Submitted by
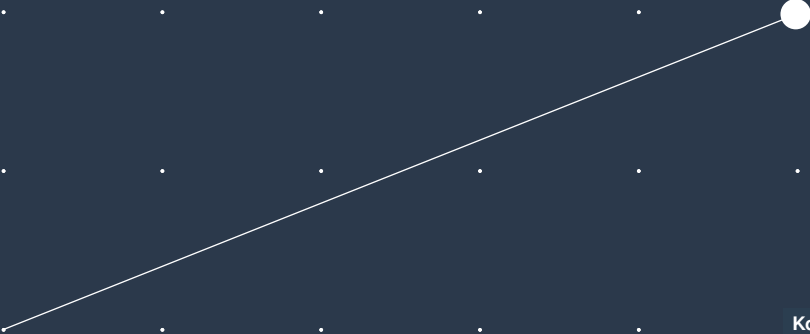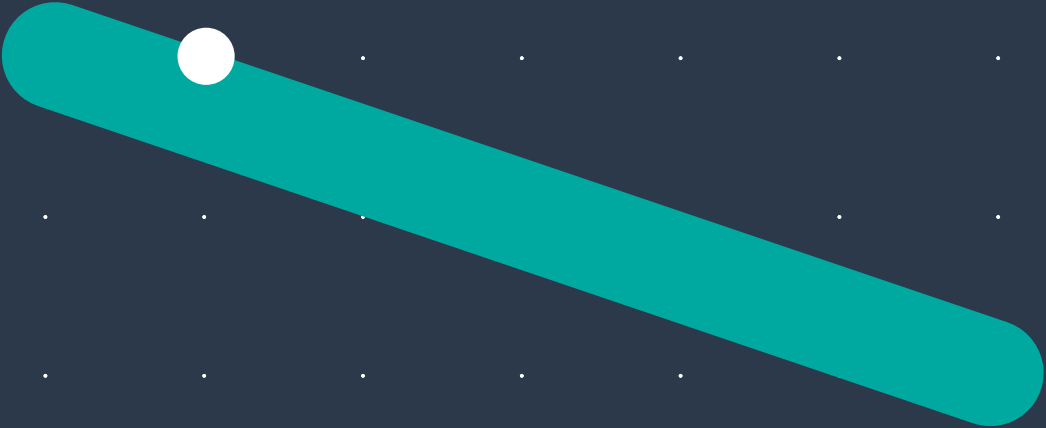Thomas Gnädig
Etienne Gramlich
Merle Wolff
Tim Hardenacke

Teamproject (Master 3. semester)

DeepRain

Extended Abstract

Topic:	DeepRain
Team members:	Thomas Gnädig, Etienne Gramlich, Merle Wolff, Tim Hardenacke
Advisor:	Hochschule für Technik, Wirtschaft und Gestaltung
	HTWG Konstanz, Institut für Optische Systeme
	Oliver Dürr

In this Paper we try to predict precipitation for a range of 35 minutes in an area around Constance. Therefore we are using machine learning techniques and train a UNet on radar data images. Here we present the result of precipitation prediction as well with regression as with classification. Both approaches provide good results. Source code and full length documentation in german can be found at GitHub: `https://github.com/thgnaedi/DeepRain`.

References

[RFB15]   Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation". In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.

Data

We are using radar Data from the Climate Data Center of the DWD[1]. Here are radarimages since 2004. This data has to be Downloaded and converted to png this is done by our Crawler[2] and Converter[3]. The rainfall images then are tailored and scaled. Unattraktive images are then rejected that only interesting images are used for the training. The images are grayscale png's. As network input five temporally consecutive of them are combined to a 3D tensor as input. The label consists of seven of them.

Architecture

To predict precipitation either with regression or classification we are using a UNet. This Architecture is a convolutional neural network that was at first developed for biomedical image segmentation. A UNet is a set of convulution layers, combined with an downsampling part, such as pooling layers, and later with upsampling parts like deconvolution layers.
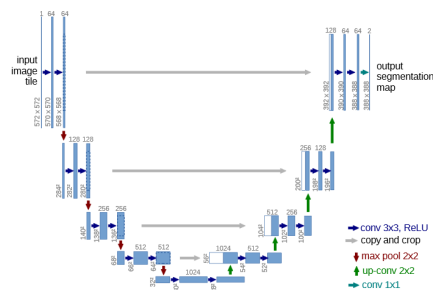

Figure 1: The image is taken from the university of Freiburg [RFB15]

The input images have a shape of 64x64 pixels. They are pooled until the shape reaches a minimum of 8x8 features. Then they are upsampled and concatenated with the downsampling part. This yields into an u-shaped architecture with crosslinks and is the reason for its name.

Regression

First approache is to predict the complete radar image. For this regression we use the MSE as lossfunction. The input layer takes the five incoming timesteps and transfers it into a 7 timestep prediction wich reaches up to 35 minutes into the future. Our network is trained on the radar data from 2005 till 2017. As validation set we use the images from 2004. further information can be found in our full project description[4]. The result of this task is shown in the following graphics.
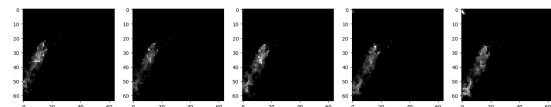

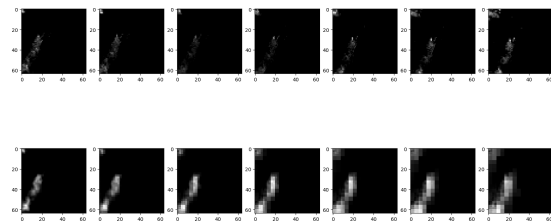Figure 2: The input is a 64x64x5 Tensor, containing the last 25 minutes, with the following content

Figure 3: The Labels are seven timesteps to predict, from 5 minutes till 35 minutes into the future. The result is shown here. In the upper row we can see the seven timesteps in the Labels. White means a lot rainfall while black is no rainfall. The second row contains the labels for this input. In the last row are the predictions of our model.

The predicted values are very close to the label for a difference of only 5 minutes to the input data. The further the time progresses, the worse the prediction becomes. In the last image it looks like the resolution of the output has been drastically decreased compared to the

[1] https://www.dwd.de/DE/klimaumwelt/cdc/cdc_node.html

[2] https://github.com/thgnaedi/DeepRain/tree/master/DWD_Crawler

[3] https://github.com/thgnaedi/DeepRain/blob/master/Data/DWDtoPngScript.py

[4] https://github.com/thgnaedi/DeepRain/blob/master/Docs/Langdokumentation.pdf

first output. In fact the resolution is the same we can say that this is an indication of uncertainty in the Network. Over time, the movement of the rainfall is well predicted. Even the tiny spot in the top left of images can be predicted. Even if the area of rainfall increases much faster than in the labels, the center is still well predicted and close to the center of the label.

Classification

The second approach is to predict the precipitation by classification. Here we try to classify into either three classes (no rain, rain, heavy rain) or two classes (no rain, rain) for each pixel. Therefore the Unets output is modified to three classes per pixel and an additional softmax layer at the output to provide propabilities. The labels are generated by a threshold for each class.
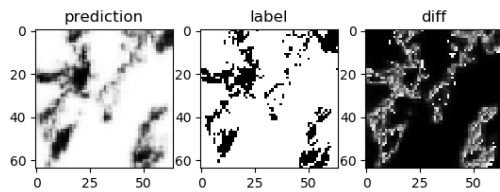

Figure 4: Classification output. In the left image the prediction of the first class (no rainfall) is plotted. The white spots are 100% no rainfall. The black spots are 0% no rainfall.

The gray spots can be rainfall or not depending on the propability of the other two classes. The right image (difference) shows us the absolute difference between the prediction and the label. Here we can see that the big black spots (no difference between prediction and label) are at the no rainfall labels. The network has a high accuracy by predicting rainfree pixels. This can be shown in the follwing ROC-image. By evaluating the confusion matrix for three class classification, you can recognize

that the last class (heavy rainfall) is higly undersampled. There are much more rainfall pixels and even more no rainfall pixels in the dataset. To increase the the performance we tried over and undersampling, but both techniques didnt work well for our scenario. Finally we changed the problem to a two clas classification with a threshold of 0, so the classes are 'no rainfall' and 'rainfall'.

The ROC-curve in figure 5 shows how well the two classes rain and no rain can be separated and predicted independent of the number of elements per class.
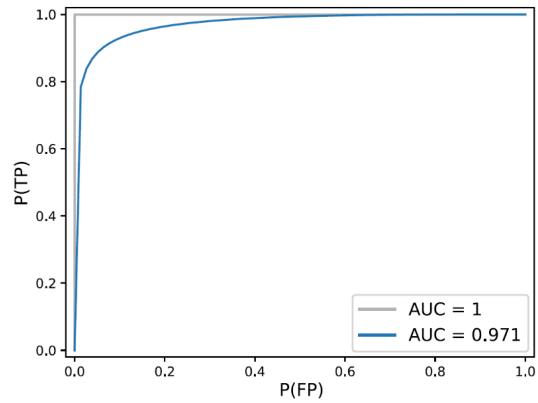

Figure 5: The ROC shows the performance by classification. The calculated AUC is close to 97%, wich means, that the model has a good class separation.

Conclusion and Future Work

In summary, this paper argued that short term precipitation prediction can be done by only observing radar data and without any other source of information. The accuracy could still be improved as well at the regression, were another lossfunction may could help, as at the classification were the 'rainfall' class is still undersampled and worse accurate than the 'no rain' class.