

Satellite to Radar: Sequence to Sequence Learning for precipitation nowcasting

MARK BRUDERER, University of Twente, The Netherlands



Fig. 1. A supercell thunderstorm at twilight in SW Oklahoma.

ABSTRACT

The forecasting of rain is a complex problem with centuries of scientific work. The implications of weather for individuals and companies continue to be important. Machine Learning approaches have been shown to outperform state of the art physics based models of weather for short term predictions. We introduce a new type of model **Sat2Rad**. Our model takes as input multi-spectral satellite data and outputs radar reflectivity at a set of time-steps ranging from 15 to 180 minutes in the future. This model is novel for precipitation nowcasting because it uses several satellite spectral bands instead of using radar data as input.

Additional Key Words and Phrases: Machine Learning, Sequence to Sequence, Radar, Satellite, Storms, Forecasting

1 INTRODUCTION

Precipitation forecasting is essential to reduce the risk of life threatening situations. Different types of rainfall ranging from mist to heavy rain have a major impact for different societal sectors including agriculture, aviation, outdoor events, and the energy industry. By having timely and accurate predictions of rainfall which in turn indicate the potential for destructive storms we can prevent injuries, assist companies in predicting energy production and use resources efficiently.

A particularly strong threat is posed by rain storms and thunderstorms. Storms are one of the most destructive weather events in nature, capable of destroying human structures and even lead to loss of life [9]. Predicting storms is crucial and presents it's own set of challenges.

At present meteorologists are able to successfully predict many instances of precipitation. Techniques that are used in practice range from manual analysis of current weather data (e.g radar or satellite images) to complex physics based simulations of our atmosphere with Numerical Weather Prediction (NWP) models. Various traditional precipitation nowcasting methods are based on *optical flow*. Optical flow functions in two steps, first cumulonimbus clouds are identified, and then their movement is tracked to predict the location of precipitation. Thus in this the case, the *cell-lifecycle* [16] is not taken into account [11].

Machine Learning (ML) approaches have also been developed to predict precipitation. An improvement of machine learning models over NWP models is that they are much faster to produce predictions, thus ML models are more suitable for real-time or near-real-time predictions, such as required in disaster response and energy management. According to the universal approximation theorem [6], deep neural networks have the property of being able to approximate any function provided they have the correct weights, thus it is suggested that machine learning models can incorporate sources of predictability beyond optical flow such as the cell-lifecycle among others [11].

Thus far most machine learning approaches for precipitation nowcasting have focused on predicting a next frame in a time series of radar reflectivity data [2, 17, 18]. However taking this approach may eliminate the possibility of learning the cell-lifecycle, due to the fact that the model only sees precipitation itself but not the cloud that is causing the precipitation.

We propose to use multi-spectral satellite data to learn spatio-temporal mappings between sequences of satellite data and precipitation data in the near future. If this is successful *cumulonimbus* clouds could be predicted from when they are mere *cumulus* clouds

TScIT 39, July 8, 2023, Enschede, The Netherlands

© 2023 Copyright held by the owner/author(s).

This is the author's version of the work. It is posted here for your personal use. Not for redistribution. The definitive Version of Record was published in , <https://doi.org/10.1145/nnnnnnnn.nnnnnnnn>.

and prior. An additional advantage is that contrary to radar data, satellite data is readily available over oceans and remote communities which allows for the prediction of precipitation over these regions. We will work towards creating a model by answering the following main research question:

1.1 Problem Formulation

We approach precipitation nowcasting as a self-supervised problem. A self supervised task arises when no explicit labels are given, but instead labels can be taken from the raw data itself as is commonly the case with time-series data. We can therefore apply known techniques in supervised learning to solve our research problem.

Consider a dataset $\{X, Y\}$ consisting of pairs of input-output sequences indexed by $i \in \mathbb{N}$, Let

$$X = \{\forall i | x^{(i)} \in \mathbb{R}^{t \times H \times W \times c}\}$$

where $x^{(i)}$ is a tensor of dimension $t \times H \times W \times c$ representing the sequence of satellite images at position i , having t timesteps, H height, W width and c channels. The set of output sequences is denoted as

$$Y = \{\forall i | y^{(i)} \in \mathbb{N}^{W \times H}\}$$

where $y^{(i)}$ represents the i_{th} $H \times W$ dimensional tensor containing discrete integers mapping to rainfall intensity classes.

The problem is formulated as finding a probability mass function $p(x)$. This function must minimize the Cross Entropy Loss expressed as follows:

$$E = \frac{1}{H + W} \sum_{i=1}^H \sum_{j=1}^W t_{ij} \log(p_{ij})$$

Let $\hat{Y} = \{\forall i | p(x^{(i)})\}$, representing the predicted outputs for each sequence of satellite images and identical in dimensions to $y^{(i)}$. find $p(x)$ such that $E(\hat{Y}, Y)$ is minimized.

2 MAIN CONTRIBUTION

This proposal has the potential to contribute a deep learning model capable of predicting precipitation in the short term with only satellite images.

3 RELATED WORKS

In this section we will discuss the existing work in precipitation nowcasting via machine learning. This section is structured by the type of input data used in the approaches, first we list radar based learning and then we discuss satellite based approaches.

3.1 Radar Based Nowcasting

Recurrent neural networks (RNNs) have been created to learn temporal relationships in data, therefore they are a natural candidate to the task of learning spatio-temporal patterns of weather. The LSTM architecture was developed by Hochreiter and Schmidhuber [8], to solve the problem of vanishing and exploding gradients in RNNs and is widely used. Taking LSTM as a base and adapting the weights to kernels, ConvLSTM [17] was introduced for the task of precipitation nowcasting. Multiple layers of ConvLSTM are used in

this paper to obtain a sequence to sequence architecture. A further improvement of ConvLSTM is TRAJGRU which was proposed by Shi et al. [18] to be able to learn the *location-variant* structure for recurrent connections.

Pure convolutional neural networks have also been used to predict precipitation. As demonstrated by Bai et al. and Gering et al. [3, 7] convolutional neural architectures can outperform recurrent neural networks for a variety of sequence modelling tasks. This is the reason why many works on precipitation nowcasting have opted for pure convolutional networks [1, 2].

Due to machine learning models attempting to minimize loss, *blurry predictions* can be produced by models. This can be alleviated by using generative models which sample from the possible futures and do not seek to provide a best average fit. Generative Adversarial Networks have been successfully applied to the task of precipitation nowcasting [12].

3.2 Satellite Based Nowcasting

In their study Chen et al. built a [5] MLP to forecast radar data from satellite data. The researchers used a combination of low earth orbit satellite passive microwave and infrared channels from two different satellites. Their model is developed to predict up to 1.5 hours in the future by recursive predictions of the model.

A study that does not predict precipitation but uses lightning as a marker for extreme precipitation was performed by Brodehl et al. [4] this study uses a convolutional network to predict lightning events, and contributes the important observation that both the visual and infrared channels are significant in differing ways to predict lightning.

The approach taken by the researchers of MetNet [19] is to combine a convolutional block for spatial downsampling, then a ConvLSTM block for temporal encoding and finally a Axial attention block [20]. MetNet is able to perform more accurate forecasts than NWP models for up to 8 hours. In this study the input data that is used is both satellite and radar data as well as the elevation, time of the year and latitude and longitude values.

4 BACKGROUND

In order to understand this research it is important to be acquainted with the machine learning techniques that will be used. Furthermore insight into this study can be enhanced by having more in depth information on the used data-sets and the meteorological theory.

For the machine learning aspect of this research it is important to understand multi layer perceptrons (MLPs) [15]. Furthermore it is important to understand convolutional neural networks and what benefits they afford in relation to MLPs [10]. It is also crucial to have an understanding of recurrent neural networks (RNNs) and their extension to include convolutions [17].

4.1 Convolutional Neural Networks

Convolutional Neural Networks (CNNs) or Fully Convolutional Networks (FCNs) are a class of deep learning models suited for computer vision tasks. Unlike fully connected neural networks, which

treat input data as a one dimensional vector, CNNs are designed to process higher dimensional data such as images. This distinction enables CNNs to exploit more spatial relationships and patterns in visual data as opposed to a flattened vector where these patterns are not recoverable.

CNNs were first introduced by [1] in 1987, but were shown to beat state of the art classification with the ImageNet paper.

This ability to automatically extract meaningful features from images is a fundamental advantage of CNNs over fully connected networks, as it significantly reduces the number of parameters and alleviates the computational burden associated with processing high-dimensional inputs. Furthermore, CNNs leverage parameter sharing, enabling them to efficiently learn translation-invariant features across the entire image. These inherent advantages make CNNs particularly effective for computer vision tasks, such as image classification, object detection, and semantic segmentation. In CNNs, the core building blocks are convolutional layers, pooling layers, and fully connected layers. A typical CNN architecture consists of multiple convolutional layers, each followed by a non-linear activation function, such as ReLU (Rectified Linear Unit), which introduces non-linearity into the network. Convolutional layers perform the convolution operation by applying a set of learnable filters to the input image, producing feature maps that capture local patterns. The pooling layers, commonly implemented as max pooling or average pooling, downsample the feature maps, reducing their spatial dimensions while preserving the most salient information. This downsampling helps achieve translation invariance and improves computational efficiency. The final layers of a CNN usually consist of one or more fully connected layers, which aggregate the learned features and make predictions based on the extracted representations. The parameters of the CNN are optimized through backpropagation and gradient descent algorithms, enabling the network to learn discriminative features that are highly informative for the given task. The modular architecture and hierarchical nature of CNNs make them well-suited for extracting spatial features from images and effectively capturing the complex patterns present in computer vision tasks. Their ability to automatically learn relevant features from raw pixel data, combined with their parameter sharing and hierarchical structure, has revolutionized the field of computer vision and significantly advanced the state-of-the-art in various visual recognition tasks.

4.2 Unet

Semantic segmentation, the task of assigning a class label to each pixel in an image, plays a crucial role in computer vision applications such as image understanding and scene understanding. U-Net, a convolutional neural network (CNN) architecture, has emerged as a highly effective solution for semantic segmentation tasks, providing accurate and detailed pixel-level predictions. The U-Net architecture is specifically designed to address the challenge of limited training data and to capture both local and global context information. The U-Net architecture gets its name from its U-shaped design, which consists of an encoder path and a decoder path. The encoder path resembles a traditional CNN and serves to capture spatial information and learn feature representations at various scales. It typically

comprises multiple convolutional and pooling layers, where each convolutional layer extracts increasingly abstract features by convolving with learnable filters and applying non-linear activation functions. The decoder path, on the other hand, aims to recover the spatial information lost during the pooling and downsampling operations of the encoder. It employs a series of upsampling and transposed convolutional layers to gradually increase the spatial resolution and reconstruct the detailed predictions. The skip connections between the corresponding encoder and decoder layers help preserve fine-grained details and enable the fusion of local and global context information, facilitating more accurate segmentation. These skip connections bridge the gap between low-level and high-level features, allowing the network to leverage both local and global context for precise pixel-wise predictions. One of the key advantages of U-Net is its ability to capture contextual information effectively. By using skip connections, the network can combine low-level and high-level features, enabling the model to refine predictions by incorporating local details and global context simultaneously. This property is especially beneficial for tasks like semantic segmentation, where precise boundary delineation and accurate classification of object categories are essential. Furthermore, U-Net has gained popularity due to its relatively small number of parameters compared to other fully convolutional networks, making it more computationally efficient while maintaining competitive performance. This aspect is particularly advantageous when dealing with limited training data or when deploying models on resource-constrained devices. In summary, U-Net has emerged as a powerful CNN architecture for semantic segmentation tasks. By leveraging its U-shaped design, skip connections, and an effective combination of local and global context information, U-Net achieves accurate and detailed pixel-wise predictions. Its ability to capture contextual information while being computationally efficient makes it an attractive choice for various computer vision applications requiring fine-grained segmentation.

4.3 Convolutional Neural Networks

Why do they exist motivation for conv nets comes from losing info when flattening but also when processing images it is possible for a pixel to receive info from all others becomes very expensive t^2 but when using conv nets its the kernel that is being learned. image transformation kernels manually developed. Combining the identically with ANN we can learn the weights of the kernels, for our given problem.

How do they work they work

4.4 Recurrent Neural Networks

4.5 Convolutional LSTM

4.6 Axial Self-Attention

To have an understanding of the meteorological context of this project it is recommended to have an understanding of radar data [13], satellite data specifically the kind used in this project [14].

RQ1: *How can a deep learning model be trained to predict radar data with multi-spectral satellite data ?*

This research question will be answered by looking at the following sub research questions:

- (1) *what is necessary to build a well performing model ?*
- (2) *what methods can be used to create this model ?*
- (3) *what is the effectiveness of the trained model ?*
- (4) *what are the conditions under which the model can and cannot be used ?*

5 METHODOLOGY

We choose to structure our project based on the CRISP-DM framework.

5.1 Business and Data Understanding

From a business perspective the data.

5.2 Data Preprocessing

For the preprocessing of data we created a pipeline which distinguishes between satellite images and radar images to preprocess each following their own needs (See figure 2).

In the case of satellite data, we first download this data contained in zip files from a storage bucket. After all files are downloaded they are extracted into zip files, the pipeline takes care of deleting any files which are not needed anymore. Then we reproject the satellite images using the satpy package. By reprojecting we reduce the dimensions of each satellite image to 256 x 256 pixels. Via the reprojection we also obtain only the area of interest, specified by the coordinates for the lower corner (0, 50) and upper corner (1) of the region, this gives us the benelux region. This is the same region as given in the radar data. Additionally the projection is matched between the satellite image and radar image. After reprojecting we perform a *statistics* step where we aggregate the dataset by finding the minimum and maximum values for each channel. The statistics are necessary for the next step which is normalization. During the normalization step we perform the *Min-Max* Normalization (equation 1).

$$x_{normalized} = \frac{x - min}{max - min} \quad (1)$$

After finalizing the normalization step we sample an image from the dataset which is visualized to check for errors in the pipeline.

The radar pipeline begins with the downloading of radar files in the form of h5 files. The collected files are each converted to decibels relative to Z (dBZ) from their previous unit.

$$dBZ = x \cdot 0.5 - 32 \quad (2)$$

The preprocessing pipeline then splits into two, one step will normalize values between 0 and 1 using *Min-Max* normalization and the other step will use levels of *dBZ* to create discrete ranges of precipitation to use as classes during training. Finally the current images in the pipeline are resized using the nearest neighbor algorithm, to avoid changing the values with bilinear interpolation. Next identically to the satellite pipeline we sample and visualize a radar image for verification purposes.

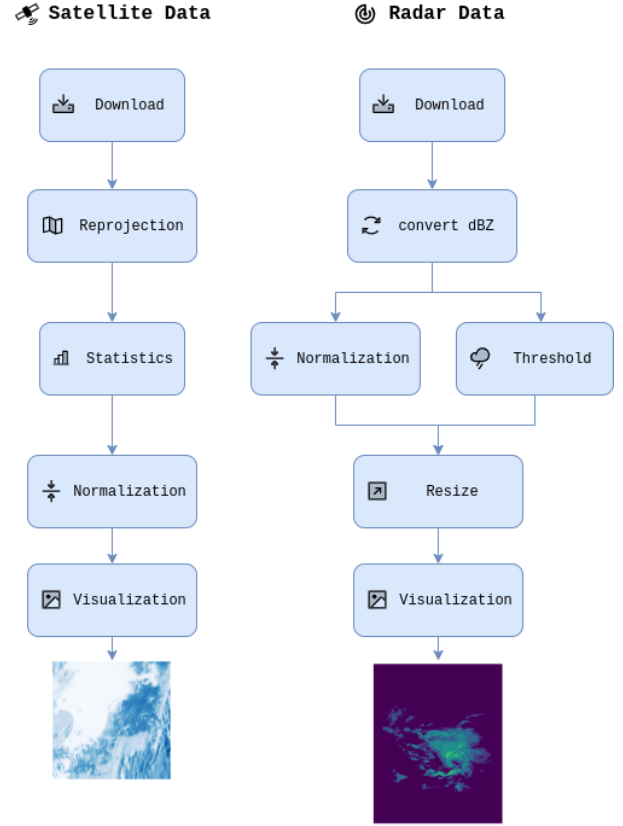


Fig. 2. Preprocessing Pipeline

5.3 Training

In order to build and train our models we worked with pytorch. We used pytorch lightning to provide a higher level interface to speed up the research process. To track our experiments we connected pytorch lightning to mlflow, using MLFlow we can log metrics, artifacts produced during training while keeping track of the used parameters during that training run. To create pipelines for preprocessing and training, we used the zenml library.

We created 4 different datasets for our research. These datasets result from the combination of sliding and sequential datasets with class and regression datasets. We implemented these datasets as subclasses of pytorch vision datasets. We also created time based utility functions to align the start and ending times of satellite data and radar data, such that when data is split based on training, validation and testing percentages, data is still in the same temporal range. Keeping in mind that the radar data has a different resolution and is received at a different strides.

5.4 Performance Evaluation

Different metrics were used for the regression and the classification.

Classification evaluation metrics:

- (1) Precision
- (2) Accuracy

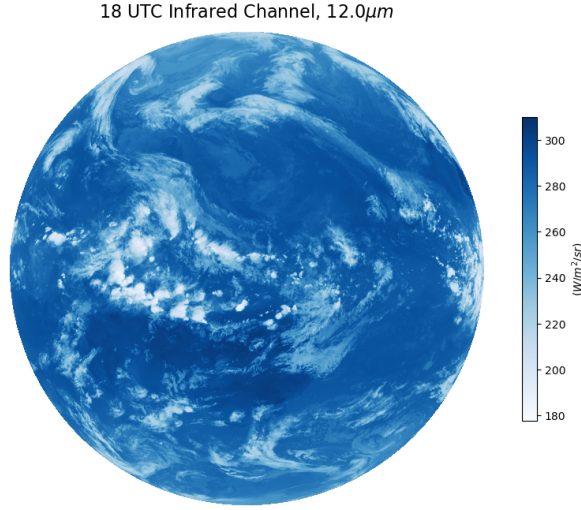


Fig. 3. Satellite Image: Infrared Channel 18UTC 12.0μm

- (3) Recall
- (4) Exact Match
- (5) F1 Score
- (6) Jaccard Index

$$MSE = \frac{\sum (\hat{y}_i - y_i)^2}{n} \quad (3)$$

$$MAE = \frac{\sum |\hat{y}_i - y_i|}{n} \quad (4)$$

$$RMSE = \sqrt{\frac{\sum (\hat{y}_i - y_i)^2}{n}} \quad (5)$$

Additional metrics come from the study [18], where the authors created B -MSE and B -MAE due to the fact that the frequencies of different rainfall levels are imbalanced. Models trained with conventional MSE or MAE are biased towards predicting low amounts of precipitation. This can be solved with a loss function that weights the higher precipitation more strongly. The weighting function w can be found in the study as well.

$$BMSE = \frac{\sum w(\hat{y}_i) \cdot (\hat{y}_i - y_i)^2}{n} \quad (6)$$

$$BMAE = \frac{\sum w(\hat{y}_i) \cdot |\hat{y}_i - y_i|}{n} \quad (7)$$

6 RESULTS

7 CONCLUSIONS

At this point in the research we have created an introduction for the research. We have also performed a literature review of available work in the field of precipitation nowcasting. We have written the methods that we will use to perform the research and listed expected results. And we have visualized some of the data from our dataset.

ACKNOWLEDGMENTS

I would like to thank my supervisor from Elena Mocanu, for her help. Furthermore I would like to thank I would like to thank Dina Lazorkeno for proofreading my thesis.

REFERENCES

- [1] Shreya Agrawal, Luke Barrington, Carla Bromberg, John Burge, Cenk Gazen, and Jason Hickey. 2019. Machine Learning for Precipitation Nowcasting from Radar Images. arXiv:1912.12132 [cs.CV]
- [2] G. Ayzel, T. Scheffer, and M. Heistermann. 2020. RainNet v1.0: a convolutional neural network for radar-based precipitation nowcasting. *Geoscientific Model Development* 13, 6 (2020), 2631–2644. <https://doi.org/10.5194/gmd-13-2631-2020>
- [3] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. 2018. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. arXiv:1803.01271 [cs.LG]
- [4] Sebastian Brodehl, Richard Müller, Elmar Schömer, Peter Spichtinger, and Michael Wand. 2022. End-to-End Prediction of Lightning Events from Geostationary Satellite Images. *Remote Sensing* 14, 15 (2022). <https://doi.org/10.3390/rs14153760>
- [5] Haonan Chen, V. Chandrasekar, Robert Cifelli, and Pingping Xie. 2019. A Machine Learning System for Precipitation Estimation Using Satellite and Ground Radar Network Observations. *IEEE Transactions on Geoscience and Remote Sensing* PP (10 2019), 1–13. <https://doi.org/10.1109/TGRS.2019.2942280>
- [6] George Cybenko. 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems* 2, 4 (12 1989), 303–314. <https://doi.org/10.1007/bf02551274>
- [7] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional Sequence to Sequence Learning. arXiv:1705.03122 [cs.CL]
- [8] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-term Memory. *Neural computation* 9 (12 1997), 1735–80. <https://doi.org/10.1162/neco.1997.9.8.1735>
- [9] NOAA National Severe Storms Laboratory. [n. d.]. Thunderstorm Basics. <https://www.nssl.noaa.gov/education/svrwx101/thunderstorms/>
- [10] Keiron O'Shea and Ryan Nash. 2015. An Introduction to Convolutional Neural Networks. arXiv:1511.08458 [cs.NE]
- [11] Rachel Prudden, Samantha Adams, Dmitry Kangin, Niall Robinson, Suman Ravuri, Shakir Mohamed, and Alberto Arribas. 2020. A review of radar-based nowcasting of precipitation and applicable machine learning techniques. arXiv:2005.04988 [physics.aos-ph]
- [12] Suman Ravuri, Karel Lenc, Matthew Willson, Dmitry Kangin, Remi Lam, Piotr Mirowski, Megan Fitzsimons, Maria Athanassiadou, Sheleem Kashem, Sam Madge, Rachel Prudden, Amol Mandhane, Aidan Clark, Andrew Brock, Karen Simonyan, Raia Hadsell, Niall Robinson, Ellen Clancy, Alberto Arribas, and Shakir Mohamed. 2021. Skilful precipitation nowcasting using deep generative models of radar. *Nature* 597, 7878 (sep 2021), 672–677. <https://doi.org/10.1038/s41586-021-03854-z>
- [13] Ronald E Rinehart. 1991. *Radar for meteorologists*. University of North Dakota, Office of the President.
- [14] J Schmid. [n. d.]. The SEVIRI Instrument. *EUMETSAT* ([n. d.]).
- [15] Juergen Schmidhuber. 2022. Annotated History of Modern AI and Deep Learning. arXiv:2212.11279 [cs.NE]
- [16] NOAA's National Weather Service. [n. d.]. NWS JetStream - Life Cycle of a Thunderstorm. <https://www.weather.gov/jetstream/life>
- [17] Xingjian SHI, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun WOO. 2015. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In *Advances in Neural Information Processing Systems*, C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (Eds.), Vol. 28. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2015/file/07563a3fe3bbe7e3ba84431ad9d055af-Paper.pdf
- [18] Xingjian Shi, Zhihan Gao, Leonard Lausen, Hao Wang, Dit-Yan Yeung, Wai kin Wong, and Wang chun Woo. 2017. Deep Learning for Precipitation Nowcasting: A Benchmark and A New Model. arXiv:1706.03458 [cs.CV]
- [19] Casper Kaae Sønderby, Lasse Espeholt, Jonathan Heek, Mostafa Dehghani, Avital Oliver, Tim Salimans, Shreya Agrawal, Jason Hickey, and Nal Kalchbrenner. 2020. MetNet: A Neural Weather Model for Precipitation Forecasting. arXiv:2003.12140 [cs.LG]
- [20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. arXiv:1706.03762 [cs.CL]

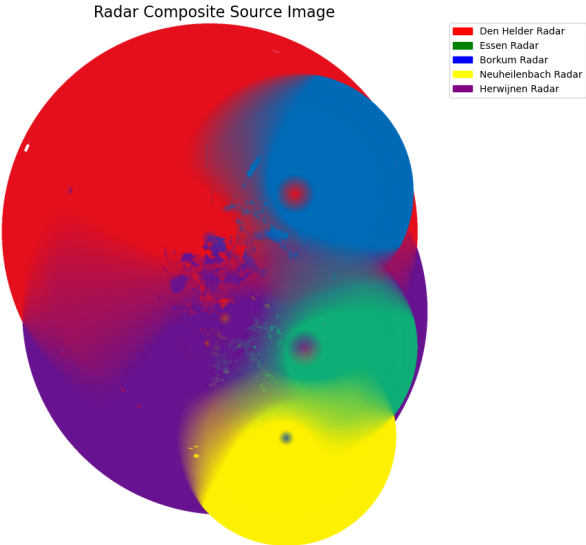


Fig. 6. Satellite Image: Infrared Channel 18UTC 12.0 μ m

Table 1. Reflectivity in dBZ versus Rainrate

LZ(dBZ)	R(mm/h)	R(in/h)	Intensity
5	(mm/h)	<0.01	Hardly noticeable
10	0.15	<0.01	Light mist
15	0.3	0.01	Mist
20	0.6	0.02	Very light
25	1.3	0.05	Light
30	2.7	0.10	Light to moderate
35	5.6	0.22	Moderate rain
40	11.53	0.45	Moderate rain
45	23.7	0.92	Moderate to heavy
50	48.6	1.90	Heavy
55	100	4	Very heavy/small hail
60	205	8	Extreme/moderate hail
65	421	16.6	Extreme/large hail

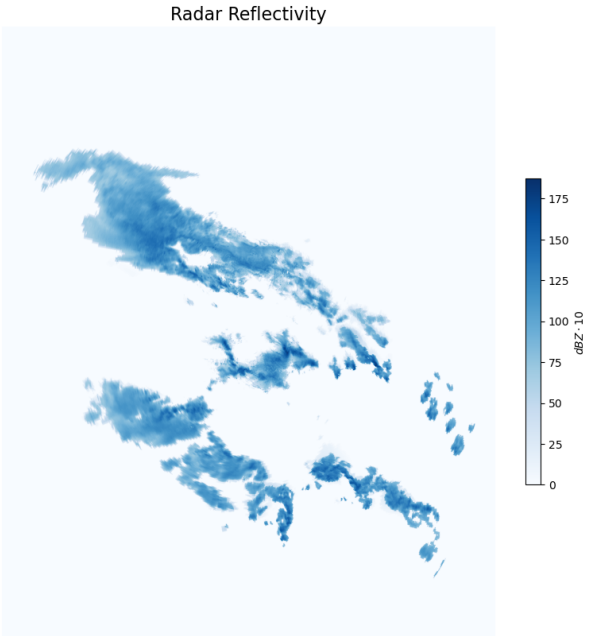


Fig. 4. Radar Reflectivity

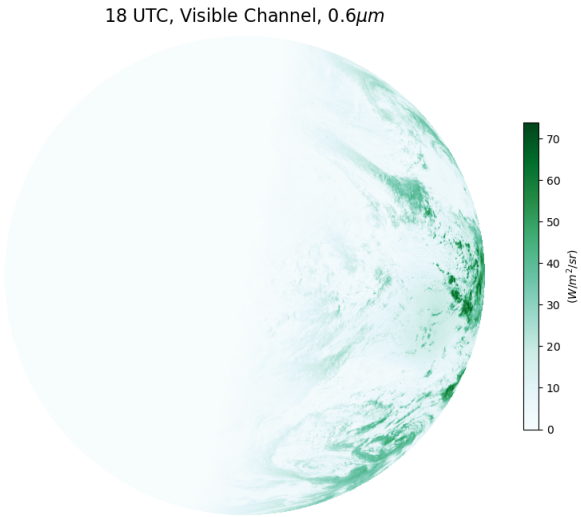


Fig. 5. Satellite Image: Visible Channel 18UTC 0.6 μ m

A APPENDIX