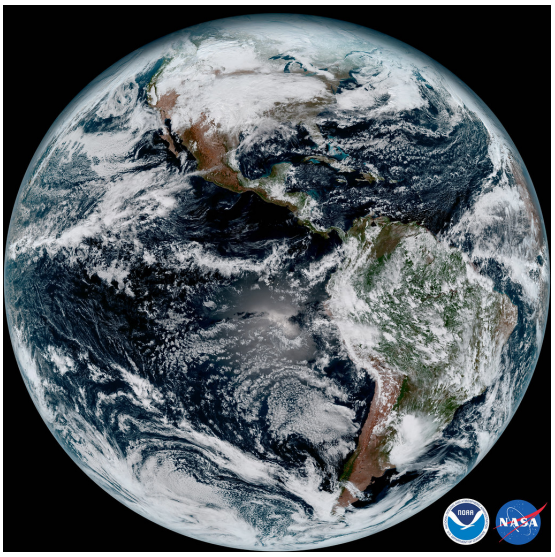


Figure 3.2: Full disk image from GOES-16 [24].

3.3 GPM

The label data collected for this project is taken from the Global Precipitation Measurement (GPM) Core Observatory satellite (GPMCO). This orbital satellite was launched in February 2014 by NASA and the Japanese Aerospace Exploration Agency (JAXA) [1]. The satellite is equipped with two different types of instruments, GPM microwave imager (GMI) and Dual-Frequency Precipitation Radar (DPR). GMI uses 13 microwave channels to scan an area of 885 km at a time and the DPR uses two radar frequencies and scans 125 km and 256 km respectively. The data available from the GPM is ordered in a series of different products. The product used in this project is the 2BCMB product which combines the GMI and DPR to produce the most accurate measurements and has a spatial resolution of $5\,\text{km} \times 5\,\text{km}$.
The quality of measurements for precipitation tools including GPM is not easy to verify since there is no currently available method to collect all precipitation over a large area. Even ground-based rain gauges are collecting data at a high spatial resolution leading to uncertainties. The accuracy of the GPM measurements is discussed in a paper by Grecu et al. [8]. They compare the results from the combined GPM with a database of ground-based measurements called Global Precipitation Climatology Project (GPCP) [11]. Figure 3.3 shows their results and at low latitudes, there is a very good agreement. Higher latitudes have worse accuracy but since this project is evaluated over the Amazon region, this will not be of concern.

3.4 Hydroestimator (HE)

The current model in use by the National Institute for Space Research (INPE) over Brazil is and adapted version of a Hydroestimator (Siqueira and Vila [6]). The original hydroetimator (Vicente et al [22]) uses a non-linear power-law regression relationship between cloud top temperature from channel 8 of the GOES ABI imager and ground-based radar observation.

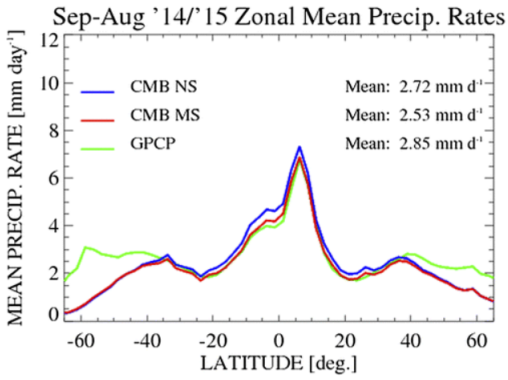


Figure 3.3: Comparison of the Combined product from the GPMCO satellite versus the Global Precipitation Climatology Project [11]. The blue and red lines are two different GPMCO products and the green is the data from the Global Precipitation Climatology Project (GPCP)

The adapted version improved the results of the original hydroestimator by implementing histogram matching of the hydroestimator with other more accurate satellite-based products such as GPM and a similar constellation called Tropical Rainfall Measuring Mission (TRMM) [12]. The result is a $4\,\text{km} \times 4\,\text{km}$ single value rain rate estimation.

4

Methods

There are two parts to this project. The first is collecting and creating the datasets for the QRNNs. The other is implementing, training, and testing the networks. As mentioned in the introduction there will be different architectures and different label area sizes explored. These configurations require different types of datasets. There will be in a total of 4 different datasets and 8 different configurations. Only a window over the Amazon rain forest is examined and the precise coordinates for the window are from $-70\,^{\circ}\mathrm{W}$ to $-51\,^{\circ}\mathrm{W}$ and $-11\,^{\circ}\mathrm{N}$ to $2.5\,^{\circ}\mathrm{N}$. Figure 4.1 shows a map of the area.

4.1 Data

The raw data for is available online. It is downloaded from two different sources, one for GOES [17] and one for GPM [23]. The data then has to be combined to create inputs with a corresponding label. The following sections describe how this is done.

Figure 4.1: Map of South America where the rectangle represents the area that this project collects data from.

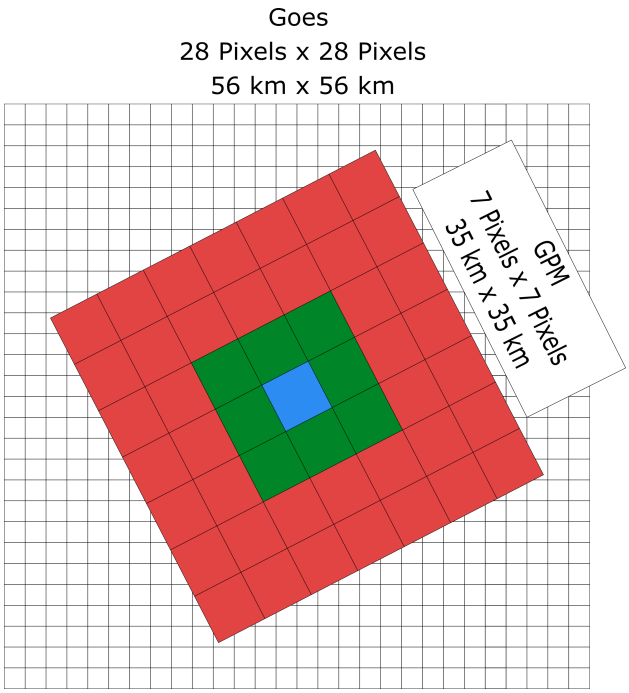


Figure 4.2: Visualisation the spacial relationship between the GOES input data and GPM label data. The different colours represent different areas used in an evaluation of what resolution is best for the label data. See the results section for more information.

4.1.1 Datasets

The first dataset, dataset 1, is used for the single 5 km × 5 km rain rate prediction. Dataset 2 is for the U-net. Dataset 3 is for evaluating different area sizes for the labels and dataset 4 is for the time-series. In dataset 1, the geostationary data consists of 28 pixels × 28 pixels resulting in a total area of 56 km × 56 km. The corresponding label data is arranged in a 7 pixels × 7 pixels image resulting in a total area of 35 km × 35 km. The geostationary data is centered around the middle pixel of the label data. See Fig. 4.2 for a visual representation. A sample of the training data is shown in appendix A in Fig. A.1. This allows training QRNNs on either single center pixel or the mean of an area up to 35 km × 35 km. The distributions of time and distance differences between the GPM and GOES can be found in Fig. 4.3. The time difference is calculated using the middle time of the full disk scan and the GPM time. Dataset 3 has the same 7 pixels × 7 pixels pixel label data as dataset 1 but instead of only having the input data centred at the middle pixel, the input data centred at all the other pixels are also present. The reason for not having only dataset 3 is that it would consume 49 times the memory to store that dataset and it would not be very useful since the images cover more or less the same area thus creating a dataset with a high degree of redundant data. In dataset 2, the input data have a size of 200 km × 200 km and 100 pixels × 100 pixels. The label data is 200 km × 200 km and 100 pixels × 100 pixels. Dataset 4 has the same data structure as dataset 1 except that it has 3 images per channel. One before the time of measurement, one closest to and one after the measurement. Because of some random sampling in the creation of the dataset, it does not cover the exact same labels. This will be explained later.

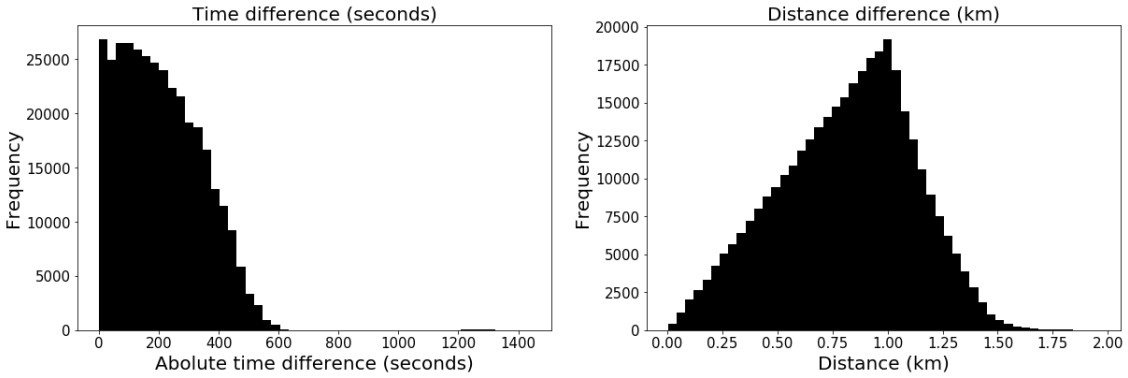


Figure 4.3: Distribution of the time and distance differences from dataset 1. The time is in seconds and distance is in km latitude.

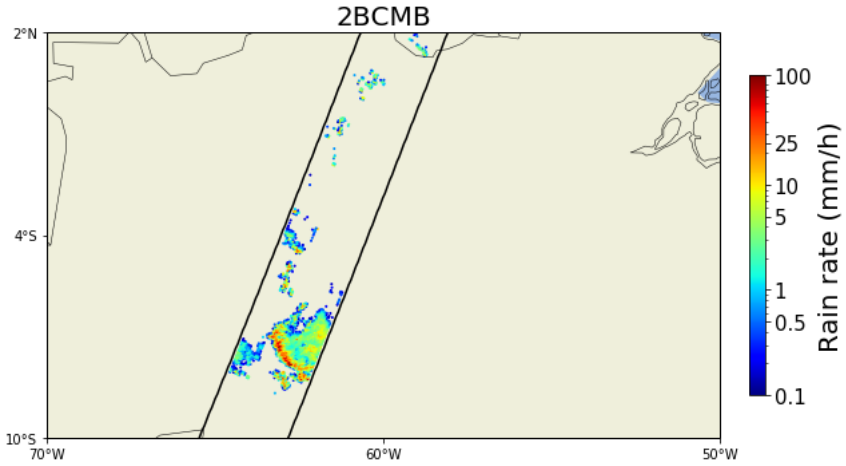


Figure 4.4: Example of rain rates from the combined product, 2BCMB, from the GPMCO satellite. The black lines are boundaries of the swath.

4.1.2 Input Data

The input data is created using the 8 and 13 channels from the ABI. There have been some experiments trying to compare what channels provide useful information [5]. The main channel is 13. As discussed by Behrangi et al. (2009) [5] the upper-level water vapour channel adds some significant information. These channels are also available at the other geostationary satellites around the world which makes the model more applicable [3]. Each data point has its corresponding longitude and latitude position. Each dimension of the input is scaled to have mean zero and unit variance over the training part of the datasets.

4.1.3 Label data

The combined product, 2BCMB described in section 3.3, is used for the label data. The resulting data matrices created by the GPMCO and GOES are not perpendicular to on another due to one being geostationary and one being orbiting. Figure 4.4 shows an example of how the GPMCO moves over the area. Dataset 1 samples the label data from the point of reference of the GPMCO pass resulting in the data being located like a diamond inside

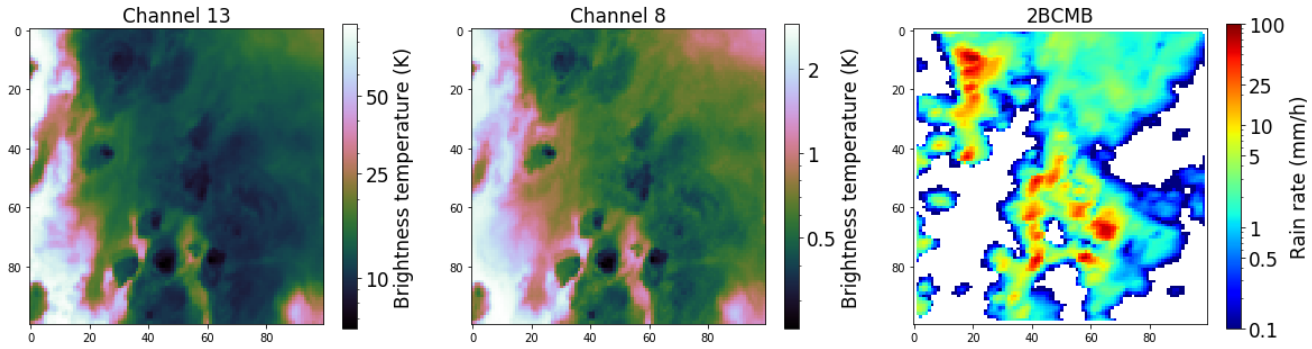


Figure 4.5: Example of images from dataset 2. The area is 100 km × 100 km. The units on the axes is pixels.

Table 4.1: Size of the datasets

Dataset number	Training samples	Testing samples	Total samples
1	175000	175000	350000
2	3100	3100	6200
3	0	3200	3200
4	160000	160000	320000

the GOES data and the centre pixel having the closest distance possible to the GOES middle point, see Fig. 4.2. The angle and direction of the GPMCO pass vary over the area resulting in different rotated diamonds for different samples. Dataset 2 consists of input data and label images that have the same coordinate system. That is achieved by selecting GPM data in a square with the same reference as GOES and then interpolating the points linearly to a mesh grid with the same pixel density as the resolution of the GOES Data, i.e 2 km. The result is found in Fig. 4.5. This creates a considerable scale-down of GPM data and is a source of errors as discussed later in the report. Since the QRNN approximates a probability distribution, the label data must represent the true a priori distribution. To achieve this, the data is collected over a whole year to decrease the fluctuations of rain/dry season. The data is collected at every GPMCO satellite pass the area in question but to limit the dataset size all points for each pass is not included and instead, a random sample is collected. The sizes of the training sets are displayed in Tab. 4.1. We see that dataset 3 consists only of testing data. Dataset 2 is considerably smaller and due to the limitation of images with the width of 200 km that can be produced from the GPMCO passes. The small size of dataset 3 is because of the 49 times increase in input images for each sample. The training data for all datasets are collected from august 2017 to a year later and the test data the following year. Notable is that the samples with rain are rare and heavy rain is really rare. For example, the fraction of rain samples in the training set of dataset 1 is 0.076 and for the test set, it is 0.08. The distribution of rain rates in the interval $(0, 20]$ mm/h for dataset 1 is displayed in Fig. 4.6

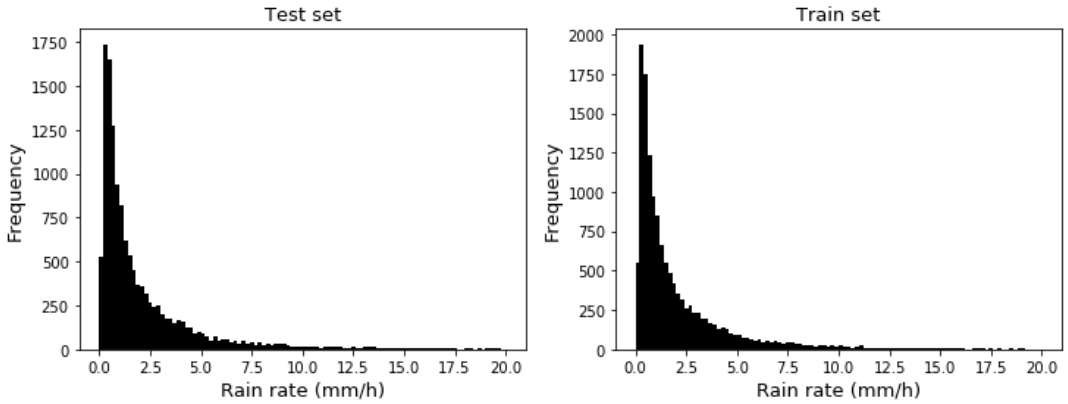


Figure 4.6: Distribution of the rain rates in the range (0,20] mm/h for dataset 1. The no rain labels are excluded.

4.2 Networks

The network structures examined in this project is Multi-layer Perceptrons (MLP), regular Convolutional Neural Network (CNN) and U-net. The network structures can be found in Tab. 4.2. Different structures for each architecture type was tested but for this report, only the best ones found are included. The main challenge was adjusting the network size so that no over-fitting occurred. The training implements Stochastic gradient descent with a learning rate decay that downscales the rate by a factor of two every ten consecutive steps that failed to decrease the validation loss. The initial learning rate was set to 0.01. The activation function is ReLu in all of the networks and all layers except the last where there is no activation. Deep learning parameters such as learning rate, optimiser, activation functions etc. are not thoroughly tested. The initial parameters were taken from the QRNN paper [18].

4.3 Interpreting the results

Because of the probabilistic nature of the QRNN, the interpretation of the results and the measurement of its performance is a little bit different than the more conventional classification deep learning algorithms. The ideal output should be sharp, i.e meaning that the predictions should not be spread out over a large interval, and they should be well-calibrated meaning that the predicted quantiles reflect the true probability distribution. One of the tools used to measure the calibration is calibration plots. It is produced by observing the frequency of which a value fails to lie within a confidence interval produced by the QRNN. The most straight forward implementation of this is to count the fraction of times the true value is below the quantile value and plotting it against the quantile. In a well-calibrated QRNN, this plot should follow the straight line $y = x$. The sharpness can be measured by, for example, the distance of the confidence intervals.

Some measurements take both the sharpness and the calibration into account. As mentioned by Pfreundschuh et. al [18], the continuously ranked probability score (CRPS) is one of those

measures.

$$\mathrm{CRPS}(F, x) = \int_{-\infty}^{\infty} (F(\hat{x}) - I_{x \leq \hat{x}})^2 d\hat{x} \tag{4.1}$$

F is the cumulative distribution function, I is the indicator function that is equal to one as the condition $x \leq \hat{x}$ is true and 0 otherwise.
Another group of performance indicators is produced by using the network as a conventional predictor with a single prediction value. One such prediction, $\hat{y}$, for a QRNN is found through the expectation value of the output distribution. Table 4.3 displays the scores for this purpose used in this project. The Mean squared error quota (QMSE) and Absolute error quota (QMAE) are intended to be more independent of what dataset is used. It achieves this by comparing the predictions against the mean value of the training set. The quota will indicate how much the network learns and does this with a lower dependence on the label. These measures will be useful since the a priori distribution is different when the GPM resolution change. It can thus be a better score when comparing configurations with different datasets. POD, CSI and FAR show how the model predicts rain and no rain samples. TP is the number of times that the model predicts rain and it was rain. FP is the false positive number, i.e the number of times that the model predicted rain and there was no rain. MS are the missing values indicating rain occurrences that were not predicted. POD shows the probability of detecting rain. FAR shows how many of the predicted rain occurrences are false. CSI shows a combined measurement of the previous two.

Table 4.2: Network structures

Network number	Type	Structure
1	MLP	8 FC with 256 neurons each, activation ReLu
2	CNN	Conv2d(32, (3,3))→ Conv2d(32, (3,3))→ MaxPool((2,2))→ Conv2d(64, (3,3))→ Conv2d(64, (3,3))→ MaxPool((2,2))→ Conv2d(128, (3,3)→ Conv2d(128, (3,3))→ MaxPool((2,2))→ Conv2d(256, (3,3))→ Conv2d(256, (3,3))→ MaxPool((2,2))→ FC(128)→FC(128)→FC(128)→FC(5)
3	U-net	Conv2d(32, (3,3))→ Conv2d(32, (3,3))→ MaxPool((2,2))→ Conv2d(64, (3,3))→ Conv2d(64, (3,3))→ MaxPool((2,2))→ Conv2d(128, (3,3)→ Conv2d(128, (3,3))→ MaxPool((2,2))→ Conv2d(256, (3,3))→ Conv2d(256, (3,3))→ MaxPool((2,2))→ Conv2d(512, (3,3))→ Conv2d(512, (3,3))→ Conv2d(256, (3,3))→ Conv2d(256, (3,3))→ Conv2dTranspose((2,2))→ Conv2d(128, (3,3))→ Conv2d(128, (3,3))→ Conv2dTranspose((2,2))→ Conv2d(64, (3,3)→ Conv2d(64, (3,3))→ Conv2dTranspose((2,2))→ Conv2d(32, (3,3))→ Conv2d(32, (3,3))→ Conv2dTranspose((2,2))→ Conv2d(32, (2,2), stide = 2, padding = none)→ Conv2d(32, (5,5), padding = none)→ Conv2d(32, (7,7), padding = none)→ Conv2d(32, (7,7), padding = none)→ Conv2d(5, (1,1), padding = none, stride = 1)
4	CNN	Conv3d(128, (1,3,3))→ Conv3d(128)→ MaxPool3D((1,2,2))→ Conv3d(256, (1,3,3))→ Conv3d(256, (1,3,3))→ MaxPool3D((1,2,2))→ Conv3d(512, (1,3,3)→ Conv3d(512, (1,3,3))→ MaxPool3D((1,2,2))→ FC(256)→FC(256)→FC(256)→FC(256)

Table 4.3: Verification Measures, $\hat{y}$ is the prediction and y is the target. $\bar{\mathbf{y}}$ is the mean of the a priori distribution. N is the total number of samples

Verification Measure	Formula	Range and Optimal Values
Mean squared error	$\text{MSE} = \frac{1}{N}\sum_i (\hat{y}_i - y_i)^2$	Range: (0,∞); Optimal: 0
Mean abslute error	$\text{MAE} = \frac{1}{N}\sum_i \lvert\hat{y}_i - y_i\rvert$	Range: (0,∞); Optimal: 0
Mean squared error quota	$\text{QMSE} = \frac{\sum_i (\hat{y}_i - y_i)^2}{\sum_i (\bar{\mathbf{y}} - y_i)^2}$	Range: (0,∞); Optimal: 0
Absolute error quota	$\text{QMAE} = \frac{\sum_i \lvert\hat{y}_i - y_i\rvert}{\sum_i \lvert\bar{\mathbf{y}} - y_i\rvert}$	Range: (0,∞); Optimal: 0
Median squared error	MEDSE = *Median of the squared errors*	Range: (0,∞); Optimal: 0
Median absolute error	MEDAE = *Median of the absolute errors*	Range: (0,∞); Optimal: 0
Pearson's Correlation Coefficient	$\text{CORR} = \frac{E[(\hat{y} - \mu_{\hat{y}})(y - \mu_y)]}{\sigma_{\hat{y}}\sigma_y}$	Range: (-1,1); Optimal: 1
Probability of detection	$\text{POD} = \frac{TP}{TP + MS}$	Range: (0,1); Optimal: 1
Failure attempt ratio	$\text{FAR} = \frac{FP}{FP + MS}$	Range: (0,1); Optimal: 0
Critical sucess index	$\text{CSI} = \frac{TP}{TP + FP + MS}$	Range: (0,1); Optimal: 1

5
Results and discussion

Many different configurations were tested in this project. Tab. 5.1 shows an overview and figure 4.2 shows a visualisation of the target types used in the configuration, not including the U-net case.

5.1 Evaluation of different label resolutions

It is not obvious that predicting the highest resolution, i.e 5 km × 5 km, will lead to the best performance. There might be too much noise in the input image to predict the smaller areas.

5.1.1 Description of the configurations

The comparison of Conf. 1,2,3,7 and 8 are intended for the investigation of what label resolution should be used. They all use network number 1 which is the dense feed-forward network without any convolutional layers. Configurations 1,2 and 3 all use dataset 1. Configuration 1 has a single pixel 5 km × 5 km prediction. Configuration 2 is trained with the mean of 3 pixels × 3 pixels and Conf. 3 is the same but with 7 pixels × 7 pixels. See Fig. 4.2 for a visualisation. Configurations 7 and 8 use another dataset, dataset 3. Configuration 8 takes the same approach as Conf. 4 as it is trained on the mean of the 7 pixels × 7 pixels

Table 5.1: Network structures

Conf.	Network	Input width	Target	Dataset	Mean before or after training
1	1	56	centre pixel value	1	Non-applicable
2	1	56	mean of centre 3x3 pixels	1	Before
3	1	56	mean of centre 7x7 pixels	1	Before
4	2	56	centre pixel value	1	Non-applicable
5	4	56	centre pixel value	4	Non-applicable
6	3	200	100x100 pixel values	2	Non-applicable
7	1	56	mean of centre 7x7 pixels	3	After
8	1	56	mean of centre 7x7 pixels	3	Before

rain rate label data but Conf. 7 predict every single pixel in the 7 pixel × 7 pixel rain rate label data and then takes the mean after training. All configurations have 56 km × 56 km input from the 8 and 13 channels.

Table 5.2: Verification measures for the exploration of different rain rate resolutions

Conf.	QMSE	QMAE	CORR
1	0.952	0.639	0.250
2	1.050	0.668	0.268
3	0.875	0.658	0.425
7	0.739	0.545	0.549
8	0.952	0.613	0.434

5.1.2 Results for different label resolutions

Table 5.2 shows error and correlation results for the configurations mentioned above. Configuration 3 performs better in both correlation and QMSE. Configuration 2 shows no improvement over Conf. 1 which is strange considering predicting the mean value of a larger area should be easier. The comparison of Conf. 7 and 8 indicates that training the network on the mean value does not increase performance. This is concluded from the fact that the correlation, QMSE and QMAE are all considerably better for predicting single pixel values, i.e Conf. 7 outperforms 8. It might, however, be because of the inconsistent orientation of the diamond-shaped GPM data with relation to the GOES input data as mentioned in the method section. The single-pixel prediction will be used for the remainder of this report.
Worth noting is also the difference in QMSE for Conf. and Conf. 8. They should be similar but this difference might be an indication of the dependence of the dataset for the QMSE. The mean squared error is highly dependent on the labels with rain and even more dependent on high rain values. Since these are so rare, the QMSE might be less reliable.

5.2 Comparison of CNN and MLP

This section presents a more extensive evaluation of Conf. 1 and 4. Configuration 4 predicts the same single pixel value as Conf. 1 over the same dataset but uses a CNN architecture. The first part of the evaluation are results generated using the expected values of the prediction distributions

5.2.1 Expected value prediction measures

The distribution of rain rates for the expected value predictions and the corresponding labels are visualised through the histograms of Fig. 5.1. Note that the frequencies are log scaled. Notable is that the label rain rates do not contain small values, below 0.1 mm/h, but the predicted rain rates have values in that interval. QRNN predicts values in that interval because there is no built-in threshold that sets predictions under a certain value to be 0.

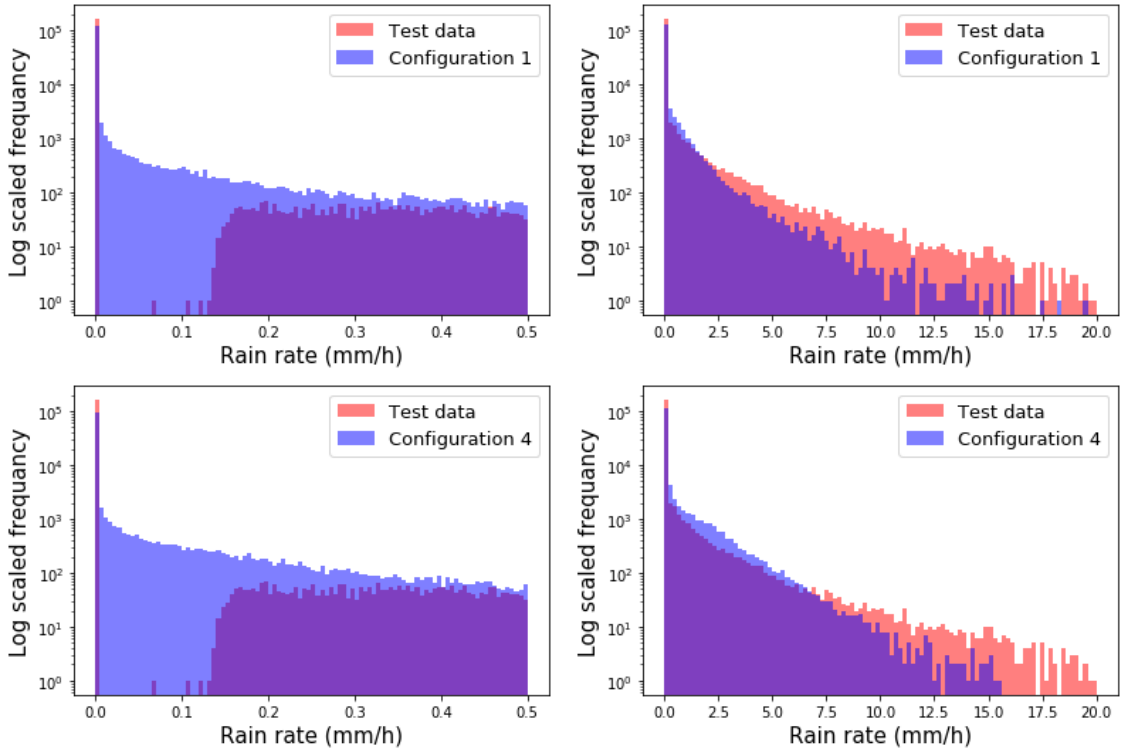


Figure 5.1: Histograms of the expected value predictions, blue, and test set, red. The left plot is an enlargement over the interval $(0, 0.5]$ mm/h. The frequencies are log scaled.

Table 5.3: Mean, median and variance of the expected value predictions and labels over the test set.

Statistic	Test set	Conf. 1	Conf. 4
Mean	0.192	0.112	0.219
Median	0.0	0.00050	0.00009
Standard deviation	1.435	0.583	0.829

Table 5.3 shows the mean, median and standard deviation of the labels and predictions. Configuration 4 resembles the test distribution better across the statistical measures in Tab. 5.3. The models are biased toward lower rain rates for the rain samples and bias towards larger rain rates for the no rain samples. This can also be seen in Tab. 5.6 located further down in results. When comparing the errors, Conf. 4 has a better distribution as can be seen in Fig. 5.2. Configuration 4 has a more even distribution around 0 mm/h especially for the rain occurrences of the test set. Configuration 4 performs better with regards to MSE, bias and correlation between label and prediction. These results are found in Tab. 5.4. However, there is no improvement with regards to MAE.
When evaluating the quantile predictions displayed in Tab. 5.5 it shows that Conf. 4 again outperforms Conf. 1. This shows that when considering MAE and MSE, the expected value prediction is not the best single value predictor for Conf. 4. The reason behind this seems to be that the 0.7 and 0.9 quantiles tend to predict far higher values as can be seen in Tab.

5.5. Notable is that Configuration 4 has a considerably higher correlation between the label and the prediction. The overall bias is close to zero for both models. Figure 5.3 shows how the bias depends on the label rain rate magnitude. Both models tend to predict lower values for the higher rain rates.
Since the dataset is highly skewed towards no rain values it is of interest to evaluate how the model performs depending on the existence of rain. Table 5.6 show statistics for this purpose. Configuration 1 outperforms Conf. 4 for no rain values and Conf. 4 performs better for rain values. One explanation for this is because the model is capable of predicting higher values and does not predict these values correct every time leading to higher errors.

5.2.2 Calibration and sharpness

The overall calibration for both configurations is good as can be seen in Fig. 5.4. The CNN, Conf. 4, has slightly worse overall calibration for the middle quantiles. The two configurations construct confidence intervals in somewhat different ways. The first thing to notice is that the mean length of the confidence intervals is much larger for Conf. 4 indicating worse sharpness. The mean, median and variance for the 80 % confidence interval lengths can be found in Tab. 5.6 and the right plot of Fig. 5.4 shows the histogram of the same confidence interval lengths. Configuration 4 predicts larger confidence intervals compared to Conf. 1 when the label rain rates are bigger as can bee seen in Fig. 5.5. As it predicts these larger intervals it also has a better calibration for those values, see the right plot of Fig. 5.6. The same pattern appear when comparing the calibration against the magnitude of label rain rates, see Fig. 5.7. The majority of CI intervals lie in the interval from 0 to 0.002 mm/h, see Fig. 5.4, and in that interval, they are either underrepresented or overrepresented as can be seen in the left plot of figure 5.6.
As mentioned before, the mean length of the confidence intervals are considerably better for Conf. 1. This might be an unfair measure considering the large standard deviation of the intervals. The median ignores the outliers and in that aspect, Configuration 4 is sharper overall and for the no rain labels. It is however considerably less sharp for the no rain labels.

5.2.3 Overall scores

The loss function of the QRNN and the CRPS are scores that take into account both calibration and sharpness. As can be seen in Tab. 5.7, Conf. 4 has a lower loss but a higher CRPS than Conf. 1. When the result is split into rain and no rain measurements, Conf. 1 performs better in both loss and CRPS for the non-rain labels. For the rain labels, it performs better with regards to CRPS but the loss value is considerably higher than Conf. 4. The results are shown in Tab. 5.8 According to these results, the CRPS score seems to have a higher emphasis on sharpness and the loss function tend more towards calibration. To investigate this idea further, the correlation between the length of the 80 % CI interval and the CRPS/loss score was calculated. The result is 0.43 for the loss and 0.82 for the CRPS. These calculations were done from the results of Conf. 4. As a comparison, the correlation between the CRPS/loss and the absolute error of the expected value prediction is 0.77 for the CRPS and 0.96 for the loss. These results are understandable since the loss function has the absolute error in it and the CPRS integrand, see Eq. 4.1, ranges from 0 to 1 and the integration interval plays a larger role.

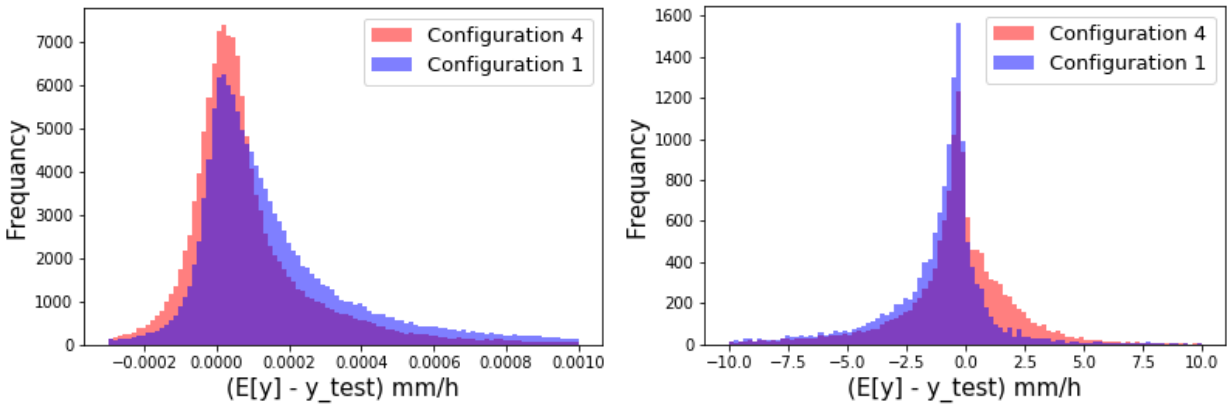


Figure 5.2: Histogram of the errors calculated as: $E[y] - y_{test}$ where y is the $E[y]$ is the expected value prediction and y_{test} is the label rain rate in mm/h. The left plot shows results for the entire test set and the right one only for the rain occurrences.

Table 5.4: Statistcal mesures for Conf. 1 and 4. The best values are indicated by bold font.

Statistic	Conf. 1	Conf. 4
MSE	1.960	**1.729**
MEDSE	0.00017	**0.00012**
MAE	**0.223**	0.249
MEDAE	3.209e-08	**1.458e-08**
Correlation target prediction	0.267	**0.428**
Bias	-0.080	**0.027**

Table 5.5: Statistcal mesures for Conf. 1 and 4. The best values are indicated by bold font.

Quantile	MSE Conf. 1	MSE Conf. 4	MAE Conf. 1	MAE Conf. 4
0.1	**2.03**	2.09	0.19	0.19
0.3	2.06	**1.89**	0.21	**0.18**
0.5	2.06	**1.70**	0.21	**0.20**
0.7	2.03	**1.92**	**0.23**	0.27
0.9	**2.26**	4.05	**0.31**	0.54

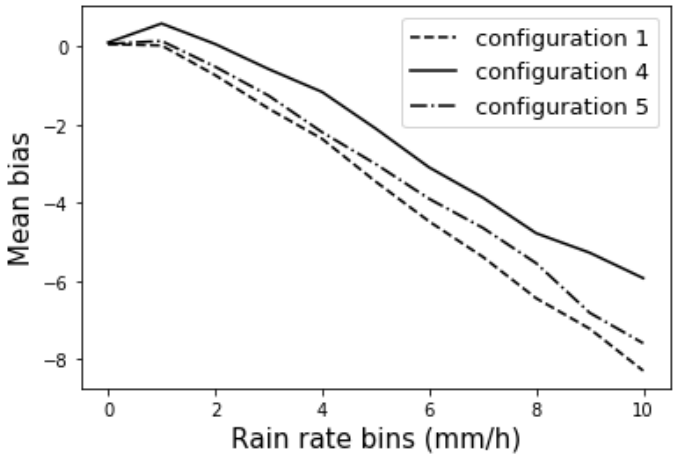


Figure 5.3: The bias as a function of rain rate magnitude.

Table 5.6: Statistics for the categories rain, no rain and overall. Configurations 1 and 4 are evaluated. 80CI is the length of the 80 % confidence interval constructed by the predictions

Conf.	MSE	MAE	Bias	80CI mean	80CI median	80CI standard deviation
1 overall	1.960	0.223	-0.080	0.2487	0.0012	0.924
4 overall	1.729	0.249	0.027	0.6014	0.0010	2.092
1 no rain	0.165	0.057	0.057	0.14	0.001	0.64
4 no rain	0.237	0.097	0.097	0.28	0.0009	1.24
1 rain	22.693	2.141	-1.667	1.54	0.69	2.00
4 rain	18.965	2.003	-0.783	4.27	2.86	4.72

5.2.4 Rain/no rain predictions

As can be sen in Fig. 5.1, rain values in the interval $(0, 0.1]$ mm/h are not present in the test data. Since rain is rarely below 0.1 mm/h, a threshold is introduced at 0.01 mm/h for the prediction to be considered a rain occurrence. Fig. 5.8 shows the CSI and Conf. 4 outperforms Conf. 1. Plots for the POD and FAR can be found in Fig. A.2 and Fig. A.3 of appendix A and these also show that overall Conf 4 outperforms Conf. 1. The FAR is slightly worse for Conf. 4 for quantiles 0.7 and 0.9.

5.3 U-net and time-series

This section presents the results from the U-net, Conf. 6, and time-series, Conf. 5. Important to keep in mind is that there are 3 different datasets for Conf. 4, 5 and 6 making this comparison less reliable than the previous section. Furthermore, the target image for the U-net is interpolated to a higher resolution thus not necessarily reflecting the true precipitation over the area. This will also lead to less reliable results. These investigations are a

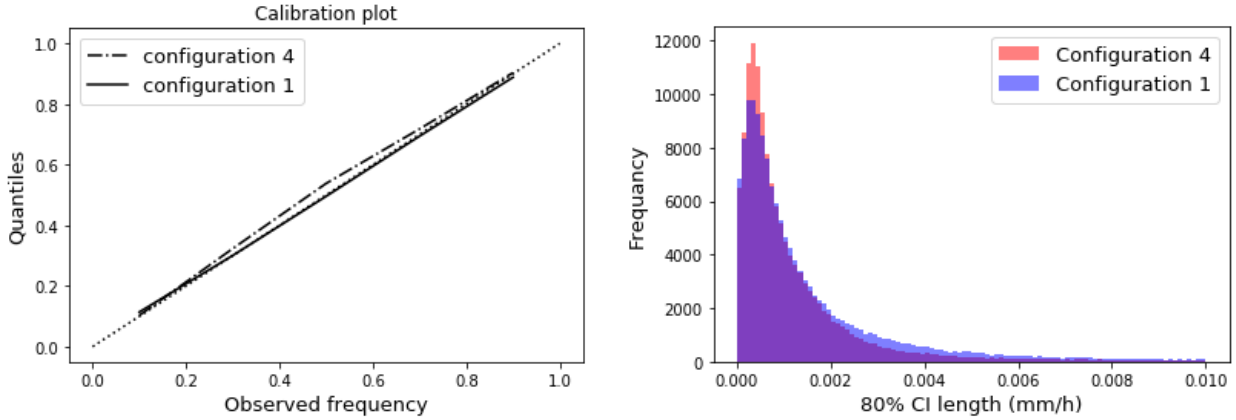


Figure 5.4: Calibration plots for Conf. 1 and 4 (left) and histogram of the 80 % CI lengths for the same configurations.

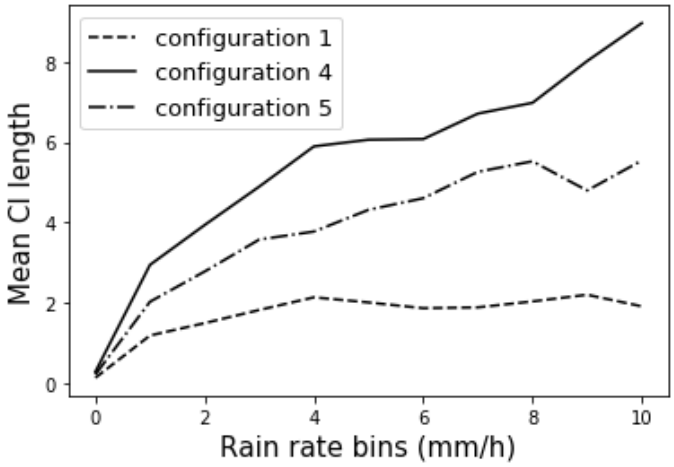


Figure 5.5: 80% CI lengths plotted against the rain rate magnitudes

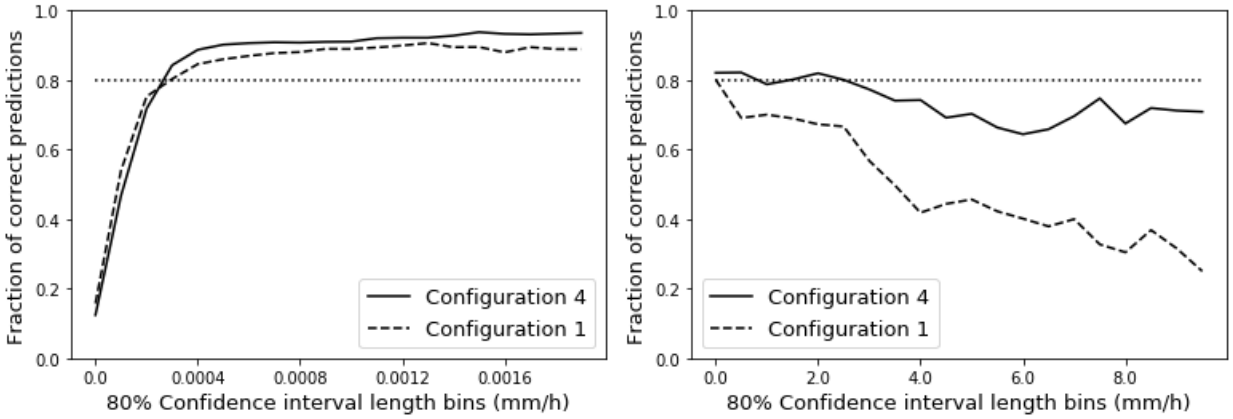


Figure 5.6: The fraction of predictions that lie within the 80 % CI plotted against the length of the same CI intervals in mm/h. The right plot covers the interval $[0, 10]$ mm/h and the left plot is an enhancement to the interval $[0, 0.002]$ mm/h. The dotted line represent the desired fraction of 0.8.

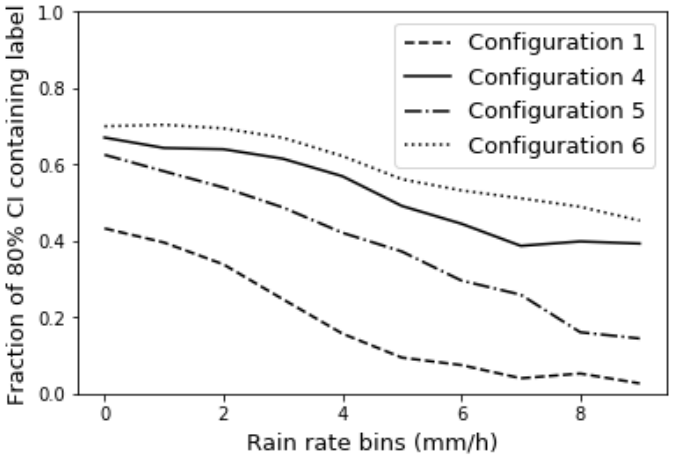


Figure 5.7: The proportions of times that the label rain rate is contained within the 80% confidence interval constructed by the prediction plotted against the magnitude of label rain rates.

smaller part of the project and did not reach significantly better results. It will, therefore, be a more brief evaluation.

The histogram of the expected value predictions can be found in Fig. 5.9 and the mean, median and standard deviation can be found in Tab. 5.9. Configuration 5 tends to predict lower rain rates than Conf. 4 and has worse mean, median and standard deviation than Conf. 4. The U-net performs slightly worse values than case 5 in that perspective. The distribution of the interpolated U-net dataset also has low rain rates that the other datasets lack. The measures related to the expected value prediction is presented in Tab. 5.10. The QMSE and QMAE values are better for the U-net and time-series configurations compared to Conf. 4. The correlation is similar for all. The sharpness is improved for Conf. 5 and worse for 6, see Tab. 5.11. Figure 5.8 shows that the CSI is similar for Conf. 4 and 6 and slightly worse for Conf. 5. Figure 5.7 shows an improvement for Conf. 6

when considering how the fraction of 80 % CI:s containing the labels as rain rate magnitude increase. Configuration 5 is slightly worse than Conf. 4 in that same perspective. Figure 5.5 also shows how the CI lengths of Conf. 5 depend in the rain rate size. It is slightly lower than Conf. 4 indicating that it is sharper for higher rain rate magnitudes but it is also less well-calibrated for those cases. The overall calibration is still good as can be seen in Fig. 5.10. The overall scores can be found in Tab. 5.12 and it shows that configuration 5 scores a better mean CRPS than Conf 4, which is understandable due to the lower mean

Table 5.7: Overall scores for Conf. 1 and 4

Score	Conf. 1	Conf. 4
Loss	0.472	0.413
Mean CRPS	0.0694	0.1014
Median CRPS	0.00024	0.00024

Table 5.8: Statistics rain and no rain measurements for Conf. 1,4, and 5

Conf.	Mean CRPS	Median CRPS	Loss
1 no rain	0.0288	0.00021	0.077
4 no rain	0.03866	0.00020	0.093
1 rain	0.5382	0.2335	5.042
4 rain	0.826	0.5098	4.100

Table 5.9: Mean, median and variance of the expected value predictions for Conf. 4, 5, 6 and the test set 4 and 5. Those corresponds to datasets 4 and 5

Statistic	Test set 4	Conf. 5	Conf. 4	Test set 3	Conf. 6
Mean	0.183	0.137	0.219	0.180	0.104
Median	0.0	0.00013	9e-5	0	4.1e-05
Standard deviation	1.324	0.494	0.829	1.22	0.407

Table 5.10: Expected value prediction measures for Conf. 4, 5 and 6

Conf.	QMSE	QMAE	CORR	Bias
4	0.839	0.713	0.428	0.027
5	0.830	0.600	0.417	-0.046
6	0.824	0.550	0.437	-0.077

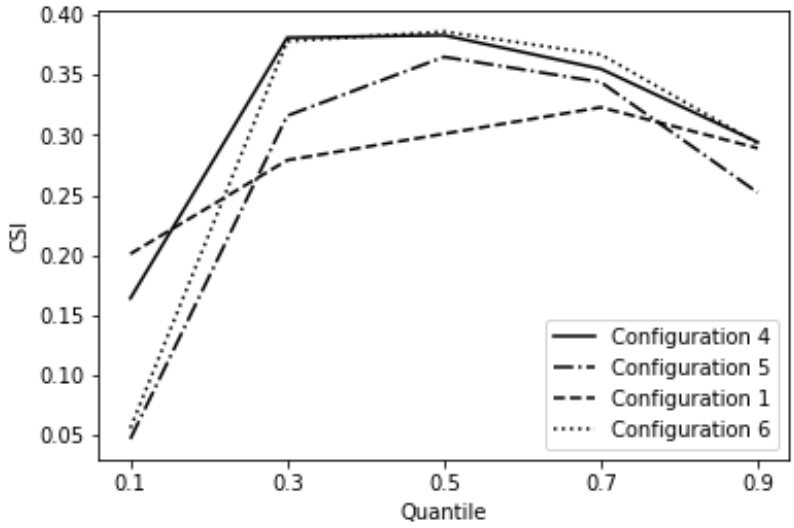


Figure 5.8: CSI for the different quantiles in Conf. 1,4,5 and 6.

Table 5.11: Mean, variance and median for the 80% confidence intervals lengths for Conf. 4,5 and 6. The unit is mm/h.

Conf.	Mean	Median	Variance
4	0.6014	0.0010	4.3776
5	0.4344	0.0014	2.0620
6	0.7391	0.0003	4.5820

confidence interval sizes. Configuration 6 scores similar to 4 in CRPS.

5.4 Examples of prediction images

Figure 5.11 shows an example of a predictions from Conf. 4 and Conf. 1 over parts of a single GPMCO pass. Figure 5.12 shows an example of a U-net prediction with its corresponding rain rate image. More examples are found in the appendix A Fig. A.4 and Fig A.5.

5.5 Comparison with the hydroestimator

This section compares the results of the CNN model, Conf. 4, and MLP model, Conf. 1, to an adapted version of a Hydroestimator(HE) [22]. The HE is described in the data section. Two different comparisons are being made. First one compares how they perform in relation to GPM data and the second one does the same in relation to gauge data. The gauge data is hourly rainfall. The results are found by selecting the closest geostationary measurements and averaging the measurements during that hour. Table 5.13 shows the MSE and correlation for the comparison against hourly gauge data collected in two days starting from 20th of march 2019. The QRNN models performs better for all scores in

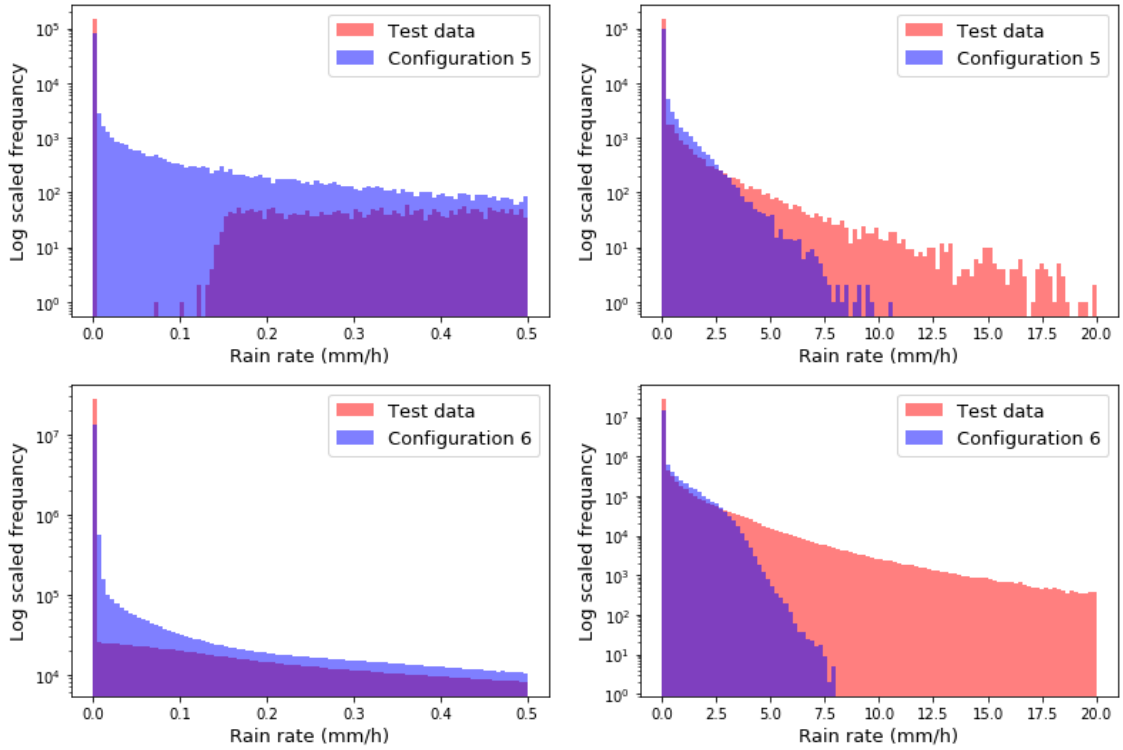


Figure 5.9: Histograms of the expected value predictions, blue, and test set, red of Conf. 5 and 6. The left plot is an enlargement over the interval $(0, 0.5]$ mm/h. The frequencies are log scaled.

Tab.5.13. Conf. 1 has the best values for MSE, MAE and bias but not for correlation where Conf. 4 is better. Figure 5.13 shows the scatter plot of the gauge values and the predicted values for Conf. 4 and HE. The comparison to the GPM data is found in Tab. 5.14 and the corresponding scatter plot in Fig. 5.14 When comparing against the GPM data, all models shows better results than the gauge comparison. The QRNN models are again showing better results than HE. Conf. 4 and Conf. 1 shows similar values expect for correlation where Conf. 4 again performs better.

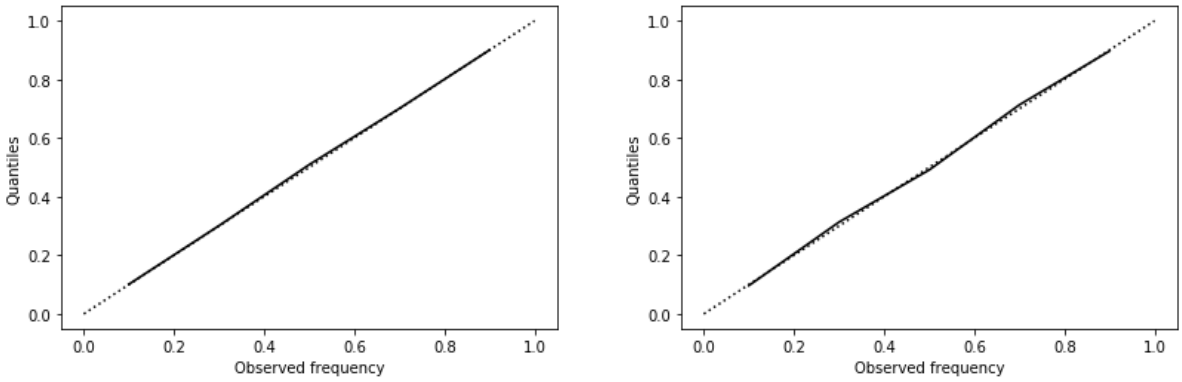


Figure 5.10: Calibration plots for Conf. 5(left) and 6(right)

Table 5.12: Overall scores for Conf. 4, 5 and 6

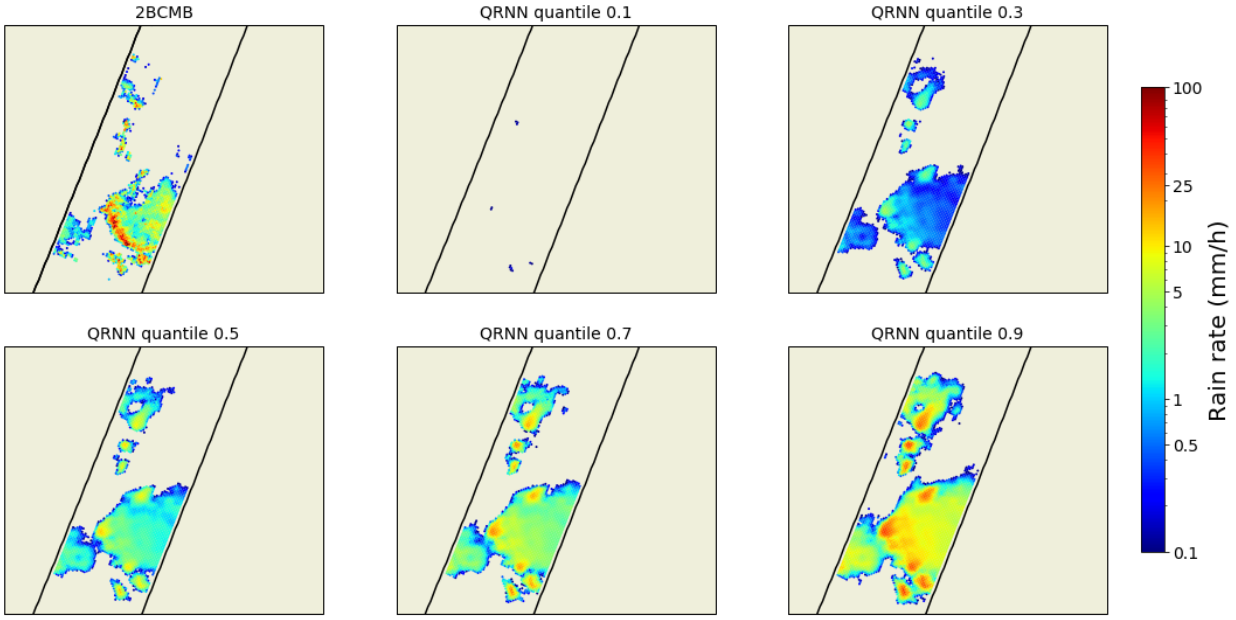
Score	Conf. 4	Conf. 5	Conf. 6
Loss	0.413	0.371	0.389
Mean CRPS	0.1014	0.787	0.107
Median CRPS	0.00024	0.00034	0.00018

Table 5.13: Gauge data results of QRNN Conf.4 and Conf.1 compared with the Hydroestimator

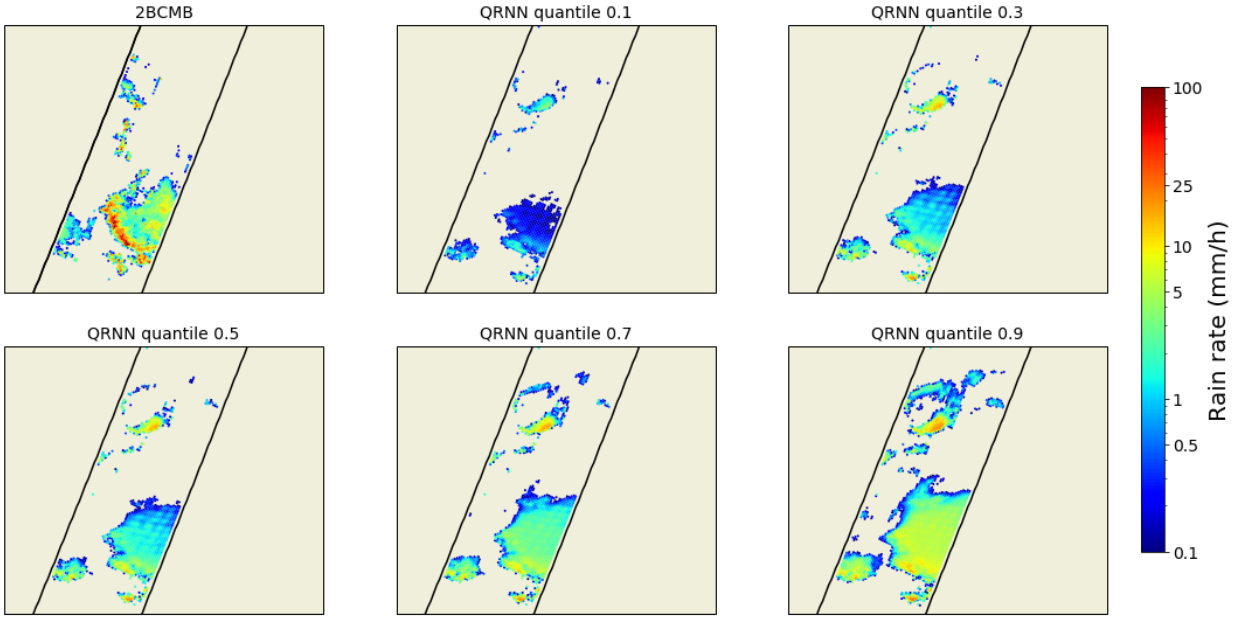
Mesurment	Conf.4	Conf. 1	HE
MSE	3.23	2.71	12.81
MAE	0.600	0.387	1.072
Bias	0.266	- 0.037	0.683
Corr	0.272	0.209	0.141

Table 5.14: GPM data results for QRNN and hydroestimator

Mesurment	Conf.4	Conf. 1	HE
MSE	1.48	1.54	4.27
MAE	0.198	0.185	0.382
Bias	-0.006	0.050	0.266
Corr	0.289	0.136	0.1068



(a) Conf. 4



(b) Conf. 1

Figure 5.11: Predictions for Conf. 1 and 4 on of a single GPM pass

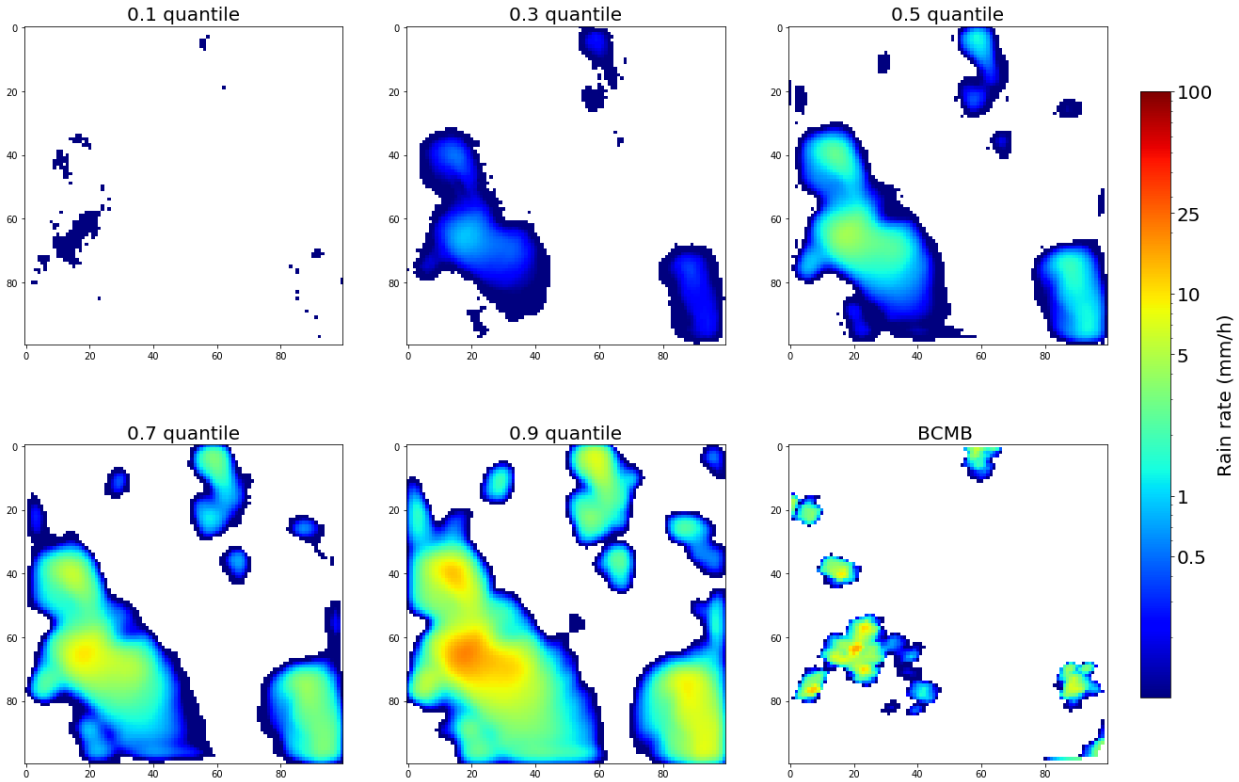


Figure 5.12: Example of prediction from the U-net with the corresponding rain rate image

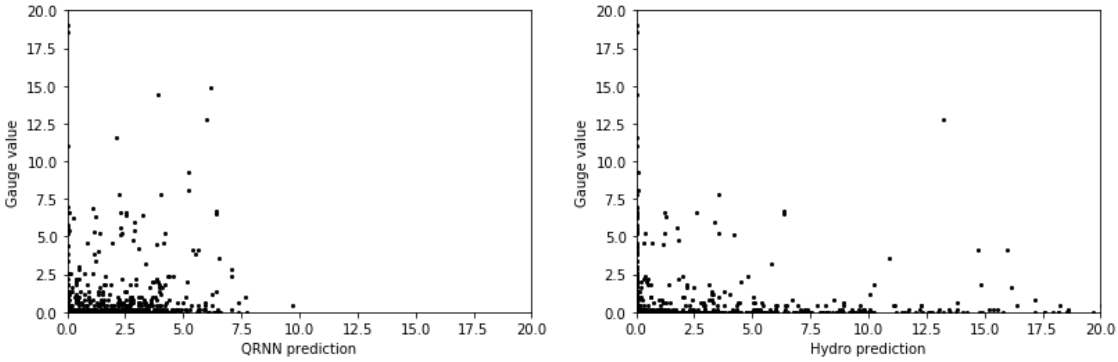


Figure 5.13: Scatter plot of QRNN Conf. 4 predictions(left) and HE predictions(right) versus the gauge data.

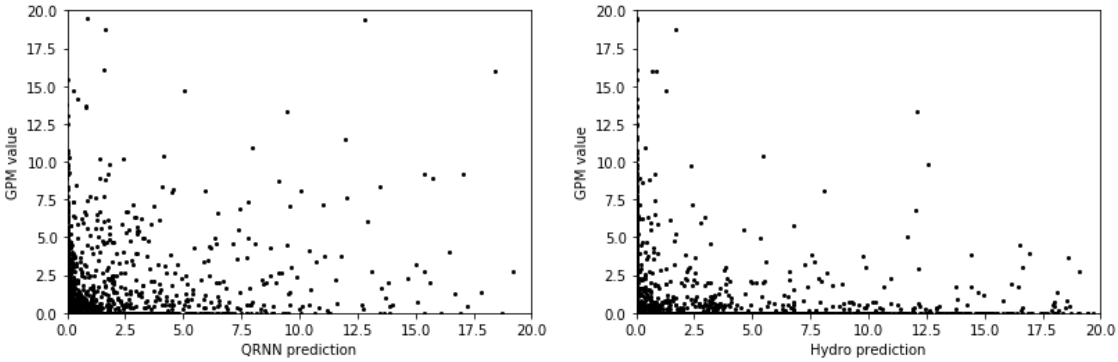


Figure 5.14: Scatter plot of QRNN Conf. 4 predictions (left) and HE predictions (right) versus the GPM data.

6
Conclusion

This project has shown some promising results for applying the QRNN on geostationary infrared data for precipitation retrievals. The networks are well-calibrated over the test data as a whole. Moreover, the convolution network has proven to be considerably better with regards to loss in the QRNN framework. The U-net shows comparable results to the regular CNN and this is what is expected since the U-net architecture has the main advantage of being fast in the inference but does not contain any type of structure that the regular CNN does not have and should thus not perform better.
The best performing configuration did outperform the adapted version of a Hydroestimator that is currently in use in Brazil. This test was however carried out over only two days and the result are therefore not very rigid.
As for taking the mean before or after feeding the network, it shows that taking the mean after is better. But one should keep in mind that because of the diamond shape of the reference data, these results should be better if the reference data has the same position in the whole dataset.
Furthermore, there is a possibility of error in the second dataset because of the interpolation. Since there is not a perfect match between the coordinate systems there will be errors introduced when matching data.
The time-series implementation shows no improvement and slightly worse results. To the best of the authors' knowledge, the reason is unknown. It could be a matter of the CNN not extracting information at the level required to capture the development of cloud formation. One idea is to implement skip connected as in the U-net and thus be able to better preserve spatial information.

6.1 Future work

This study is most of all a broad perspective of how the QRNN is applied to this problem. There are many areas that one could further investigate. The machine learning aspect is not thoroughly examined. Examples of improvements could be dropout, batch norm or another optimiser. Then there is also many other types of architectures that would be interesting to investigate. For example Capsule networks that might be better at handling the spatial information of the learnt features.
Another area of exploration is the window of the Geostationary data. It was set to $56\,\text{km} \times 56\,\text{km}$ throughout the study except for the U-net. This could be interesting to evaluate other window sizes.
The evaluation of what types of cloud formations is learnt by the network would also be interesting. To visualise the features represented in the convolutions would be most interesting

to get a better understanding of what type of network should work the best.
The evaluation of the QRNN framework could be further investigated. The results in this report indicate that a lower loss does not necessarily result in a model that is sharper. When comparing against the CRPS, it shows that a lower loss has a higher CRPS indicating a difference in how these scores evaluate the balance between sharpness and calibration.
Another aspect to evaluate are channels. In this study only the 6.2 µm and the 10.3 µm are used but there might be more useful information in other channels as well.
Scaling the labels could also be interesting. The large label variance is not optimal for neural network. By scaling the labels and decreasing the variance, the result might be improved.
Then there is of course the task of applying these models to other areas of the Earth.

Bibliography

[1] Global precipitation measurement. https://www.nasa.gov/mission_pages/GPM/spacecraft/index.html. Accessed: 2029-03-22.

[2] Instruments: Advanced baseline imager (abi). https://www.goes-r.gov/spacesegment/abi.html. Accessed: 2029-03-22.

[3] List of all satellites. https://www.wmo-sat.info/oscar/satellites. Accessed: 2029-03-22.

[4] Apple. Blurring an image, 2020. [Online; accessed June 4, 2020].

[5] Ali Behrangi, Kuo-lin Hsu, Bisher Imam, Soroosh Sorooshian, George J. Huffman, and Robert J. Kuligowski. Persiann-msa: A precipitation estimation method from satellite-based multispectral analysis. *Journal of Hydrometeorology*, 10(6):1414–1429, 2009.

[6] Ricardo Almeida de Siqueira and Daniel Vila. Hybrid methodology for precipitation estimation using hydro-estimator over brazil. *International Journal of Remote Sensing*, 40(11):4244–4263, 2019.

[7] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[8] Mircea Grecu, William S. Olson, Stephen Joseph Munchak, Sarah Ringerud, Liang Liao, Ziad Haddad, Bartie L. Kelley, and Steven F. McLaughlin. The gpm combined algorithm. *Journal of Atmospheric and Oceanic Technology*, 33(10):2225–2245, 2016.

[9] ROBERT HECHT-NIELSEN. Iii.3 - theory of the backpropagation neural network**based on "nonindent" by robert hecht-nielsen, which appeared in proceedings of the international joint conference on neural networks 1, 593–611, june 1989. © 1989 ieee. In Harry Wechsler, editor, *Neural Networks for Perception*, pages 65 – 93. Academic Press, 1992.

[10] Kou-lin Hsu, Xiaogang Gao, Soroosh Sorooshian, and Hoshin V. Gupta. Precipitation estimation from remotely sensed information using artificial neural networks. *Journal of Applied Meteorology*, 36(9):1176–1190, 1997.

[11] George J. Huffman, Robert F. Adler, David T. Bolvin, and Guojun Gu. Improving the global precipitation record: Gpcp version 2.1. *Geophysical Research Letters*, 36(17), 2009.

[12] T. Iguchi, R. Meneghini, J. Awaka, T. Kozu, and K. Okamoto. Rain profiling algorithm

for trmm precipitation radar data. *Advances in Space Research*, 25(5):973 – 976, 2000. Remote Sensing and Applications: Earth, Atmosphere and Oceans.

[13] Roger Koenker. *Quantile Regression*. Econometric Society Monographs. Cambridge University Press, 2005.

[14] Mathworks. How cnns work, 2020. [Online; accessed June 4, 2020].

[15] Michael Nielsen. Why are deep neural networks hard to train?, 2019. [Online; accessed June 4, 2020].

[16] NASA. Active and passive remote sensing diagram, 2020. [Online; accessed June 4, 2020].

[17] NOAA. Goes-16, 2018. [Online; accessed June 6, 2020].

[18] Simon Pfreundschuh, Patrick Eriksson, David Duncan, Bengt Rydberg, Nina Håkansson, and Anke Thoss. A neural network approach to estimating a posteriori distributions of bayesian retrieval problems. *Atmospheric Measurement Techniques*, 11:4627–4643, 08 2018.

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.

[20] Robert A. Schowengerdt. Chapter 2 - optical radiation models. In Robert A. Schowengerdt, editor, *Remote Sensing (Third Edition)*, pages 45 – XIII. Academic Press, Burlington, third edition edition, 2007.

[21] Francisco J. Tapiador, F.J. Turk, Walt Petersen, Arthur Y. Hou, Eduardo García-Ortega, Luiz A.T. Machado, Carlos F. Angelis, Paola Salio, Chris Kidd, George J. Huffman, and Manuel de Castro. Global precipitation measurement: Methods, datasets and applications. *Atmospheric Research*, 104-105:70 – 97, 2012.

[22] Gilberto A. Vicente, Roderick A. Scofield, and W. Paul Menzel. The operational goes infrared rainfall estimation technique. *Bulletin of the American Meteorological Society*, 79(9):1883–1898, 1998.

[23] William Olson. Gpm dpr and gmi combined precipitation l2b 1.5 hours 5 km v06, 2017. [Online; accessed June 6, 2020].

[24] www.nesdis.noaa.gov. Full disk image, 2017. [Online; accessed June 4, 2020].

A

Appendix 1

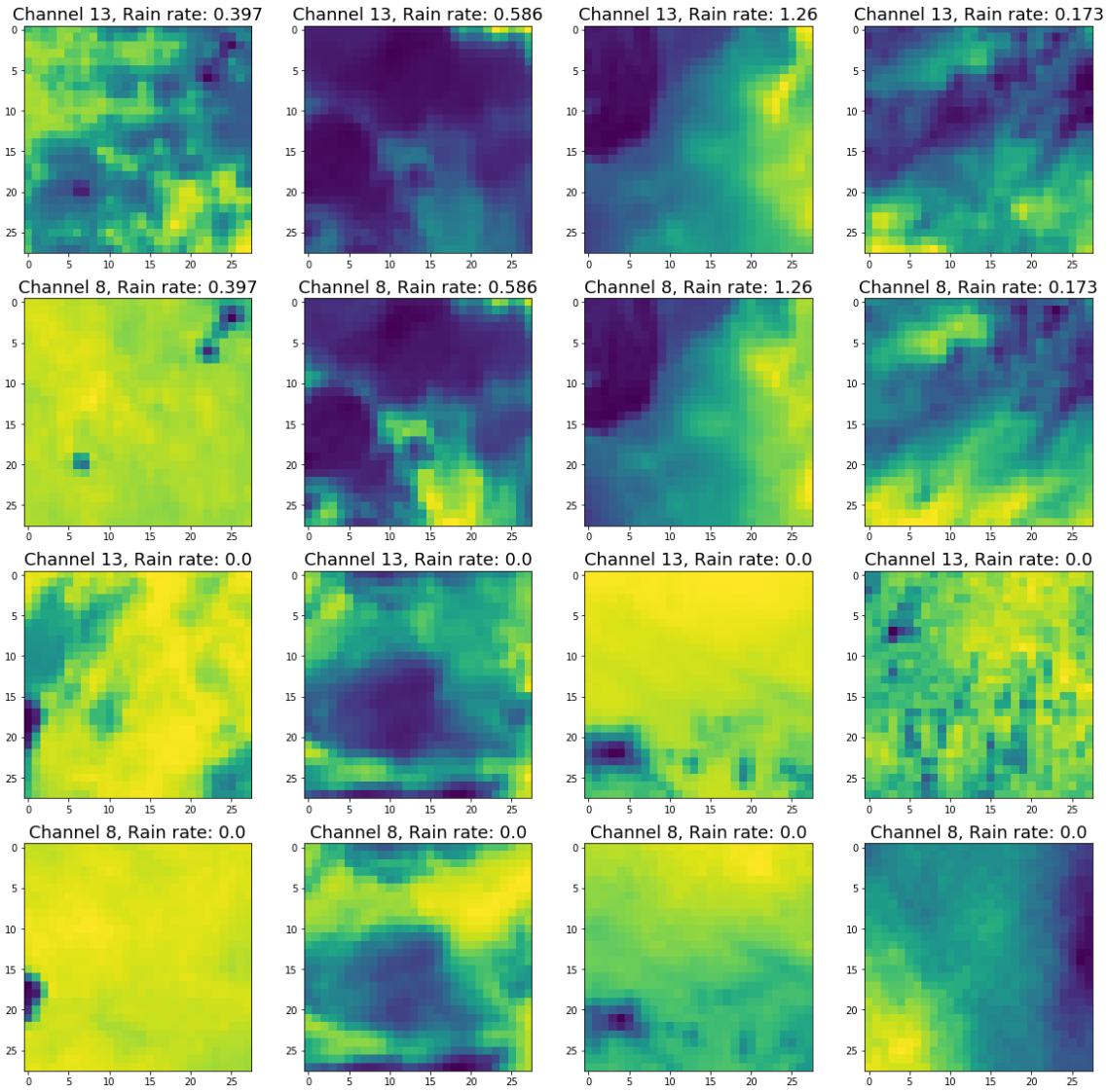


Figure A.1: Sample of the training data from dataset 1. The top two rows shows rain instances and the bottom two rows display examples of no rain. Row 1 contains channel 1 and row 2 the corresponding image for channel 8. The same pattern is displayed for the bottom two rows.

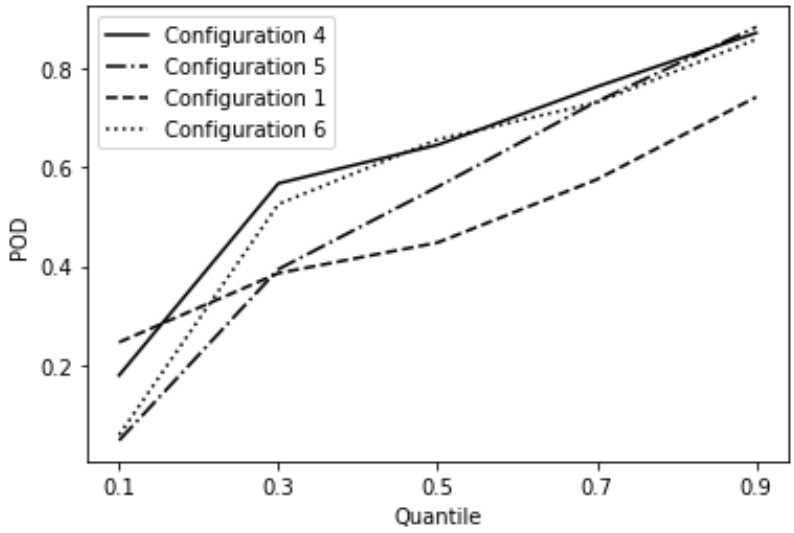


Figure A.2: POD for the different quantiles in Conf. 1,4,5 and 6.

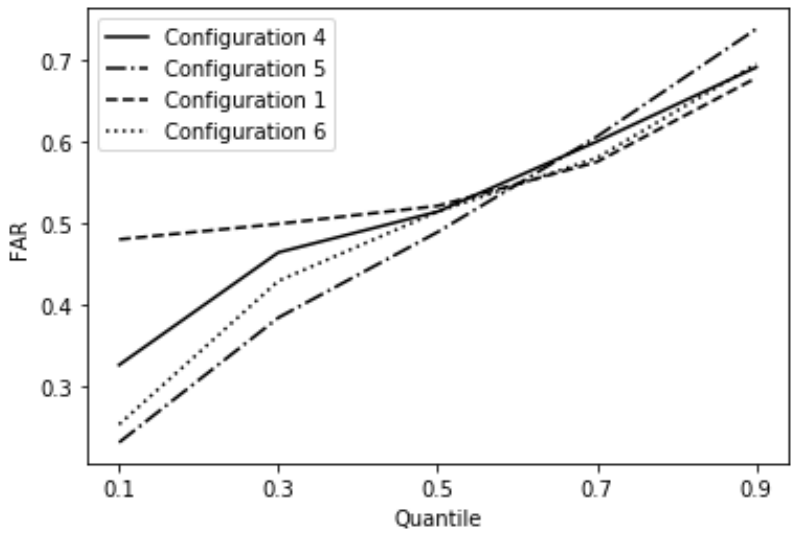


Figure A.3: Far for the different quantiles in Conf. 1,4,5 and 6.

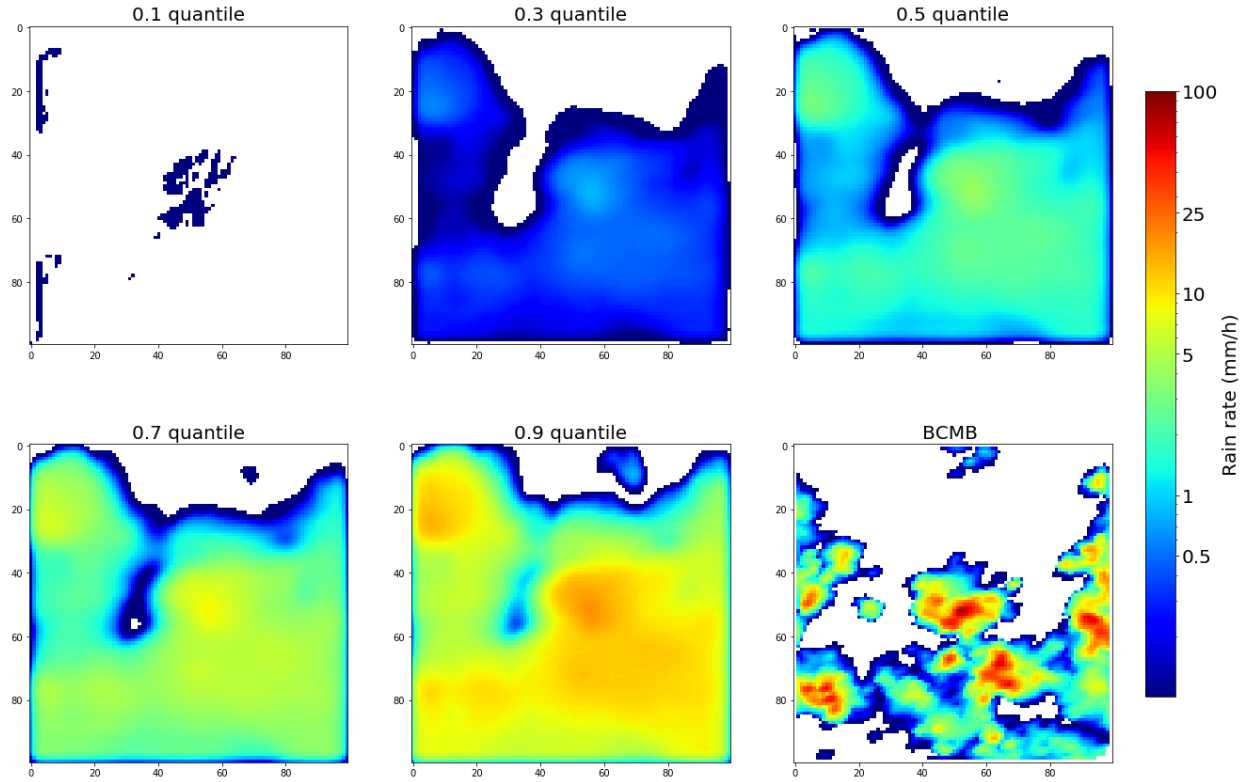


Figure A.4: Example prediction from the U-net

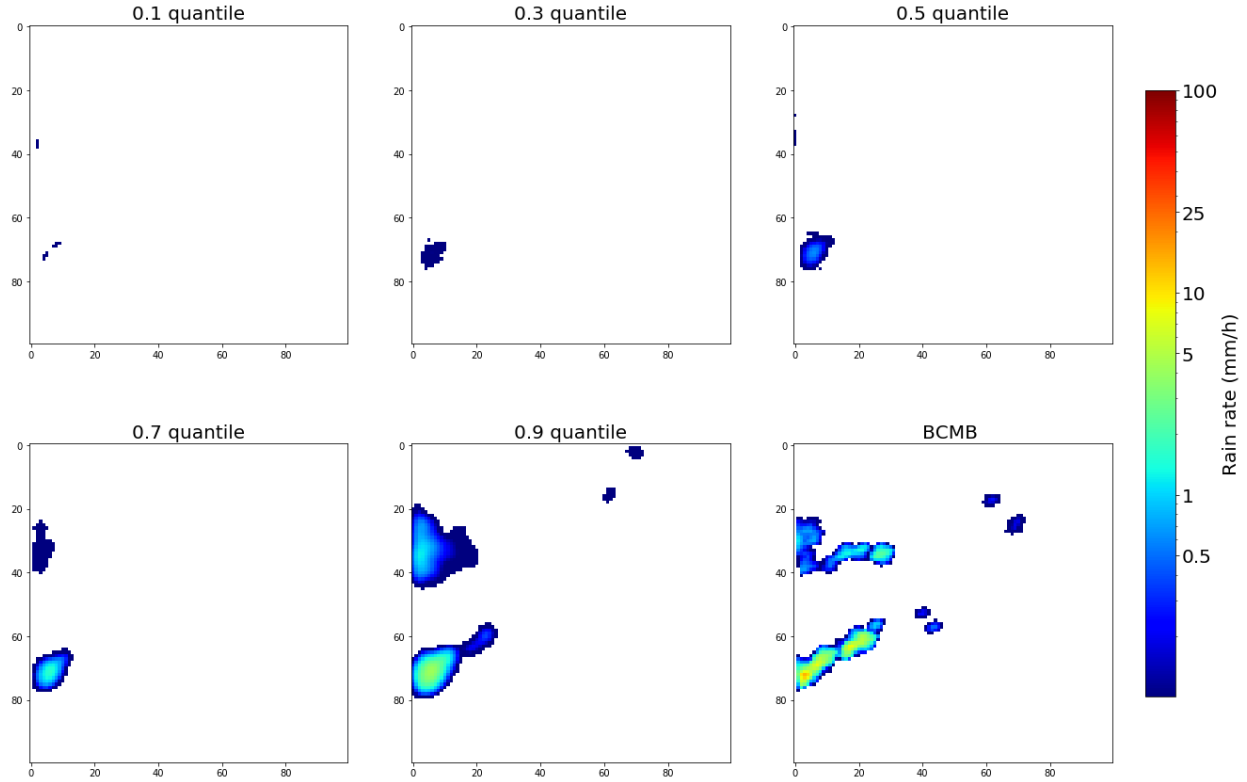


Figure A.5: Example prediction from the U-net