OSNABRÜCK UNIVERSITY

MASTER THESIS

Deep Hyperresolution for Weather Forecasting

Author:
Jonas REBSTADT

Supervisors:
Pascal NIETERS
Prof. Dr. Gordon PIPA

A thesis submitted in fulfillment of the requirements for the degree of Master of Science

in

Cognitive Science

August 14, 2019

OSNABRÜCK UNIVERSITY

Abstract

Cognitive Science

Master of Science

Deep Hyperresolution for Weather Forecasting

by Jonas REBSTADT

Driven by the human induced climate change, the significance of extreme precipitation extremes did not only increase the past year, but rather it will increase the next years even more [Stott, 2016]. Due to the relevance of rain not only for humans in their private environment, but also for industrial and agricultural applications, correct and highly resolved weather forecasts are significantly important, but at the same time highly complex, non-linear and resource demanding [Nayak, 2013]. According to that, systems that are able to increase the precision of rain forecast without exorbitant higher computational demand are offering a high value for many stakeholders. The approach presented in this thesis is trying to increase the spatial resolution of a currently productively used forecast model developed from the Deutscher Wetterdienst (DWD) by training a neural network based on higher resolved radar images as target. The concept has been implemented and evaluated, to finally propose next steps for the DeepRain project.

Contents

List of Figures

List of Tables

List of Abbreviations

DWD	**D**eutscher **W**etter**D**ienst
BMBF	**B**undes**m**inisterium für **B**ildung und **F**orschung
JSC	**J**ülich **S**upercomputing **C**entre
RADOLAN	**Ra**dar-**On**line-**An**eichung
ASCII	**Am**erican **S**tandard **C**ode for **I**nformation **I**nterchange
KENDA	**K**m-scale **En**semble **D**ata **A**ssimilation
MPI-M	**M**ax **P**lanck **I**nstitute for **M**eteorology
LETFK	**L**ocal **E**nsemble **T**ransform **K**alman **F**ilters
HR	**H**igh **r**esolution
LR	**L**ow **r**esolution
SISR	**S**ingle **i**mage **s**uper **r**esolution
SR	**S**uper **r**esolution
SRCNN	**S**uper **r**esolution **c**onvolutional **n**eural **n**etwork
SR	**S**uper **r**esolution
MAE	**M**ean **a**bsolute **e**rror
MSE	**M**ean **s**quared **e**rror
PSNR	**P**eak-**s**ignal-to-**n**oise **r**atio
SSIM	**S**tructural **sim**ilarity index
MS-SSIM	**M**ulti-**s**cale **s**tructural **sim**ilarity index
GAN	**G**enerative **A**dversarial **N**etwork
GRIB	**Gri**dded **B**inary
HDF	**H**ierarchical **D**ata **F**ormat
SLEVE	**S**mooth **le**vel **v**ertical
WMO	**W**orld **M**eteorological **O**rganization

Chapter 1

Introduction

Rainfall forecasts are important for many people, not only in their private life. Whole industries like solar energy or agriculture rely on the predictability of weather, either to predict outcomes or to make decisions for example regarding crop rotation. Moreover, the predictions are a foundation of water resource management and integral for flooding prevention [Nayak, 2013]. But even though rain forecasting has a significant importance, current forecasts still lack sufficient precision especially in spatial dimensions. As [Nayak, 2013] mentions, even though rainfall is one of worlds most important tasks, at the same time it is one the most challenging, because rainfall is non-linear and highly complex and highly dependent on the elevation profile. This randomness and its missing continuous characteristics is differentiating rainfall from other parameters like temperature [Nikam and Meshram, 2013].

Due to the importance of weather forecasts, nearly every country established institutions that are responsible for the operation of weather stations, prediction of future weather and alerting in case of extreme weather events. Example for this country wise organized institutions are the Royal Meteorological Institute of Belgium [RMI, 2019], the Indian Meteorological Department [IITM, 2019], the Hong Kong Observatory [HKO, 2019] or the Deutsche Wetterdienst (DWD) [DWD, 2019c]. Next to these, bigger organizations like the National Oceanic and Atmospheric Administration for the US are composed of different organizations like the National Weather Service, the National Centers for Environmental Prediction and below that nine service center. In Europe the European Centre for Medium-Range Weather Forecasts (ECMWF) [ECMWF, 2019] is working on top of local institutes like the DWD as research institute and operational service, providing data for their members. But next to the ECMWF there are other cooperations developing local as well as global forecast model as for example COSMO and ICON [Baldauf, Klink, and Reinhardt, 2016, Reinert et al., 2019].

This thesis is embedded in a BMBF-Project called DeepRain, coordinated

by the Jülich Supercomputing Centre (JSC). It is a cooperation of five institutions (JSC, DWD and Universities of Osnabrück, Bonn and Bremen) joining a wide variation of knowledge [JSC, 2019]. The project focuses on the improvement of precipitation forecasts in Germany using machine learning. The thesis should offer first insights in the potentials and challenges of machine learning in the precipitation context as baseline for the project, focused on the increasement of spatial resolution for the currently used prediction models offered by the DWD.

To achieve a higher spatial resolution in the model forecasts itself, many additional parameters need to be added, resulting in a significant increase of processing time. To resolve this problem, an approach is to learn the mapping from low resolution COSMO-DE(-EPS) data towards high resolution radar data. For a solid understanding of the used model, the strength and the limitation of the currently used model in Germany are presented as first part of the thesis. Next hyper- and super-resolution are introduced as base for the mentioned mapping and a systematic literature review is done to show the current state of research in the field of super-resolution for weather prediction. Using this theoretical foundation, a general concept has been developed, implemented and evaluated to finally conclude the thesis and give suggestions for next steps in the ongoing project.

Chapter 2

Theoretical Foundations

2.1 Tasks and Products of the DWD

2.1.1 Introduction

Founded as executive agency of the Federal Ministry of Transport and Digital Infrastructure, the DWD has nine main tasks, as officially defined in [DWD, 2018].

First, the DWD should offer climatologically and meteorologically services for individual customers, as well as for the general public. Examples of individual customer are trade, industry, water management and environmental protection. Second, it has the duty to provide information for maritime shipping, traffic routes and other critical infrastructure as for example energy supply. Third, the DWD has to publish warnings about weather phenomena that may be a potential danger for public safety directly or indirectly by their relation to critical weather or climate events. Fourth, the DWD has the responsibility to make short- and long-term observations of meteorological processes as well as the composition of the atmosphere. Fifth, it has to detect interactions between the atmosphere and other environmental areas out of a meteorological perspective. Sixth, the DWD has to create forecasts of meteorological and climatological processes and has to analyze climate change and its impact. Seventh, especially in case of emergencies, the DWD has to track radioactive substances in the atmosphere and predict the future transport. Eighth, the DWD should operate the necessary infrastructure to ensure the duties listed beforehand. Ninth, the DWD needs to take care of availability, archiving, documentation and release of climatological and meteorological services and data.

The mentioned duties result in different products of the DWD providing the required information. Below a selection of these products is presented:

COSMO-DE:
Model generating forecasts for Germany with a grid point distance of 2.8 km.

COSMO-DE-EPS:
Ensemble of models generating forecasts for Germany with a grid point distance of 2.8 km.

COSMO-D2:
Extension of COSMO-DE, resulting in a model generating forecasts for Germany with a grid point distance of 2.4 km.

COSMO-D2-EPS:
Extension of COSMO-DE-EPS, resulting in an ensemble of models generating forecasts for Germany with a grid point distance of 2.4 km.

ICON:
Model generating forecasts for the whole world with a distance around 13 km between points.

ICON-EPS:
Ensemble of models generating forecasts for the whole world with a distance around 13 km between points.

RADOLAN:
Composite of radar data in Germany approximating the rainfall close to the ground with the shortest possible temporal delay without losing to much data quality.

Due to the focus of this thesis and agreements with project partners like the DWD, in the following chapter COSMO-DE, COSMO-DE-EPS, ICON and RADOLAN are presented in more detail. The information about the presented model is based on different handbooks published by the DWD, that are mainly [DWD, 2017], [DWD, 2018], [Baldauf, Klink, and Reinhardt, 2016], [Theis et al., 2017] and [Reinert et al., 2019]. To increase the readability of the chapter, an individual citation of each sentence has been omitted.

2.1.2 Radar data

The first product presented is RADOLAN (Radar-Online-Aneichung), a radar based system. It is used to get insights into the current and past status of rainfall in Germany [DWD, 2017]. Under development since 1997 and live sice

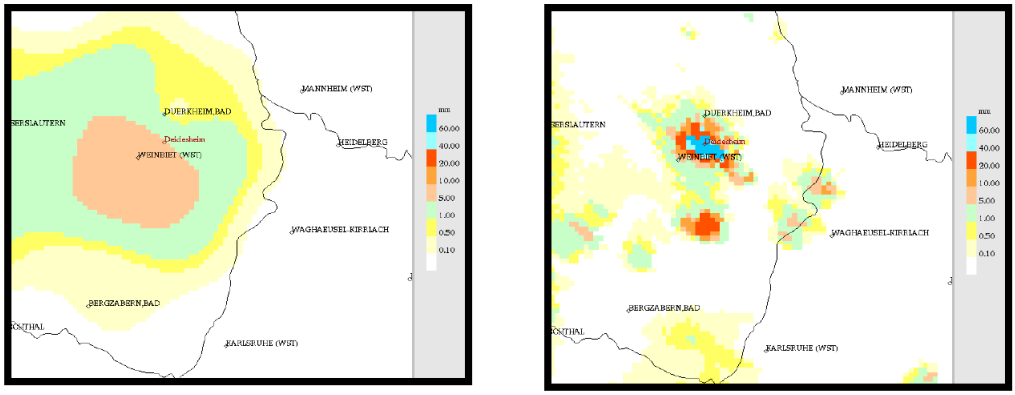


FIGURE 2.1: Comparison of ombrometer data (left) and radar data (right), based on [DWD, 2018].

2005, RADOLAN is a composit of 17 c-band radar stations. It has been developed to get close to live information about rainfall and floods in case of extreme weather events [DWD, 2018]. Relevant for this is mainly the amount of rain or alike phenomena (e.g. snow) close to the ground. But radar is only able measure reflections of rain in the atmosphere around the available radar stations. Based on that even in a composed manner the computed values are quite different from the ones measured by ombrometers on the ground, as one can see in figure 2.1. On the left side the rainfall measured by ombrometers is shown and on the right side using uncorrected radar data is presented.

Regarding to that it is either possible to use the obrometer data directly (and accept a quite high temporal delay) or the radar data needs to be improved to get data with an acceptable quality and small temporal delay. Due to the focus of RADOLAN presenting near to live data in case of floods or comparable scenarios it is important to keep the temporal delay as small as possible. Because of that the DWD uses ombrometer data to adjust the radar data and get improved quality while keeping a quite small delay. Example results can be found in the appendix under A.1 for an extract and under A.2 for whole Germany.

To get results like this, 17 c-band radar stations with a radius of 150 km are are used as well as more than 1200 ombrometer stations. One major difference between these data sources it the temporal delay between measurement and provision of data. Radar data is updated every five minutes, and ombrometer stations are delivering their results hourly. The combination of this values can be done in different ways, but independent of the aggregation algorithm it is not trivial to achieve high dataquality. To get reasonable data, only values with a maximal distance of 100 km to the next radar station and

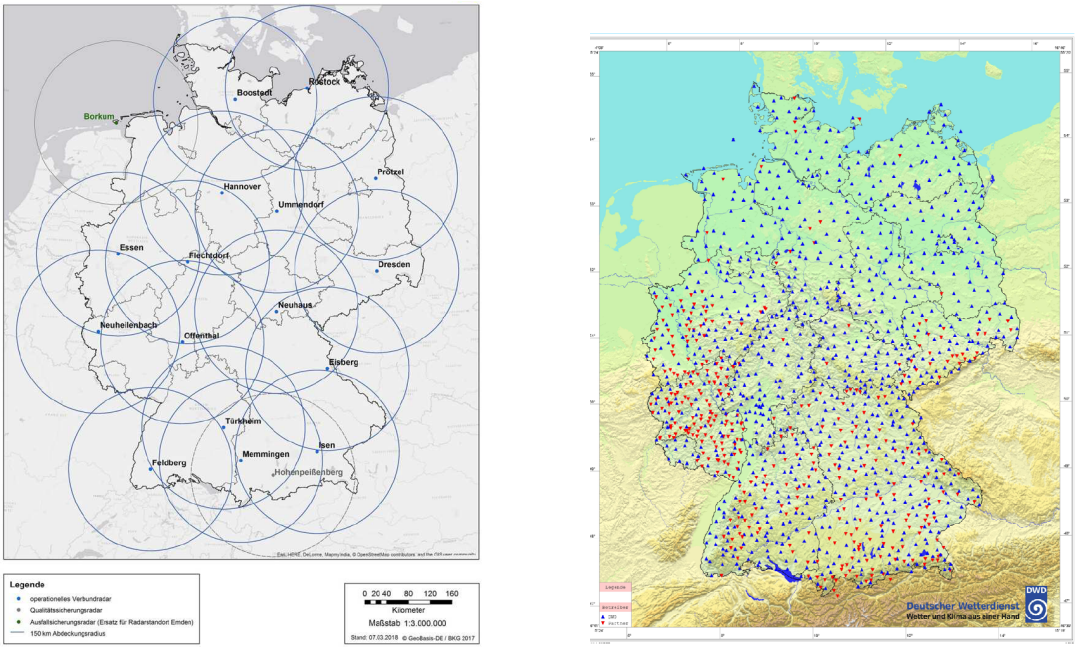


FIGURE 2.2: Network of radar stations (left) and ombrometer stations (right) in Germany, based on [DWD, 2018].

areas with at least 40 to 60 obrometers in the respective 100 km radius should be used. This prerequisites are only fulfilled completely in Bayern, Baden-würthenberg, Rheinland Pfalz and Sachsenanhalt as examinable in figure 2.2.

The expansion of this network as well as the integration of radar stations outside of Germany in RADOLAN is under development. Currently, the product with the shortest possible temporal delay without losing to much data quality is the hourly RW-product accessible for example on a ftp-server of the DWD [1]. The data is saved in binary or ASCII (short for American Standard Code for Information Interchange) format with 900 times 900 data points (810000 in total) using two byte per data point with the little endian convention. More about the extraction of this data can be found in chapter 4.

2.1.3 Prediction model

2.1.3.1 Overview

Next to RADOLAN as radar based product, several prediction models have been developed from the DWD over the years. The more recent ones are distinguished in local models, mainly developed for Germany, and global models, offering predictions for the whole world.

[1] $ftp://ftp-cdc.dwd.de/pub/CDC/grids_germany/hourly/radolan/$

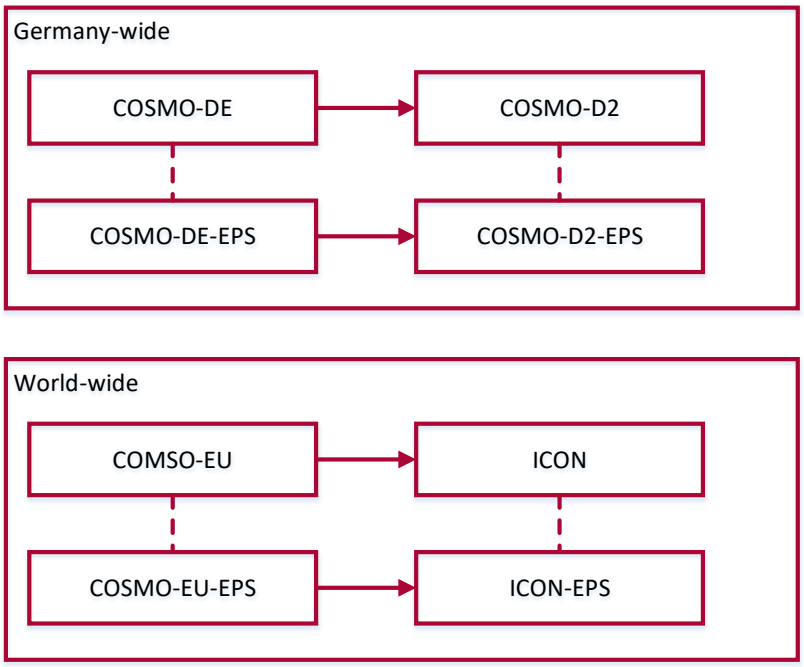


FIGURE 2.3: Development of COSMO model.

The currently relevant local models of the DWD have been developed under the head of COSMO, the consortium for small scale modeling [Baldauf, Klink, and Reinhardt, 2016]. COSMO was formed in 1998 by the DWD, the Geoinformationsdienst der Bundeswehr, the Weather center of Italy, Greek, Poland, Romania, Russia and Switzerland. It has been extended by Israel in 2016 and has the order to improve the formerly independent developed model of each country.

Profiting from this cooperation, the DWD developed COSMO-DE as base and COSMO-DE-EPS as probabilistic ensemble model. As shown in figure 2.3 COSMO-DE has been extended to COSMO-D2 and its probabilistic extension COSMO-D2-EPS. Apart from this local model, global model like COSMO-EU have been developed to offer predictions for the whole world. But similar to COSMO-DE, major improvements lead to two new models called ICON, and ICON-EPS, the first one as base model and the second one as probabilistic ensemble model setting up upon ICON.

In the next steps COSMO-DE, COSMO-DE-EPS as local model will be explained in more detail, as well as ICON and ICON-EPS as global model.

2.1.3.2 COSMO-DE

The two probably most important prediction model for this thesis are COSMO-DE and COSMO-DE-EPS. As already mentioned COSMO-DE is the foundation of the COSMO-DE-EPS model offering predictions for Germany, the Alps (Austria and Switzerland) and parts of the neighbouring countries [Baldauf, Klink, and Reinhardt, 2016]. The model gets started eight times per day predicting values up to 27 hours into the future. It needs around one hour for the prediction run on the mainframe of the DWD, as shown in table 2.1.

TABLE 2.1: Start and end points of prediction runs.

Starting time (UTC)	**Completion time (UTC)**
00:00	01:00
03:00	04:00
06:00	07:00
09:00	10:00
12:00	13:00
15:00	16:00
18:00	19:00
21:00	22:00

COSMO-DE is using a model consisting of unfiltered euler equations of hydro-thermodynamical base equations, which are solved using spatial discretization and temporal integration. To model dynamics of the world's atmosphere and the resulting weather, several physical parameter have to be taken into account. Important are for example radiation, different types of precipitation, moisture convection, partial clouds, vertical turbulent streams and soil processes. To show the complexity of the processes that is required to get reliable results, the differentiation of precipitation is shown with the resulting processes. First, it is important to differentiate between water vapor, cloud ice, cloud water, rain, snow and graupel. To model a dynamical system upon this, many transitions between them need to be included. This results in a quite significant complexity of 10^8 - 10^9 differential equations that need to get solved using time integration [DWD, 2019b]. A graphical overview of the mentioned transitions is presented in the appendix at A.3.

Given a model definition, the initial settings of parameter and border handling are the next steps. In COSMO-DE the initial setting is defined using

analyses of a driving model like ICON (further described in the next chapter). Despite the quality of this model, the initial state is not sufficiently defined. This may result in a spin-up[2] after tree to six hours lead time. Different techniques haven been investigated and some of them are still used to improve the results. One technique is called nudging, including current values as for example radar data to increase predictions. Additionally, it is important to add informations that are not modeled due to the spatial focus. To include relevant information from outside of Germany that are influencing the model, interpolated data from the ICON-model is used.

Predictions of COSMO-DE are organized in a grid-like structure with 2.8 km precision of the grid elements horizontally and 50 different layers vertically. The grid is further described by an underlying coordinate system. Each data-point's position in the grid has rotated (λ, φ)-coordinates in horizontal direction. They are based on geographic (λ_g, φ_g)-coordinates with a shifted north pole. This kind of shift is resulting in a north pole located in the pacific ($40.0N/170.0W$). But even though for further use of these grids transformations back into polar coordinates are required, it is offering higher flexibility and increased computational efficiency. Using the mentioned precision of 2.8 km, the grid has a size of 421 x 461 fields, resulting in 194081 grid points in total. To save this grid in a standardized data format Gridded Binary (short GRIB) is used. GRIB itself is described in the implementation chapter.

2.1.3.3 COSMO-DE-EPS

COSMO-DE-EPS is an ensemble prediction system which has been initiated 2007, fully based on COSMO-DE and ICON [Theis et al., 2017]. It is a probabilistic model combining multiple instances of COSMO-DE with different starting conditions to improve the predictions of the model without changing the dynamical system itself. The resulting grid is unified with COSMO-DE, similar to the operation times, starting predictions eight times per day as shown in figure 2.1. To achieve an improvement, 20 ensemble member are generated with different properties. Important properties to vary are border conditions, starting conditions, model physics, soil moisture and sea surface temperature.

Border conditions of COSMO-DE-EPS are computed using ICON-EPS data. As mentioned before, ICON-EPS is the ensemble model based on ICON (cf. section 2.1.3.4). But even though ICON-EPS has a finer resolution in Europe

[2] Adjustment of flows to the highly resolved topography

than in the rest of the world (20 km vs. 40 km), it is still necessary to interpolate the values used for the borders of COSMO-DE-EPS. To compute these values, ensemble data assimilation is used. 40 different ICON-EPS model are calculated from which 1 - 20 member are selected as base for interpolation.

But not only border condition need to get varied, also for starting conditions of the model differentiation is required. Even though the starting conditions should only represent the current situation of the environment, due to measurement errors the deviation to reality is already to big. To eliminate or at least minimize this error Km-scale Ensemble Data Assimilation (KENDA) [Schraff et al., 2016] is used. KENDA uses Local Ensemble Transform Kalman Filter (LETKF) to estimate the actual state of the atmosphere as well as the analysis error (random, not systematic) based on current observations and a-priori information. Using this, it is possible to align analysis ensemble mean and the most probable current state as well as ensemble spread and estimated analysis error. Parameter that need to be modeled and can be varied are 3D-windcomponents, temperature, specific humidity, cloud water, cloud ice and pressure on lower model areas. All other variables are set to their one hour predictions. To improve this method even further, KENDA is used in combination with latent heat nudging [Stephan, Klink, and Schraff, 2008].

Similar to this, disturbances in soil moisture and sea surface temperature are added. This is necessary, because uncertainties of heat flowing from earths surface into the atmosphere need to be taken into account.

Additionally, model physics need to be taken into account. Twelve parameters are varied randomly, securing that in at least 50% of the member standard values without disruptions are used. In comparison to the mentioned nudging, this parameters are constant during one prediction run. A graphical overview about these starting conditions is presented in the appendix at A.4.

Based on these variations, it is possible to generate different ensemble member. In case of COSMO-DE-EPS the assignment of starting conditions to member is not as static as in older ensembles. Rather the nth ICON-EPS model and the nth member of KENDA are used for starting conditions of the nth COSMO-DE-EPS model in combination with randomized physical disruptions.

As already mentioned the output of COSMO-DE-EPS is using the COSMO-DE grid. Additional to the values of each member, the product is offering *ensemble mean*, *ensemble spread* defined for example using standard deviation

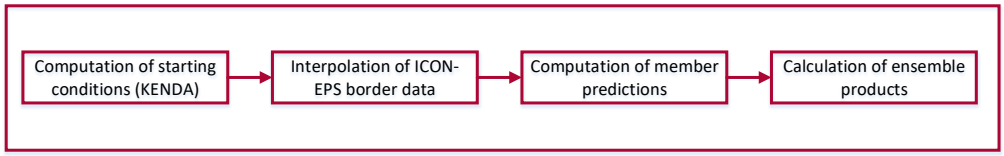


FIGURE 2.4: Steps required for operational use of COSMO-DE-EPS, based on [Theis et al., 2017].

or the distance between 25% and 75% quantiles, *maximum and minimum of all member*, *quantiles* and *probabilities*, offering for example information about the exceedance of a threshold regarding precipitation.

These values are computed separately for each grid-point, but it is also possible to generate these values for less fine grained spatial or temporal resolutions. To achieve this the values of more hours are simply joined or the width of grid elements is increased to for example 28 km.

Summed up for the operational usage of COSMO-DE-EPS, four main steps have to be executed, as graphically presented in the appendix at A.4. First, the computation of initial settings using KENDA. Second, the interpretation of ICON-EPS data for border interpolation. Third, the member itself need to be computed. And last, ensemble products like ensemble mean are determined.

2.1.3.4 ICON and ICON-EPS

In comparison to COSMO-DE and COSMO-DE-EPS, ICON and its ensemble model ICON-EPS are global model [Reinert et al., 2019]. Developed jointly by DWD and Max Planck Institute for Meteorology (MPI-M) in Hamburg, ICON is operational since 2015. The model is used from the MPI-M as core for their climate modeling system, from the DWD for short/ medium range global forecasting and wave modeling and from the German armed forces as relocatable local model.

The prediction process itself is similar to COSMO-DE. Due to its properties as numerical weather model, one of the major topics is the initial value problem. It is highly dependent on estimations of the current atmosphere

FIGURE 2.5: Icosahedron and its projection onto a sphere, based on [Reinert et al., 2019].

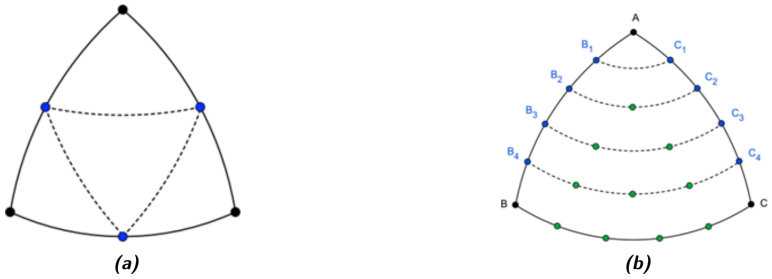


FIGURE 2.6: Used seperation steps in ICON (left: bisection, right: section devision), based on [Reinert et al., 2019].

state. In general, the estimation is done by combining all available observations with a first guess (initial prediction step) of a short-range model forecast. Due to its complexity, combining these parts is not trivial and therefor done using an ensemble variational data assimilation system. This system is called EnVar and is offering initial values for the deterministic ICON-model. EnVar is implementing a variational data assimilation system based on LETKF for ensemble model as mentioned in section 2.1.3.3.

Even though this part on ICON has its similarities to COSMO-DE, the output format of ICON is completely different. Starting with the need of an efficient way to represent points on the whole globe. The horizontal part of the used grid is based on a projected icosahedron. It generates spherical triangles spanning the entire sphere, as shown in figure 2.5.

The resulting representation gets improved by two main steps. First, the initial 20 triangles are divided into n sections each, as presented on the right side of figure 2.6. In the second step, a bisection is applied k times on all resulting triangles (left side of figure 2.6).

This can be done for a variation of n and k values, resulting in different grids as for example the left and the middle one in figure 2.7. In comparison to this, the traditional lat-/lon-grid as used for COSMO-DE is plotted.

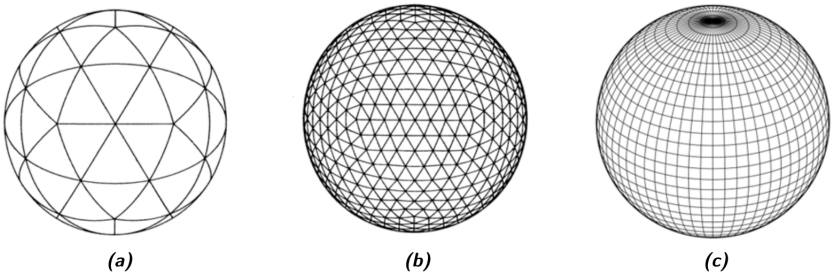


FIGURE 2.7: Different grid types (left R2B00, middle R2B02 and right traditional lat-/lon-grid), based on [Reinert et al., 2019].

This kind of icosahedron based generation is mainly responsible for the naming of different grid solutions. They are named based on the number of applied sections (n) and bisections (k), resulting in the following naming:

$$R\underline{n}B\underline{k}$$

If for example in the first step *2* sections are used and *0* bisections are applied, the resulting name is

$$R\underline{2}B\underline{00}$$

and the generated grid is the left one in figure 2.7.

Equally to COSMO-DE, a dynamical system representing the weather of the world realistically can hardly be 2-Dimensional. Regarding that, the vertical dimension of the grid is added, by simply using multiple horizontal grids stacked above each other. This process generates a set of vertical layers with height based vertical coordinates. To improve simulations, the layer are not spaced equally. Rather, they are defined using smooth level vertical coordinates (also called SLEVE) [Leuenberger et al., 2010], as visualized in the appendix at A.5. This kind of structure is avoiding flat top layer but is also offering a higher resolution in the lower layer given a fixed number of 90 layers.

In the beginning of ICON the mentioned grid was used for all required global predictions. But due to increasing demand of more fine-grained resolutions in Europe, the model has been extended. A nest with higher horizontal resolution has been added for Europe by bisecting the triangles lying above Europe one additional time, presented in the appendix at A.6.

But due to these additional grid points the computation time needed to calculate this model increased rapidly. To allow a similar operational use

as done for COSMO (eight runs per day) the vertical resolution had to be decreased above Europe. Instead of just removing every x layer, the density of layers has been kept constant in the lower atmosphere, but all layers above the lower stratosphere have been removed. In comparison, the global model defines layers up into the mesosphere.

As already mentioned, it is also possible to combine different runs of the deterministic ICON model to create an ensemble model called ICON-EPS. The model is defining 40 members with 40 km resolution globally and 20 km above Europe. The model is used for boundaries of COSMO-DE-EPS and due to that it needs a similar time schedule with at least eight runs per day. But in comparison to COSMO-DE-EPS, ICON-EPS is partially offering predictions up to 180 hours into the future. The time schedule is defined as presented in 2.2, broken down into Europe specific and global predictions:

TABLE 2.2: Maximal forecast times for each model run (lead-time in hours).

Starting time (UTC)	**Maximal forecast time refinement Europe**	**Maximal forecast time world wide**
00:00	30	180
03:00	30	30
06:00	180	180
09:00	30	30
12:00	30	180
15:00	30	30
18:00	180	180
21:00	30	30

Similar to COSMO-DE-EPS, one important aspect is the definition of the different initial members. Because the general procedure is similar to the one presented in the chapter before (except the border problem) it will not be described in more detail. If one is interested in a more detailed discussion of initial perturbations, implementations and evaluations [Hunt, Kostelich, and Szunyogh, 2007], [Schraff et al., 2016] and [Freitag and Potthast, 2013] are good starting points.

2.1.3.5 Strengths and Limits of prediction model

The presented models and their improved versions offer predictions that are hardly reachable by other model for Germany. Due to that many companies

in aviation industry or maritime sector are using the DWD model as foundation of their process [DWD, 2019a]. But nevertheless there are some shortcomings and problems limiting further improvements. As mentioned, with 10^8 - 10^9 differential equations just for COSMO-DE the models are highly complex itself and include a lot of observed and simulated information. But due to the resulting complexity much computation power is required to increase the local precision of the model even a little bit. Because of that it may be interesting to look into other ways to increase spatial resolution than including more variables into the existing model. One promising concept may be hyper-resolution, already used to increase the resolution of natural images, as presented in the following chapter.

2.2 Hyper-resolution

2.2.1 General concept and classical approaches

Even though it is quite straightforward that the term hyper-resolution is correlated with high resolutions in some way, its use in the weather domain is ambiguous. On the one hand it is used to describe highly resolved input variables for traditional prediction models (e.g. orography) [Beven et al., 2015, Singh et al., 2015], but on the other hand it can be interpreted as the exaggeration of super-resolution as a process of resolution increasing on images or grid oriented data [Yang, Ma, and Yang, 2014, Timofte, De Smet, and Van Gool, 2015]. In the following, the term super-resolution is used for the latter. In the super-resolution process a high resolution (HR) image is generated only using low resolution (LR) versions of itself. The problem is trying to fill the information gap between the existing LR image and the required HR image in a meaningful way. From a more mathematical perspective it is a one-to-many mapping from LR space to HR space [Shi et al., 2016]. Due to this, the problem is underdetermined inverse [Dong et al., 2016] and not trivially solvable [Shi et al., 2016]. The key aspect that makes this process of information generation to some extent possible for natural images is the assumption of redundancy in the specific image (local spatial correlation) or in between different images [Shi et al., 2016]. For other grid oriented data this is not proven. But for a successful super-resolution process either redundancy or some kind of additional input is required because information cannot be generated out of nowhere [Shi et al., 2016].

Super-resolution is an important and already well-established research-field in the case of natural images and is used in several areas as for example medical, satellite imaging or face recognition [Peled and Yeshurun, 2001, Shi et al., 2013, Thornton, Atkinson, and Holland, 2006, Gunturk et al., 2003]. To move from low resolution images to high resolution images different initial settings are possible. If the number of LR-images available is greater than one, the combination of these images can offer information that can be used to generate one image with an increased resolution [Farsiu et al., 2004]. The tackled problem differs if only one image is available, resulting in diverging applicable techniques. Due to the similarity to the use-case of this thesis, the important part is single image super resolution, also called SISR [Kim, Lee, and Lee, 2015a]. But even for SISR the variety of usable techniques is manifold.

Techniques used for natural images are divided into several categories dependent on the current focus and age of the paper [e.g. Dong et al., 2016, Shi et al., 2016, Daniel Glasner, 1993, Yang, Ma, and Yang, 2014]. In the following it will be exemplary divided into patch-based, edge based, image statistic based and prediction based. Patch based techniques are trying to find examples of the image that are repeated with different resolutions and replace the LR-parts with the HR-parts [Daniel Glasner, 1993]. This can be done in the image itself, using the additional assumption of mathematical self-similarity (e.g. in images of specific plants or animals) [Huang, Singh, and Ahuja, 2015] or using patch dictionaries containing LR-, HR-image pairs. In this case the algorithm is trying to find patches in the images that are saved in the dictionary in low resolution and exchange them with a corresponding HR-parts saved in the dictionary [Wang et al., 2012]. Edge based approaches are trying to learn priors from edges to especially generate sharp edges in the SR process [Sun, Xu, and Shum, 2011, Freedman and Fattal, 2011]. Even though this might work quite well for edges itself it has shortcomings in SR of other image parts as for example texture. Image statistic based methods are using statistical methods to improve specific properties of an image. Examples for this are the sparsity properties of large gradients or the exploitation of heavy tailed gradient distributions [Shan et al., 2008, Kim and Kwon, 2010]. In comparison to these explicitly defined concepts, non-classical prediction based approaches are trying to increase the resolution without manual engineering. In most cases the LR-HR-image pairs are used to train machine learning algorithms like neural networks or random forests to be able to predict HR images given only its LR-counterpart. One important aspect that

many of these SR techniques have in common is that only one channel is used for the process [Dong et al., 2016]. In this case mostly gray scale images are used or more color channels are given, the images are transferred into another color space (like for example YUV) and only the luminance channel is used for SR. For more information on classical SR methods [Yang, Ma, and Yang, 2014] provides a detailed evaluation of methods prior to the increasing usage of neural networks. In the following, neural network based methods will be presented in more detail.

2.2.2 ANN based super-resolution

In the recent past, more and more neural networks have been applied to solve the problem of super-resolution (SR) for natural images in an easy and successful way [Dong, Loy, and Tang, 2016, Dong et al., 2014, Kim, Lee, and Lee, 2015a]. Early models have been quite small [Shi et al., 2016] but the number of layers has been increasing over the years [Dong et al., 2016] as well as the usage of other promising methods like recursive [Kim, Lee, and Lee, 2015b] or residual connections [He et al., 2016]. One of the initial convolutional neural networks used for super-resolution has been developed by [Dong et al., 2016]. Before neural networks have been common, three main steps were applied for patch-based SR. These steps are patch extraction and representation, non-linear mapping and reconstruction. In the first step, overlapping patches are extracted from a LR image and added into a feature map consisting of one high dimensional vector for each patch. In the second step, these vectors are mapped in a non-linear way onto other vectors still representing one patch each. Using this high dimensional feature map, in the last step these vectors are aggregated into a final HR image. [Dong et al., 2016] reformulated these steps creating an initial three layer deep neural network as presented in figure 2.8.

In figure 2.8 every step is interpreted as one convolutional layer. First, an n-dimensional feature map is created. Second, all vectors are mapped in a non-linear way. Third, the feature map is aggregated using 1x1 convolutions. Based on this, a first successful super resolution convolutional neural network, short SRCNN, was developed. The only preprocessing step required for this network is a bicubic interpolation to increase the size of the LR image to the size of the HR image that should be generated. Next to this first version, [Dong et al., 2016] already presented suggestions for improvement, like adding layers for the non-linear mapping step. In the following more and

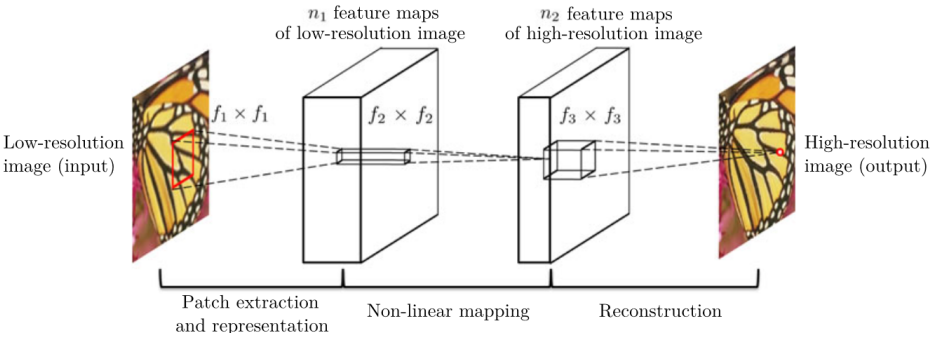


FIGURE 2.8: Procedure of super resolution as base for neural networks [Dong et al., 2016].

more ideas have been implemented to improve SR. There are two main aspects that offer a high potential of improvement. One discussed point is the moment and method of resolution increase during the process. To achieve SR, at some point the LR image needs to match the size of the HR image. [Dong et al., 2016] are increasing the resolution as in initial step using bicubic interpolation. But as [Shi et al., 2016] mention, interpolation methods are not adding information and therefor are not helpful for the SR process. Instead, [Osendorfer, Soyer, and Smagt, 2014] are gradually increasing the resolution during the process. This gradual increase can be done in several ways. One possibility is to use pooling layers. But even though pooling layers do not need additional parameters that need to be covered by additional data, [Johnson, Alahi, and Fei-Fei, 2016] state that it may be a significant improvement to use convolutional layers with a stride of $\frac{1}{2}$ instead. As already mentioned, the second important aspect is the amount of data required for the training process. [Kim, Lee, and Lee, 2015b] tackle this problem by applying recurrent layer instead of simply stacking convolutional layer. Using this it is possible to increase the filter size indirectly and at the same time keep the number of required parameters low. Next to this, most techniques from neural network based image classification involving convolution can improve the quality of SR, as for example the use of residual connections [He et al., 2016].

Next to the defined architecture, [Johnson, Alahi, and Fei-Fei, 2016] figured out the high influence of loss functions to achieve satisfying SR. The problem in SR for natural images is that the perceived quality of the resulting images is not necessarily congruent with a low per-pixel difference. But measuring the perceived quality of an image mathematically usable as loss for a

neural network is not trivial. Because of that, a variety of different measurements are used to quantify the quality of a generated image. The probably most common ones are measurements based on per-pixel differences like l_1- or l_2-norm, also called mean absolute error (MAE) and mean squared error (MSE) [Zhao et al., 2016] as defined in equations 2.1 and 2.2.

$$MAE(f,g) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}|f_{ij} - g_{ij}| \tag{2.1}$$

$$MSE(f,g) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(f_{ij} - g_{ij})^2 \tag{2.2}$$

Using this simple per-pixel measurements one of the most known measurements for image generation has been developed, the so called peak-signal-to-noise ratio or in short PSNR [Horé and Ziou, 2010]. PSNR is defined based on the Mean Squared Error in the following way:

$$PSNR(f,g) = 10\log_{10}(\frac{255^2}{MSE(f,g)}) \tag{2.3}$$

Due to the possible number of values each color channel can chose in case of natural images (in RGB), PSNR is defined with 255 in front of MSE. Next to per-pixel measurements like PSNR, the structural similarity index (SSIM)[Wang et al., 2004] and its extension the multi-scale structural similarity index (MS-SSIM) [Wang et al., 2003] are two of the most established indexes. They are differentiable, which allows their use in backpropagation based algorithms like neural networks [Zhao et al., 2016]. SSIM has been explicitly designed to approximate human based visual perception by combining luminance distortion, loss of correlation, and contrast distortion, as shown in equation 2.4.

$$SSIM(f,g) = l(f,g)c(f,g)s(f,g) \tag{2.4}$$

with

$$\begin{cases} l(f,g) = \frac{2\mu_f\mu_g+C_1}{\mu_f^2+\mu_g^2+C_1} \\ c(f,g) = \frac{2\sigma_f\sigma_g+C_2}{\sigma_f^2+\sigma_g^2+C_2} \\ s(f,g) = \frac{\sigma_{fg}+C_3}{\sigma_f\sigma_g+C_3} \end{cases}$$

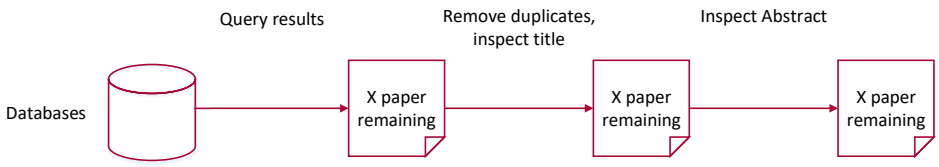


FIGURE 2.9: General procedure systematic literature review.

In this equation μ_f and μ_g are representing the mean luminances of the compared images and the contrast is defined by the standard deviations σ_f, σ_g and the covariance σ_{fg}. The additional constants C_1, C_2, C_3 are used to keep the denominator unequal to zero. MS-SSIM is extending SSIM in a multiscale manner by combining SSIMs on different scales [Horé and Ziou, 2010]. But even though this measurements are quite established, it is promising to define the loss of image transformation processes in a less direct way. One idea to achieve this are Generative Adversarial Networks (GANs) as presented by [Ledig et al., 2017]. [Johnson, Alahi, and Fei-Fei, 2016] are trying to establish a similar idea. They are using the insight that for image classification pretrained neural networks already learned an encoding of semantic and perceptual information. By fixing the network it can be used as loss function. This works quite well for natural images, because classification in this area is a highly established field with a lot of work done.

2.2.3 Existing approaches of Hyper-resolution in weather forecasting

A lot of work has been done in super-resolution of natural images, due to that it is comparably easy to find entry points for an effective literature search. In contrast to that, it is much harder to find all the work done in a specialized area like "super-resolution for weather data" and to ensure, that there are no published approaches implementing the concept presented in this thesis. Due to this, more formalized ways of literature reviews have been developed, up to the grade that the professional realization of review can be scientific problem itself [Fettke, 2006]. Based on [Webster and Watson, 2010] [Fettke, 2006] presented methodological guidelines to ensure the scientific level of reviews. The general procedure used in this paper is presented in figure 2.9.

The first step in this process is the definition of a search term. Important for the search term is the conjunction of "hyper-resolution" as observed

method and the target domain "weather". Due to the already mentioned ambiguous use of the term hyper-resolution and the existence of similar methods, possible other terms describing these methods are included as well, resulting in the following search term:
"(hyperresolution OR hyper-resolution OR superresolution OR super-resolution OR upsampling OR upsampling OR upscaling) AND weather"
This search term has been applied in seven databases. Initially the full text searches have been applied. In that case unfortunately many only marginal connected papers have been found. The resulting number of papers for each database are listed in the second column (Hits without restriction) of table 2.3. To reduce this overhead, the search has been restricted to specific parts of the paper in the second step. The considered fields for each database and the resulting lowered hits are presented in columns three (Considered fields after restriction) and four (Hits with restriction). Additional to the field restrictions in the search, the documents have been limited to articles, to secure the scientific topicality.

TABLE 2.3: Systematic literature review: Number of hits separated for each data base.

Database	**Hits without restriction**	**Considered fields after restriction**	**Hits with restriction**
ScienceDirect	14,222	title, abstract and Keywords	69
EBSCOhost	17	title, abstract and Keywords	1
SpringerLink	1,658	title	767
Emerald	404	title	0
Wiley Online Library	7,139	title and keywords	1
Web of Science	365	topic	234
Google scholar	18400	title	10

This restricted number of papers is the foundation for the following in figure 2.10 presented procedure. For this procedure 1082 paper are considered. In the first selection phase duplicates have been removed and the paper have been selected based on their title, resulting in 52 paper. In the second selection phase this 52 paper have been reduced based on their abstracts. The remaining ten paper are shortly presented below, especially highlighting the differences towards the approach presented in this thesis.

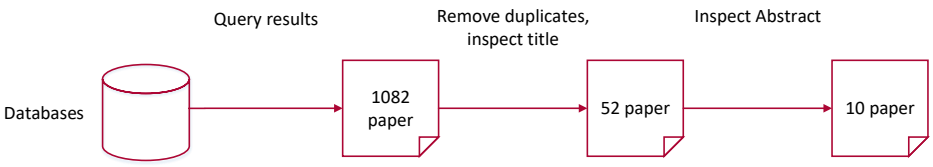


FIGURE 2.10: Specific systematic literature review including number of remaining paper.

[Ahrens and Beck, 2008] evaluated classical upscaling methods like regression or smoothing for an area in Austria and presented their limitations. Model data has been used as base for upscaling methods and rain gauge data as evaluation data. Therefore, the used data is quite similar to the approach in this paper, but the aim of the paper is as different as the used techniques. [Ha, Gowda, and Howell, 2013] presented statistical scaling methods to increase the spatial resolution of evapotranspiration maps to improve irrigation management. Even though the topic is quite similar, neither rain prediction nor neural networks are used. [Berrocal, Gelfand, and Holland, 2010] developed a mapping from model predictions using grid data onto point measures of monitoring networks including downscaling. To allow this kind of mapping correlated spatial Gaussian processes have been used. Even though this may not be neural network based, it is an interesting way of mapping from grid to point data that may be a way of overcoming the limits of aggregated radar data as presented in chapter 2.1.2. [Zhang, Wu, and Yang, 2014] took a closer look towards the super-resolution of radar data, one of the relevant technologies in this thesis. But in comparison to this thesis, the radar data has been used in its initial format and the super-resolution process is defined as equation solving problem. [Mahour et al., 2015] presented an approach of super-resolution for evapotranspiration. Again the general topic is quite similar but the used techniques and the input data differ a lot. [Im et al., 2016] in comparison used machine learning algorithms (random forest, boosted regression tree and Cubist) to learn super-resolution but in this case the used data differs due to the focus on soil moisture. [Li et al., 2012] focused on the horizontal resolution increase of reflectivity data to reduce the error of rainfall retrieval using mainly truncated singular value decomposition. Similar to this, [Zhang et al., 2019] tried to increase the resolution of radar data using non-local self-similarity in sparse representations to analyze small-scale weather phenomena. [Ebtehaj, Foufoula-Georgiou, and Lerman, 2012] shifted their

focus towards image enhancement for remotely sensed precipitation images. They used non-linear estimators to find small details out of a more military oriented perspective. [Ben Bouallègue and Theis, 2014] presented upscaling methods that are actually using the COSMO-DE-EPS model as input. But in contrast to this thesis, no mapping towards radar data has been done, furthermore classical smoothing and upsampling is used including additional neighborhood information.

Chapter 3

Applied approach

Even though COSMO-DE(-EPS) and RADOLAN data itself might contain a lot of information, the main purpose of this thesis is to join these two data sources to reveal additional information useful for rainfall prediction. The drawbacks of adding new variables to model like COSMO-DE(-EPS) to increase the resolution should be overcome by combining existing model with super-resolution methods used for natural images. Inspired by super-resolution for natural images, an algorithm should be developed that is able to map from a low resolution to a high resolution grid. Defined by the DeepRain project and confirmed in their effectiveness by literature of natural image super-resolution, neural networks are used to learn this mapping from low resolution model data (COSMO-DE, 2.8 km precision) to high resolution target data (RADOLAN data, 1 km precision - redefined to 1.4 km precision).

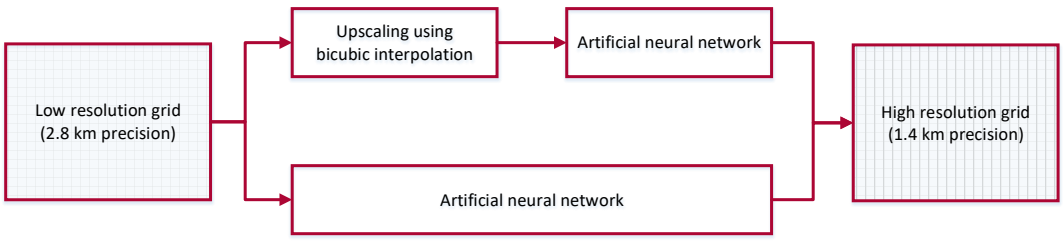


FIGURE 3.1: Upscaling process: General procedure.

In figure 3.1 one can see the two main procedures. They differ only in the upscaling part. The upper version is upscaling the grid in the data generation process using bicubic interpolation. In this case the network joins the different feature maps and increases the quality of the resulting grid. In the lower version, the upscaling process needs to be done by the network itself, offering a higher variability in moment and procedure of upscaling. More

details about the implementation and the chosen network architectures are presented in the following chapters 4 and 5.

Chapter 4

Implementation

4.1 Procedure and general architecture

The project is mainly implemented in Python [Van Rossum and Drake Jr, 1995] and uses TensorFlow [Abadi, Martın et al., 2015] as deep-learning framework. To make sure, that the training process is monitored correctly, Sacred [Greff et al., 2017] is used in combination with Omniboard[1]. For the data generation as well as for the conduction of the experiments the grid system from the Institute of Cognitive Science[2] is used. The project is divided into two main parts, the data generation and the training process of the neural network itself. It has been tried to enclose as much of the process as possible in the data generation step to decrease the total computation time. Due to the single execution of the data generation process for each dataset and the repeated execution of the training process for each model, a shift of as much process-parts as possible towards the data generation seems to be reasonable.

In the next chapter the data generation process is shown in more detail and afterwards the training process will be presented.

4.2 Data

4.2.1 General structure

As presented in chapter 3, the project is trying to learn a mapping between DWD model data (COSMO-DE-EPS) and radar data to increase the spatial resolution of rain forecasts. To train machine learning algorithms, example data is necessary. In this chapter the generation of this dataset is presented. In the process information from files with two different data types are extracted

[1] https://github.com/vivekratnavel/omniboard
[2] www.ikw.uni-osnabrueck.de

out of their file structure, prepared for the needs of the project and joined in one dataset. This process is presented in figure 4.1.

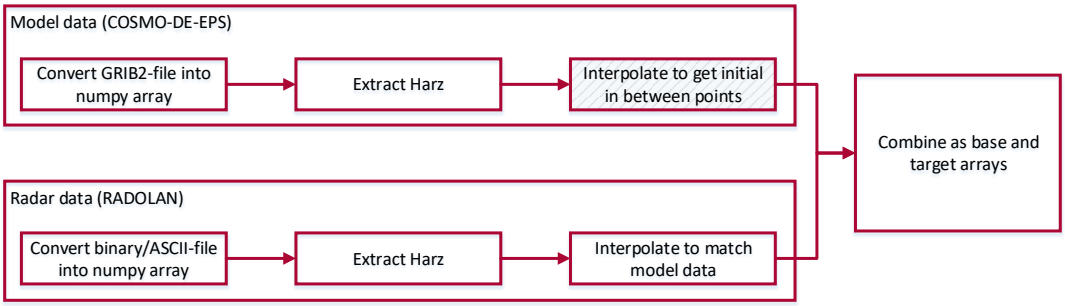


FIGURE 4.1: Procedure for the extraction of weather data.

The generation process is initially split into the preparation of COSMO-DE(-EPS) data and the preparation of radar data to finally join these two parts into one dataset. COSMO-DE(-EPS) data is saved as GRIB2-file. In the first step the data is extracted from these files and converted to a numpy array. Second, all datapoints in the grid not corresponding to the considered parts of Germany (in our case the Harz area) are removed. The third step is optional and is dependent on the aspired procedure in the training step. If the training step requires input grids with the same spatial dimensions as the output grid, at this point interpolation is used to double the size of each spatial dimension. If the algorithm itself is doing the upscaling, the subset is used directly without interpolation. The other part of this dataset is the mentioned radar data called RADOLAN. RADOLAN is offering two main products, summing up rainfall either hourly or daily. The daily summary is saved as a binary file while the hourly data is saved in ASCII-format. According to this two extractors have been defined, converting the input files into numpy arrays. Analog to the procedure for COSMO-DE(-EPS) data, next the Harz area is extracted, to finally interpolate the grid for matching sizes between input and output data.

In figure 4.2 the shapes of the numpy arrays after each generation step are shown. COSMO-DE(-EPS) is offering 461×421 spatial data points representing the area of Germany for each of the selected 9 features. In the following, the spatial dimensions are reduced towards 41×26, by selecting only datapoints inside the Harz region. Afterwards, the array is either prepared for the joined dataset or it needs to be interpolated towards spatial dimensions

of 82 × 52. RADOLAN is starting with 900 × 900 points and is reduced to 82 × 52 in the already mentioned two steps.

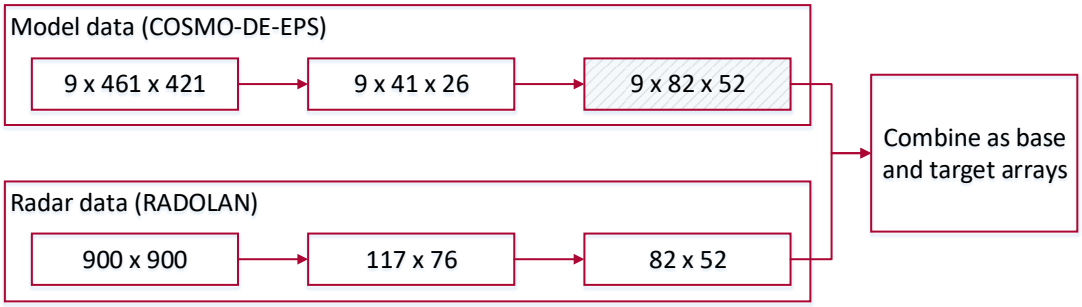


FIGURE 4.2: Shapes during the extraction of weather data.

In the following sections the available data and its extraction process are described in more detail.

4.2.2 COSMO-DE(-EPS)

4.2.2.1 Fileformat - Grib(2)

Data generated by COMSO-DE(-EPS) is encoded in files using the GRIB(2) (General Regularly-distributed Information in Binary form) format. GRIB2 is a data format presented by the World Meteorological Organization [3] (WMO) and used by many meteorological organizations like the DWD [4]. GRIB2 is a table based binary data format specialized on efficiency in saving, transmitting and processing data.

GRIB2 consists of eight different sections. The first six are mainly offering general and structural information to interpret the data correctly. They are followed by the data itself and concluded by an end section.

4.2.2.2 Selection of variables

Prediction models run by COSMO-DE-EPS are simulating the dynamics of many variables. To achieve the defined aim of predicting rain in a higher resolution than currently reasonably possible, a lot of these parameters may be important. Under the premise of limited computational power a selection of the available parameters is necessary. In the appendix under A.2.1 fields offered by COSMO-DE-EPS are listed with the corresponding description (if it

[3] https://public.wmo.int/en
[4] https://www.dwd.de/DE/derdwd/it/_functions/Teasergroup/grib_.html

is available) to show possible input parameter for the used machine learning model.

For the initial dataset used in this thesis the available parameter had to be reduced. The selection has been done during one of the DeepRain-Meetings (7.3.2019 - 8.3.2019) in agreement with the DWD. The remaining parameter are listed in table 4.1, based on descriptions offered by [Baldauf, Klink, and Reinhardt, 2016]. On the left side one can see the shortnames used by the DWD with the corresponding explanations on the right side.

TABLE 4.1: Selection of parameter simulated by COSMO-DE-EPS.

Shortcut	Description
CAPE_ML	Convective Available Potential Energy, mean layer
eva	Evaporation
TQC	Total Column-Integrated Cloud Water
TDIV_HUM	Vertical integral of divergence of total water content (s)
TQI	Total Column-Integrated Cloud Ice
TWATER	Total Column integrated water (all components)
TQG	Total column integrated grauple
u	Zonal wind in 850 hPa
w	Vertical wind in 500 hPa
v	Meridional wind in 850 hPa

4.2.2.3 Data extraction

Even though it is possible to decode GRIB files based on the given definitions manually, it is useful to automate this procedure by using existing software packages like ecCodes [5]. Using this software, the grid based values for each relevant variable in the file can be extracted easily. The selection of relevant parameters is done using the defined shortnames presented in the section before. After the extraction process a numpy array with a shape of $9 \times 461 \times 421$ is handed over to the next preprocessing step, where 9 is defining the number of features and 461×421 is defining the spatial dimensions.

[5] https://github.com/ecmwf/eccodes

4.2.2.4 Required Preprocessing for Hyper-resolution

The extracted data is describing whole Germany. To reduce the relevant amount of data and focus on regions with challenging orography, the data is reduced to the Harz region. To remove irrelevant data points, two other grids defining the coordinates of all points are matched onto the data grid. Having this information it is possible to remove all data point outside this region, but this process is not trivial. In the DeepRain project only corners of the relevant region have been defined using coordinates. Due to the structure of coordinate based grids (as presented in chapter 2) points with the same longitude are not necessarily in the same column. To get a subset with clear edges all rows and columns that do not contain any points with relevant coordinates are removed. This results in in 41×26 grid points for each feature.

Dependent on the used architectures in the training step, the grid is either ready for the combination step or it needs to get upscaled to the same size as the target images. The upscaling is done using bicubic interpolation, generating a grid with $9 \times 82 \times 52$ datapoints. To get a grasp on these grids, in figure 4.3 the grid is shown using one subplot for each of its features.

Due to the significant differences of the value range in between their features, another dataset for each size (with or without interpolation) is generated with feature-wise normalized data. For the normalization, maximum and minimum of all samples have been computed for each feature individually. Using these feature specific maxima and minima, all values have been updated using equation 4.1.

$$normalizedValue[f,m,n] = \frac{value[f,m,n] - minimum[f]}{maximum[f] - minimum[f]} \tag{4.1}$$

In 4.1 m and n are describing the spatial dimensions while f defines the currently normalized feature.

4.2.3 Radar data

4.2.3.1 Fileformats - binary vs. ASCII

The dataset used as target is representing measurements of radar stations in a grid oriented format and is called RADOLAN. It is divided into two parts. The first one is offering information on an hourly basis and the second one is summing up these values to get daily aggregates. For both of them extractors have been defined. In comparison to the hourly data, the extraction of daily

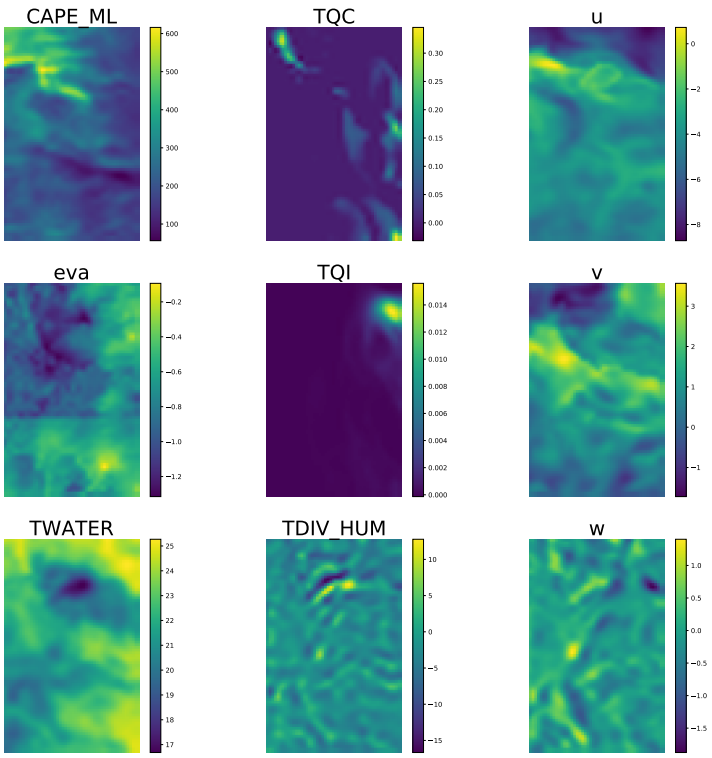


FIGURE 4.3: Overview of the generated input data broken down to its features.

radar data needs to be done by hand, due to its binary format. The binary files consist of a header and a body for the radar data itself. In the header, metadata and information about the format of the radar data are saved. In the body 1.620.000 bytes are saved sequentially. To get reasonable results, these Bytes need to get interpreted in pairs, resulting in 810.000 values. The combination of these pairs is dependent on the currently used operating system, because not every operation system is using the little-endian convention. If the pairs are combined correctly, the extracted values can be reshaped into a 900×900 numpy array as one can see in the appendix at A.7.

Hourly data is saved in ASCII format. This data can be easily extracted using gdal[6] and can be returned instantly without further configurations. In both formats the rain is measured in mm. To get cm based values it needs to be divided by ten.

[6]https://gdal.org/

4.2.3.2 Required Preprocessing for Hyper-resolution

Based on the successful extraction, the removal of non Harz parts of Germany works analog to the procedure for COSMO-DE(-EPS), as presented in the section before. The interpolation of the RADOLAN grid is necessary because the distance between points in the current grid is 1 km. Due to the distance of 2.8 km between points in COSMO-DE(-EPS) a simple upsamplpling with the factor two is not possible. To create a matching ratio, the resolution of the RADOLAN grid is reduced down to a distance of 1.4 km. Similar to the last section, this is done using bicubic interpolation. The resulting grid can be visualized like it is done in figure 4.4 for three different time points.

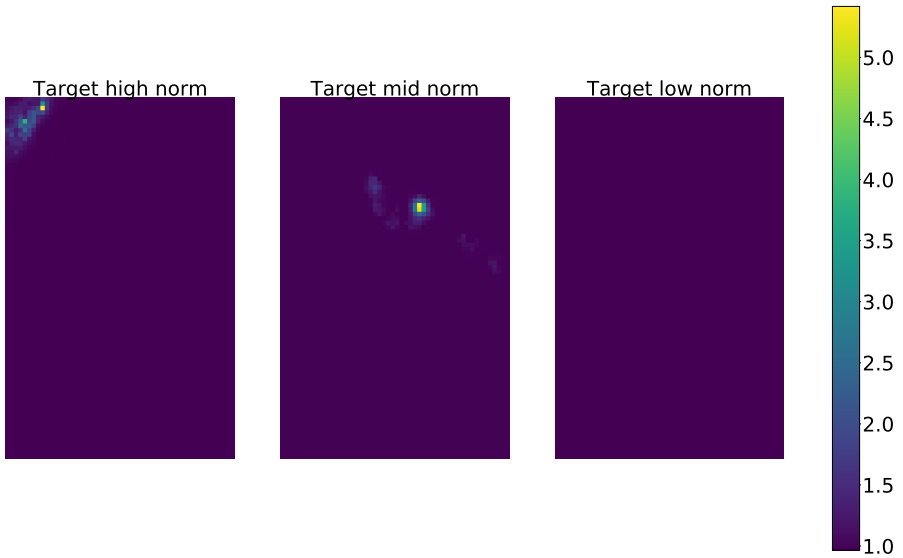


FIGURE 4.4: Overview of the generated target data for three different timepoints.

The three timepoints have been selected by computing the sum over all features of the input data and choosing the ones with the lowest and highest values as well as the one with the most average value. Figure 4.4 is using normalized input data as reference for this computation.

4.2.4 Joined dataset based on COSMO-DE and radar data

To train the neural networks presented in chapter 5, a dataset has to be created. In this dataset it needs to be possible to combine the available parts of

COSMO-DE(-EPS) data with the matching target data, such that for example the prediction of 3 o'clock in the morning with a leadtime of six hours is matched to the radar data of 9 o'clock in the morning of the same day. But on the other hand it is important to keep relevant information like structural information of the data, the lead time, run time or the observation time of every point in time. Due to this and to keep transparency, it is not expedient to simply save the matching pairs without additional information. But on the other hand one wants to be able to create this pairs easily. To achieve this, the following data structure has been created using the fifth version of the Hierarchical Data Format, short HDF5 [Folk et al., 2011].

HDF5 is a data file format used to save data in a flexible, efficient way that is not dependent on a specific language but at least partially transformable to language specific types like numpy. HDF5-files contain groups (white and gray in figure 4.5) and datasets (red in figure 4.5) to structure the data. While groups can be accessed in a dictionary like manner, datasets are similar to numpy arrays, making the use quite intuitive.

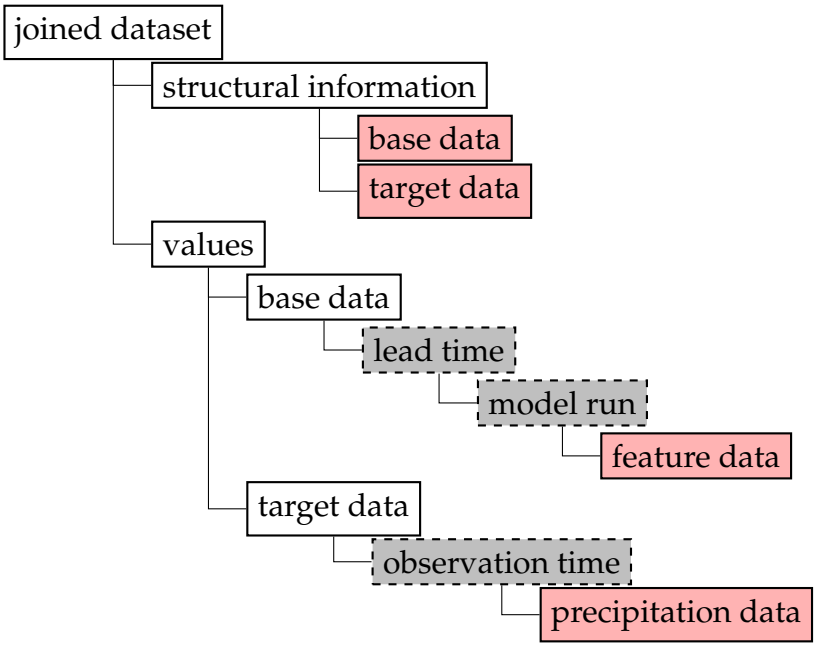


FIGURE 4.5: Structure of the joined dataset.

The HDF5-file saves the required structural information of the two data sources and the values itself. Structural information are mainly arrays with coordinates enabling the reconstruction of spatial position for every part of the grid, saved for target and base data. The values are split into base and target data as well, but they are sorted differently. The feature arrays of the

base data are sorted by lead time and model run time. Precipitation data on the other hand is only sorted by their observation time.

This kind of structure can be used to create matching pairs by iterating over the base data and adding up the lead time and the model run time receiving a corresponding observation time of the target data.

4.3 Algorithms

As already mentioned, there are different techniques usable for super-resolution. But due to the focus of DeepRain, only neural network based techniques are of interest. The networks are implemented in TensorFlow using a generator extracting matching pairs out of the created dataset(s) as described in the last chapter. In this matching procedure for each COSMO-DE(-EPS) model run the RADOLAN grid with the correct time point needs to be extracted by adding up the time point of the model run and the relevant leadtime(s). Based on this generator, models can be defined in lean classes, inheriting from a base class implementing the generator and the remaining parts of the training process. The specific models used for the experiments are described in chapter 5.

Chapter 5

Experiments

5.1 General setup

As evaluated in section 2.2.3 the task this thesis wants to accomplish has not been solved so far. But a lot of progress happened in super-resolution of natural images. Due to this, many papers present usable techniques for super-resolution as listed in chapter 2. Led by the focus on neural network (as defined by the DeepRain project), the relevant techniques are reduced, but still a wide variation of methods is given. To take care of these possibilities a systematic way of testing is required. In this chapter the procedure is presented including the conducted experiments and their results, based on the general assumption, that the task is learnable given the available data.

Initially, if one wants to compare different algorithms like neural networks it is necessary to define metrics. As presented in chapter 2, it is not intuitively possible to create meaningful metrics for the quality of images, especially for natural images, because the perceived quality of images differs from per point measurements. In contrast to that, radar data does not need to look pleasurable to humans, rather it should be point wise as similar as possible to the target image. This is relevant, because every point in the grid is averagely 1.4 km away from the next point and even small pixel wise shifts are highly important for some industries described in chapter 3. Due to this, classical metrics like mean squared error, mean absolute error and peak signal to noise ratio have been selected and accordingly are saved for each experiment. Additionally, for every experiment three data points are selected, to offer a visual intuition of the prediction quality. To achieve a high variability, the input arrays are saved that are resulting in the highest, the lowest and the average value if all values are summed up. For this data points, input arrays and predictions of the network are saved as well as the target arrays.

For machine learning algorithms example data is of significant importance just by definition [Russell and Norvig, 2016]. For the conducted experiments, the available data is highly limited. In general, it is possible to use model and radar data since the first models have been developed. But due to the significant differences between the forecast models it may be promising to focus on a single model, especially because of the different outputs of the models (for example different parameters and varying distances between points). In consultation with all project partners of the DeepRain project the scope of the considered data has been limited again, because the differences between the seasons, between the different leadtimes and between the different runs each day are too high to get reasonable results (more details about leadtimes and executed runs in chapter 2.1.3.2 and 2.1.3.3). Additionally, the different ensemble member may be too similar to add information if all of them are used. In the current state of the project only data from three months has been provided. According to this, data from COSMO-DE-EPS is available for September, October and November 2017, consisting of 20 ensemble members with 40 different leadtimes and eight runs each day. After the reduction of this already limited dataset based to the recommendations of the DeepRain consortium, only 91 data points remain $(30 + 31 + 30)$. To get at least a somehow reasonable number of datapoints, all runtimes of these days are considered, resulting in 728 data point in total $((30 + 31 + 30) \cdot 8)$.

Next the variable parameters need to be defined. Based on the literature of super-resolution for natural images 12 aspects have been identified, that should be checked experimentally. These parameters are *dataset*, *model*, *number of features*, *size of the image in x direction*, *size of the image in y direction*, *number of output channel*, *activation function* and *leadtime* as structural parameter, describing the data and the properties of the used model. Additionally *number of epochs*, *batch size*, *loss function* and *learning rate* are defining the training process of the network in more detail.

Most of these are atomic decisions but some are further reaching, as for example the used dataset and the model. Two main datasets are available based on DWD data. The only difference between these datasets is the size of the input images. Either the input images are already interpolated towards the size of the target images or the input images are kept in their initial size. The probably most important aspect is the chosen model. The model defines the actual network. The varied aspects are the depth of the network, the point of feature combination and where the actual upscaling happens during the process.

By varying these parameters several experiments can be defined. The conducted experiments are presented in the following section.

5.2 Implemented experiments

To evaluate the effect of the changeable parameters the network results given their possible values need to be compared, while the other aspects are kept largely constant. But not all parameters need to be changed. Some of the parameters are only included to allow the expandability of the system or are kept constant initially, because an increase is still possible afterwards. This kind of parameter are number of feature, size of the image in x direction, size of the image in y direction, number of epochs, batch size and number of output channels. In contrast to that, dataset, model, loss function, activation function and learning rate are varied. According to the paper presented in chapter 2, the highest effect may be expected by the choice of a specific model and the corresponding dataset. Loss function, activation function and learning rate may be important as well but probably they will not have as much influence as using a different model. In general one can consider every possible combination of these parameters, but to see the effect of one parameter change, a balance between the isolation of parameters and the resulting combinatorial effects needs to be kept. Due to that, for the evaluation of models and their corresponding dataset, loss function, activation function and learning rate are kept constant. Similar to this the evaluation of loss function, activation function and learning rate is done using one model with its corresponding dataset. This results in the experiments listed in table 5.1, including all parameters that are changed in the different experiments.

Additionally, all experiments are repeated with feature wise normalized input data. With these configurations, the influence of four main aspects have been tested. Initially the moment where the nine input features are combined to finally get one output value for each grid point is evaluated. Second the moment when the size of the input image gets increased during the process is checked. Third the depth of the network is varied and finally the influence of loss function, activation function and learning rate are examined.

For the evaluation of feature combination, the dataset with same sized input and target images (presented in chapter 4) is used. In this context, same sized data means, that the first two dimensions are same, the third dimensions differs even in the same sized dataset because it represents the features

TABLE 5.1: List of parameters for implemented experiments.

Dataset	Model	Loss function	Activation function	Learning rate
same size	combine features last	mse	relu	1e-3
same size	combine features first	mse	relu	1e-3
same size	combine features first extended	mse	relu	1e-3
same size	cnn simple	mae	relu	1e-2
same size	cnn simple	mae	relu	1e-3
same size	cnn simple	mae	relu	1e-4
same size	cnn simple	mae	tanh	1e-2
same size	cnn simple	mae	tanh	1e-3
same size	cnn simple	mae	tanh	1e-4
same size	cnn simple	mse	relu	1e-2
same size	cnn simple	mse	relu	1e-3
same size	cnn simple	mse	relu	1e-4
same size	cnn simple	mse	tanh	1e-2
same size	cnn simple	mse	tanh	1e-3
same size	cnn simple	mse	tanh	1e-4
different size	deeper cnn	mse	relu	1e-3
different size	deeper cnn with small kernel	mse	relu	1e-3
different size	cnn upsampling beginning	mse	relu	1e-3
different size	cnn upsampling mid	mse	relu	1e-3
different size	cnn upsampling end	mse	relu	1e-3

extracted from COSMO-EPS-DE and no spatial dimension. Loss function, activation function and learning rate are kept constant to *MSE*, *relu* and $1e-3$. What differs is the model. In the first case the feature combination using a 1x1 convolutional layer is applied in the beginning, followed by just one 3x3 Kernel afterwards, as one can see below.

```
1 x = Convolution2D(1, (1, 1), activation=self.activation_function,
    padding='same', name='level0')(inputs)
2 x = Convolution2D(1, (3, 3), activation=self.activation_function,
    padding='same', name='level1')(x)
```

LISTING 5.1: Model structure with combination of features in the beginning.

To figure out the influence of feature combination in the beginning or in the end two more models have been defined. The first one simply switches the order of the convolutional layers.

```
1 x = Convolution2D(9, (3, 3), activation=self.activation_function,
      padding='same', name='level0')(inputs)
2 x = Convolution2D(1, (1, 1), activation=self.activation_function,
      padding='same', name='level1')(x)
```

LISTING 5.2: Model structure with combination of features at the end.

The second one adds two additional layers to figure out the dependence of the network depth for the feature combination.

```
1 x = Convolution2D(1, (1, 1), activation=self.activation_function,
      padding='same', name='level0')(inputs)
2 x = Convolution2D(1, (3, 3), activation=self.activation_function,
      padding='same', name='level1')(x)
3 x = Convolution2D(5, (5, 5), activation=self.activation_function,
      padding='same', name='level2')(x)
4 x = Convolution2D(1, (3, 3), activation=self.activation_function,
      padding='same', name='level3')(x)
```

LISTING 5.3: Model structure with combination of features in the beginning with additional layers.

Next to the moment of feature combination, the moment when the input image is upscaled in some way may be relevant as presented in chapter 2. To evaluate this point of upscaling, another dataset has been introduced. In this dataset the input data is not interpolated to the size of the output image. It is kept in its initial sizes half as big in x and y direction compared to the output grid. Using this dataset the process of upsampling can be included into the model. Again, this has been done with three different models. The first model upsamples in the beginning as shown below.

```
1 x = UpSampling2D((2, 2))(inputs)
2 x = Convolution2D(1, (1, 1), activation=self.activation_function,
      padding='same', name='level0')(x)
3 x = Convolution2D(64, (9, 9), activation=self.activation_function,
      padding='same', name='level1')(x)
4 x = Convolution2D(32, (5, 5), activation=self.activation_function,
      padding='same', name='level2')(x)
5 x = Convolution2D(1, (5, 5), activation=self.activation_function,
      padding='same', name='level3')(x)
```

LISTING 5.4: Model structure with upscaling towards the doubled size in the beginning.

As first comparison, the upsampling process can be done as last step as well, as one can see below.

```
1 x = Convolution2D(1, (1, 1), activation=self.activation_function,
    padding='same', name='level0')(inputs)
2 x = Convolution2D(64, (9, 9), activation=self.activation_function,
    padding='same', name='level1')(x)
3 x = Convolution2D(32, (5, 5), activation=self.activation_function,
    padding='same', name='level2')(x)
4 x = Convolution2D(1, (5, 5), activation=self.activation_function,
    padding='same', name='level3')(x)
5 x = UpSampling2D((2, 2))(x)
```

LISTING 5.5: Model structure with upscaling towards the doubled size at the end.

In contrast to the feature combination, in this case also an upsampling in the in the middle of the network is considered, mainly due to the success of [Osendorfer, Soyer, and Smagt, 2014].

```
1 x = Convolution2D(1, (1, 1), activation=self.activation_function,
    padding='same', name='level0')(inputs)
2 x = Convolution2D(64, (9, 9), activation=self.activation_function,
    padding='same', name='level1')(x)
3 x = UpSampling2D((2, 2))(x)
4 x = Convolution2D(32, (5, 5), activation=self.activation_function,
    padding='same', name='level2')(x)
5 x = Convolution2D(1, (5, 5), activation=self.activation_function,
    padding='same', name='level3')(x)
```

LISTING 5.6: Model structure with upscaling towards the doubled size between the convolutional layers.

One main trend in the deep learning community has been towards much deeper networks, also for super-resolution, as shown in chapter 2. To understand the influence of the network depth in the special case of super resolution with limited data, two networks with higher depth have been evaluated. Starting with the following one.

```
1 x = UpSampling2D((2, 2))(inputs)
2 x = Convolution2D(1, (1, 1), activation=self.activation_function,
    padding='same', name='level0')(x)
3 x = Convolution2D(32, (9, 9), activation=self.activation_function,
    padding='same', name='level1')(x)
4 x = Convolution2D(64, (9, 9), activation=self.activation_function,
    padding='same', name='level2')(x)
5 x = Convolution2D(32, (9, 9), activation=self.activation_function,
    padding='same', name='level3')(x)
6 x = Convolution2D(16, (5, 5), activation=self.activation_function,
    padding='same', name='level4')(x)
```

```
7 x = Convolution2D(1, (5, 5), activation=self.activation_function,
    padding='same', name='level5')(x)
```

LISTING 5.7: Model structure of a deeper neural network.

Due to the limited size of the input data, additionally a version with smaller kernel sizes has been implemented. Using this, it is possible to reduce the number of parameters dramatically but at the same time keep a deeper neural network.

```
1 x = UpSampling2D((2, 2))(inputs)
2 x = Convolution2D(1, (1, 1), activation=self.activation_function,
    padding='same', name='level0')(x)
3 x = Convolution2D(32, (3, 3), activation=self.activation_function,
    padding='same', name='level1')(x)
4 x = Convolution2D(64, (5, 5), activation=self.activation_function,
    padding='same', name='level2')(x)
5 x = Convolution2D(32, (5, 5), activation=self.activation_function,
    padding='same', name='level3')(x)
6 x = Convolution2D(16, (5, 5), activation=self.activation_function,
    padding='same', name='level4')(x)
7 x = Convolution2D(1, (3, 3), activation=self.activation_function,
    padding='same', name='level5')(x)
```

LISTING 5.8: Model structure of a deeper neural network with small kernel sizes.

As presented in chapter 2 next to the architecture of the network, other parameters like the loss function may be relevant. To figure out the importance of the loss function and its interaction with the used activation function and the learning rate, one fixed model has been chosen. The model is based on [Dong et al., 2016], simply extended by an initial layer to combine the features of the DWD model output. After this combination, the grid is mapped into a high dimensional feature space, one non-linear operation is applied and the space gets mapped back towards the dimension of the target image. The model is shown below.

```
1 x = Convolution2D(1, (1, 1), activation=self.activation_function,
    padding='same', name='level0')(inputs)
2 x = Convolution2D(64, (9, 9), activation=self.activation_function,
    padding='same', name='level1')(x)
3 x = Convolution2D(32, (5, 5), activation=self.activation_function,
    padding='same', name='level2')(x)
4 x = Convolution2D(1, (5, 5), activation=self.activation_function,
    padding='same', name='level3')(x)
```

LISTING 5.9: Model structure of a simple neural network based on the model presented by [Dong et al., 2016

As listed in 5.1, the parameters varied for this experiment are loss function, activation function and learning rate. Possible values for the loss function are mean squared error (mse) and mean absolute error (mae). Peak signal to noise ratio $(psnr)$ is ignored in this case because it is computed based on *mse* and has the drawback of producing possibly infinite values. As activation functions $relu$ and $tanh$ are considered and the learning rate is varied from 1e-2 towards 1e-4.

For all these experiments a number of parameters are kept constant. An overview of these parameters is presented in 5.2:

TABLE 5.2: Static parameters of all experiments.

Parameter	Value
Number of features	9
Size of the image in x direction	82
Size of the image in y direction	52
Number of epochs	50
Batch size	16
Number of output channel	1
Leadtime	11, 12, 13

The number of features describes the number of input variables that are considered for each spatial data point, while the number of output channels is describing the same dimensionality for the output image. The sizes in x and y direction are describing the spatial dimensions of the target image. Number of epochs and batch size are hyper parameters for the training process, defining how long the model is trained and how many grids are grouped for each batch. Finally, the leadtime describes the number of hours the model should predict into the future or more exactly which predictions of the model should be chosen.

5.3 Results

Conducting these experiments created many quite unambiguous results. In general, it seems like the assumption that the task is learnable given the available data does not hold. To train neural networks the correct choice of a loss function and especially the minimization of this loss is fundamental [Li et al., 2017]. Even though the correct choice of loss functions is not intuitive for natural image super-resolution tasks, as presented in chapter 2, the problem

is in the most cases not the ability of the network to learn a mapping. The problem is rather that this mapping does not match the human perception. In the case of grid based weather data the second aspect is not as important but unfortunately even the minimization of loss in general is not possible, as one can see in figures 5.1 and 5.2.

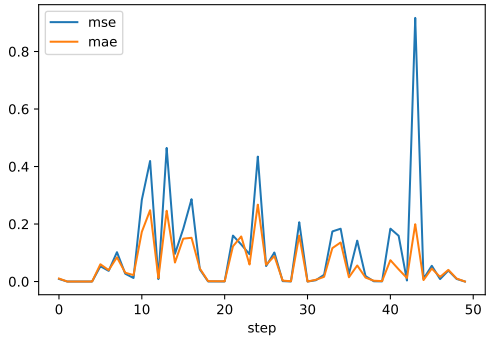


FIGURE 5.1: Mean squared error and mean absolute error presented exemplary for one training process.

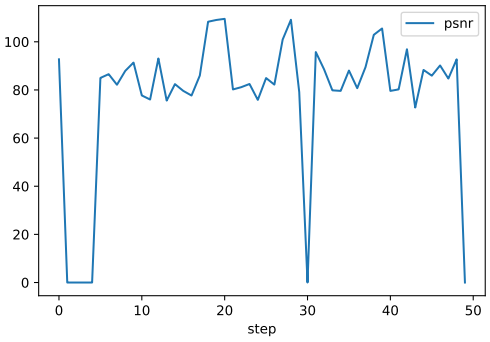


FIGURE 5.2: Peak signal to noise ratio presented exemplary for one training process.

The figures are representing the metrics saved for the training processes, that are quite similar for all experiments presented in the chapter before. In figure 5.1 mean squared error (mse) and mean absolute error (mae) are shown for 50 epochs. Figure 5.2 in comparison is representing the peak signal to noise ratio $(psnr)$, a widely used metric for super-resolution processes. In comparison to *mse* and *mae*, *psnr* is only used for comparison reasons and not as loss function.

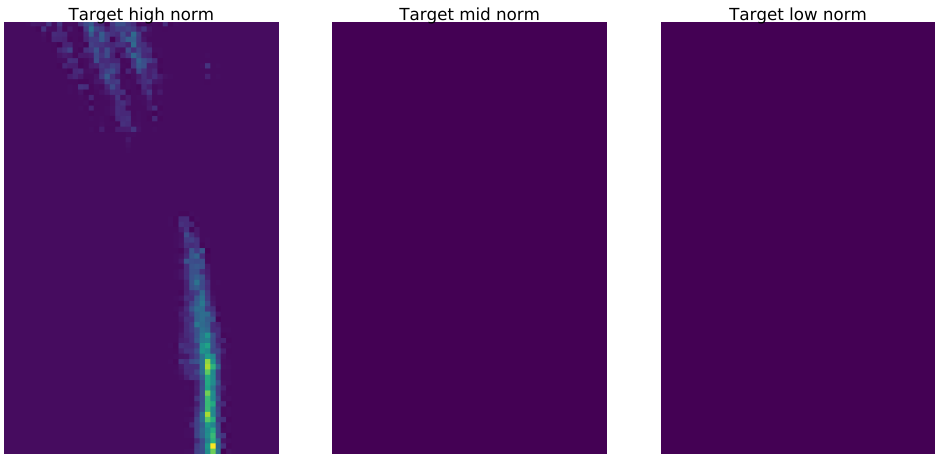


FIGURE 5.3: Example of target data with high, mid and low norm.

To understand the problems of this algorithm, the structure of the target data offers some insights. In figure 5.3 a selection of target images is presented, selected by the height of the summed up values in the corresponding base image. On the right side the image with the lowest sum is shown. Not that surprisingly, the image shows close to zero rain. In the middle and on the left size, the amount of rain should be higher, but as one can see the image with an average norm has close to zero rain measured for the Harz area. This differs in the left image with the highest norm, but still the amount of rain is quite sparse (Brighter values are representing a higher amount of rainfall in cm).

Sparse data like this is not easily learnable, because it is possible to get good values for the loss function by simply predicting no rain by default. This behavior can be seen by most of the networks. Two examples are shown below, figure 5.4 for the initial dataset and figure 5.5 with the same experimental setup except that the input data has been normalized.

The target images of figure 5.4 and figure 5.5 differ, because the sum is computed based on the sum of the input data and after the feature wise normalization some features are no longer as dominant as before (Example of the value differences can be seen in chapter 4).

Nearly none of the other experiments offers more insight, if some artifacts are ignored. Plots like the ones in figure 5.4 or 5.5 are listed in the appendix

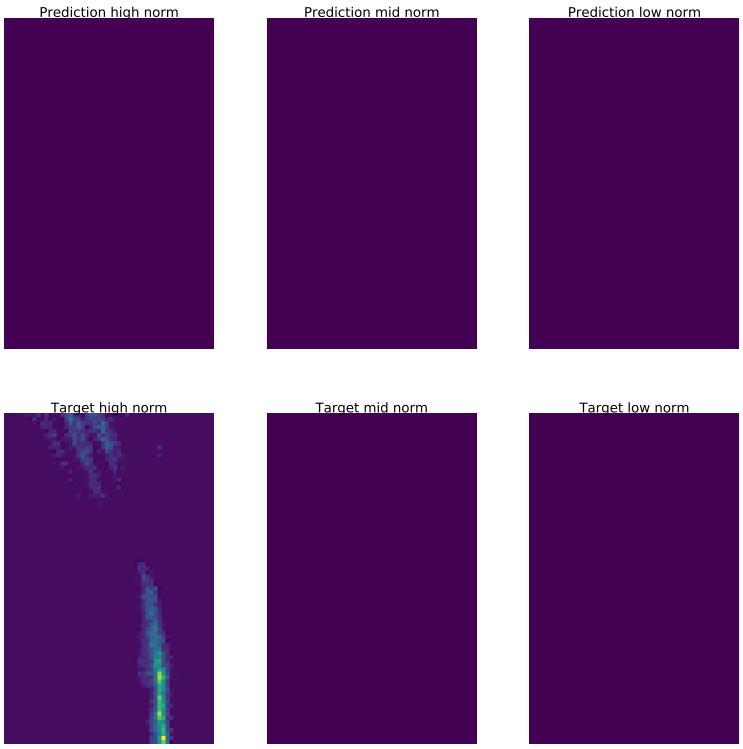


FIGURE 5.4: Predictions and target data for the deeper neural network with smaller kernel.

for all conducted experiments. One group of experiments however is differing from the the shown examples quite obviously, as one can see in figure 5.6.

This behavior can be seen in eight of forty experiments. All of them are done with the dataset composed of same sized grids and the simple convolutional network. But more important, they are all using *tanh* as activation function, independent of the loss function and for different learning rates. Unfortunately it is not possible to explain the behavior only based on the activation function because there are experiments left that are using *tanh* without this behavior as one can see in figure 5.7.

But if one combines activation function and learning rate the behavior may be explainable. The greenish artifact dominated images are only predicted by *tanh* networks with small learning rates (below 1e-2). According to that, the behavior may be based a slower training process of networks with

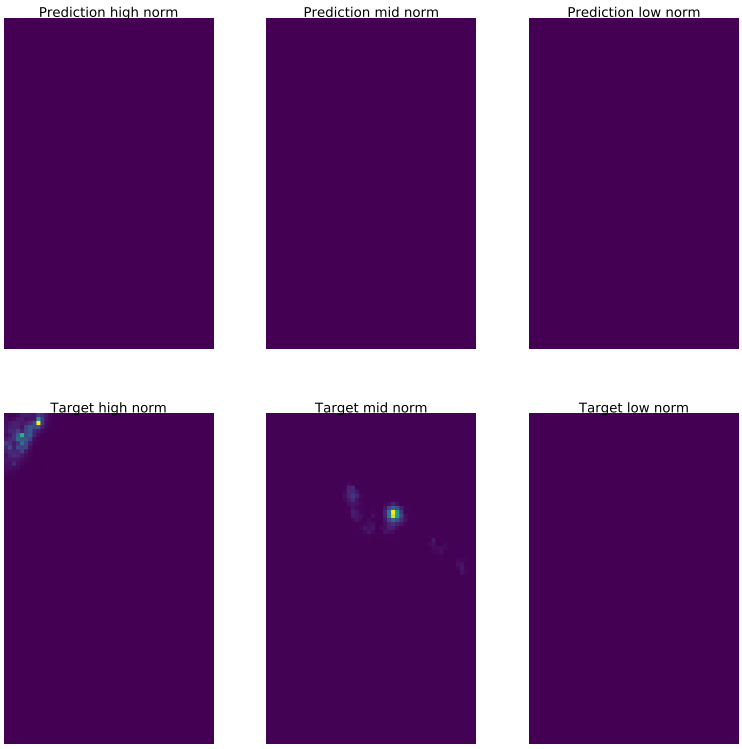


FIGURE 5.5: Predictions and target data for the deeper neural network with smaller kernel on normalized data.

tanh as activation function and smaller learning rates.

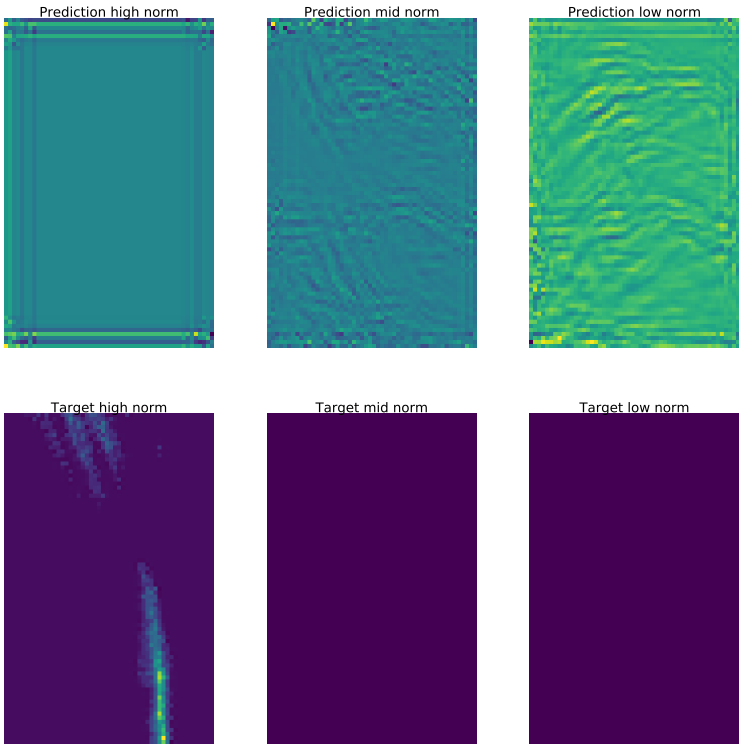


FIGURE 5.6: Predictions and target data for the simple cnn with tanh as activation function, mse as loss function and a learning rate of 1e-3.

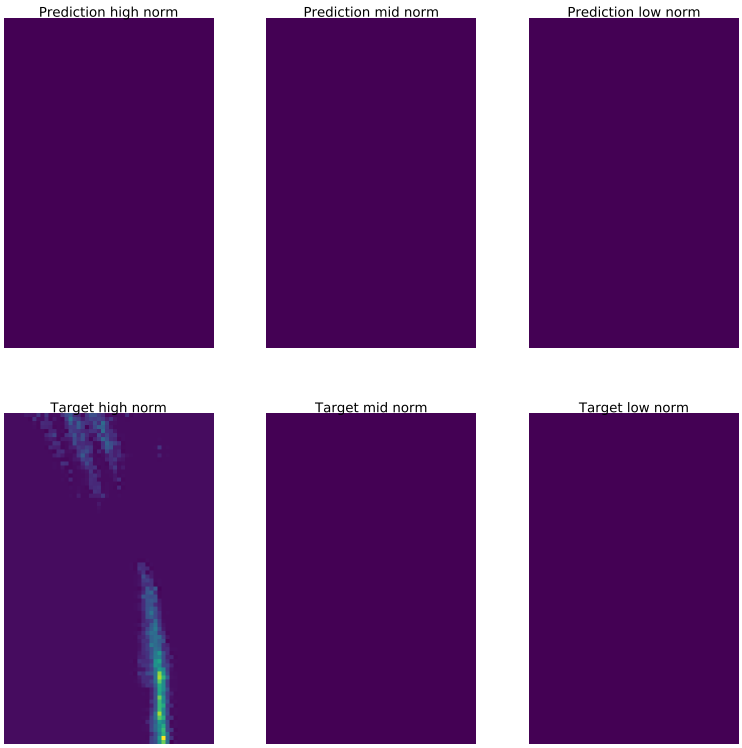


FIGURE 5.7: Predictions and target data for the simple cnn with tanh as activation function, mse as loss funtion and a learning rate of 1e-2.

Chapter 6

Conclusion

The approach presented in this thesis is based on the reliability of radar data. But as presented by [Toth, Brath, and Montanari, 2000], radar detection is especially complicated in mountain regions. In combination with the limitations presented by the DWD itself (listed in chapter 2), the reliability of radar data as ground truth can be put in doubt. But even apart from that, the combination of just 728 data points with really sparse data makes it quite hard to predict precipitation events, as shown in chapter 5. Additionally, the task that should be solved is even a lot more complex than upscaling of natural images, because next to the super-resolution itself the features of the model need to be combined in a meaningful way to generate cm values for the amount of fallen rain. This combination makes it very hard to learn the task. Due to this it plausible that the best solution for the networks is to just predict zeros. For other results, more information need to be added. Apart from just increasing the amount of data, it may be helpful to add orography or feed forward layers to learn characteristics of each point out of a spatial perspective. If orography information is added even augmentation like rotations may be meaningful, because the network is not longer dependent on the specific position of each grid point. In addition to using more months as input data, next to 12 hours the leadtimes 11 and 13 can be considered, because the prediction-differences of COSMO-DE-EPS due to varied leadtimes may be small enough to not confuse the network. Next to this, currently only the nine most important features are used, so that an increased number of features may be an opportunity as well. If enough space on the ikw-grid is available or the Jülich system can be used, an integration of all ensemble members may worth a shot. Next to the datasource, the network structures can be improved using most of the concepts developed for image classification in the past years. One of the most interesting aspects may be recursion, because it increases the kernel size indirectly without adding more parameters and skip connections. Independent of currently well-known techniques,

it may be promising to use convolutions with a stride of 1/2 instead of simple upsampling. If neural networks are not longer a requirement, random forest as presented by [Im et al., 2016] may be worthwhile. As already presented in chapter 2.2.3, it may be an interesting idea to overcome the problems of grid based radar data by learning directly on the point measurements of rain gauge data. To allow this kind of mapping from grid based data towards point measures gaussian processes or attentive neural processes are a promising idea [Berrocal, Gelfand, and Holland, 2010, Kim et al., 2019]. But for the next step one should think about starting with a simpler task like temperature prediction in plain areas, to make sure, that the used architectures are working correctly. If this works out, a gradual difficulty increasement can be done by first switching to rain forecasting in a plain area and finally trying the complex task of rain forecasting in areas with many height differences like the Harz.

Appendix A

Appendix

A.1 Theoretical Foundations

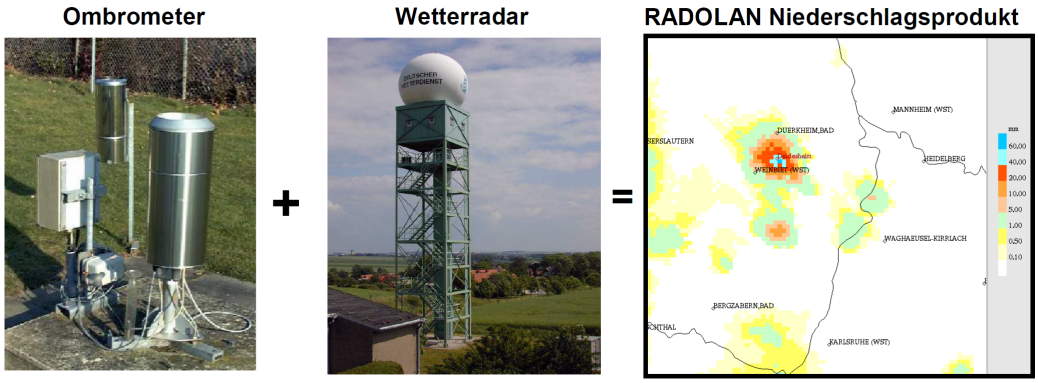


FIGURE A.1: Combination of rada data and obrometer data exemplary, based on [DWD, 2018].

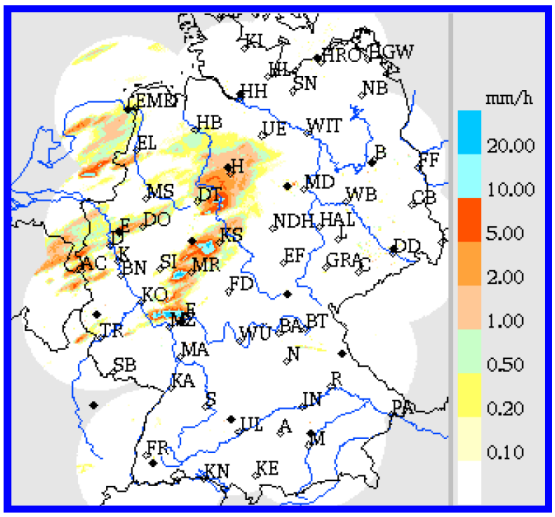


FIGURE A.2: Combination of rada data and obrometer data for whole Germany, based on [DWD, 2018].

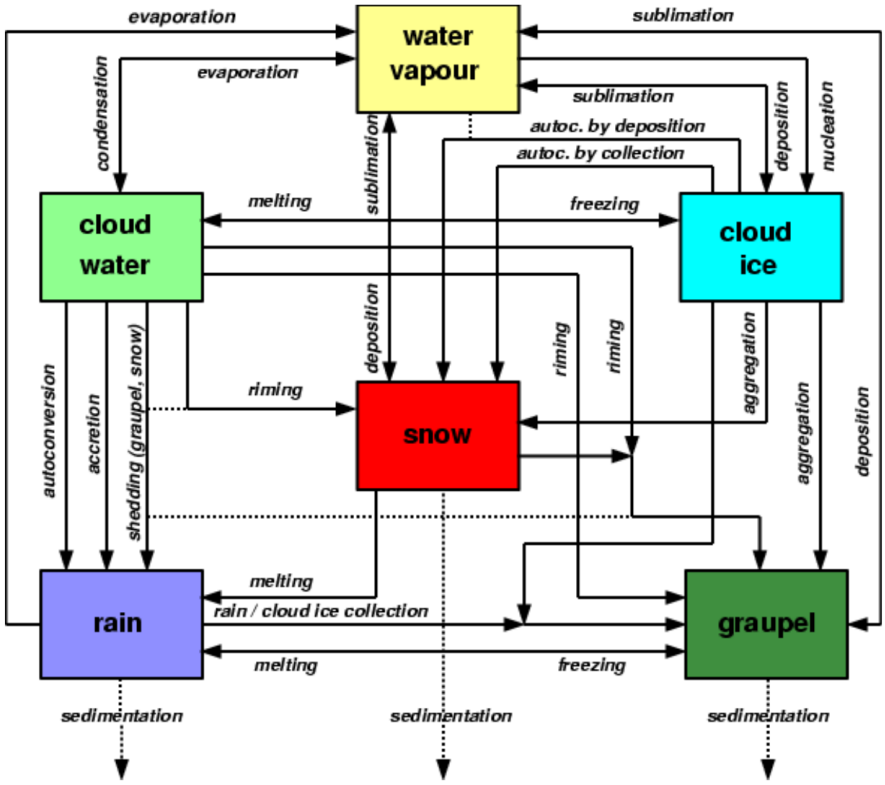


FIGURE A.3: Precipitation dynamics modeled by COSMO-DE, based on [Baldauf, Klink, and Reinhardt, 2016].

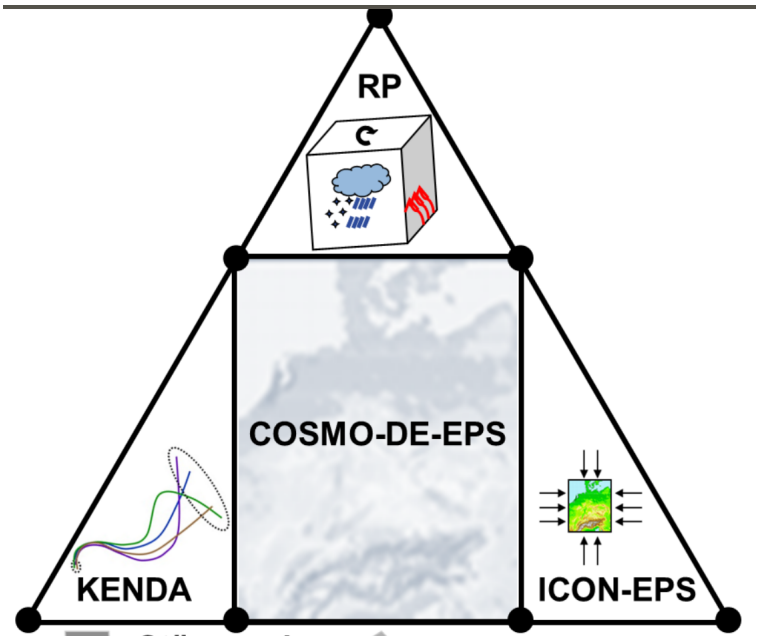
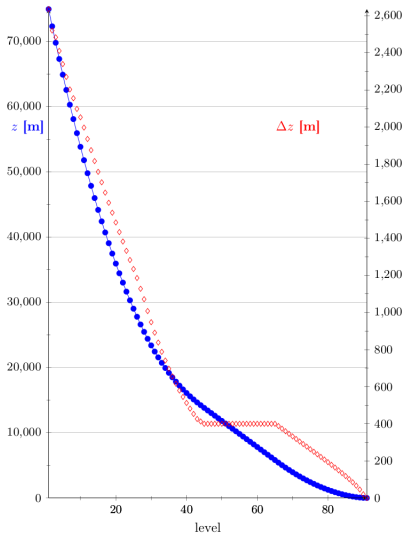


FIGURE A.4: Varied starting conditions used for generation of COSMO-DE-EPS ensemble model, based on [Theis et al., 2017].



level	$[m]$	$[Pa]$
1	75 000	2.1
5	64 946	10.0
10	53 878	46.3
15	44 198	158.8
20	35 958	487.2
25	29 039	1 355.0
30	23 409	3 211.8
35	19 202	6 209.4
40	16 108	10 113.6
45	13 822	14 504.3
50	11 822	19 882.1
55	9 822	27 166.6
60	7 822	36 528.6
65	5 822	48 347.1
70	3 954	62 009.2
75	2 432	75 325.6
80	1 255	87 126.2
85	436	96 190.0
90	20	101 085.0

FIGURE A.5: Distribution and height of vertical ICON layer, based on [Reinert et al., 2019].

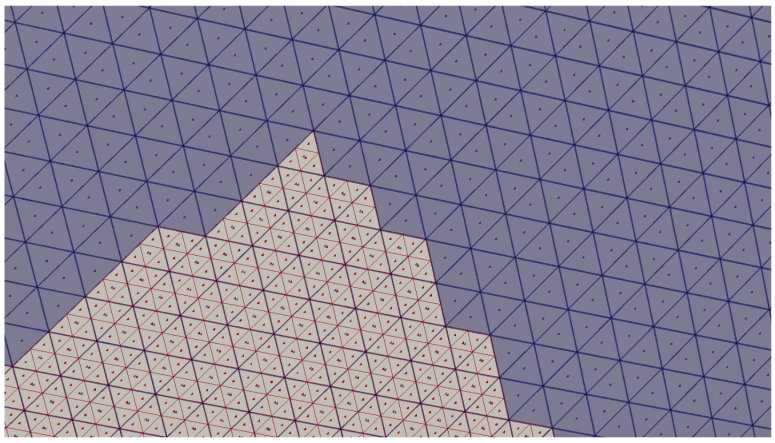


FIGURE A.6: Refinement of the ICON grid in Europe using bisection, based on [Reinert et al., 2019].

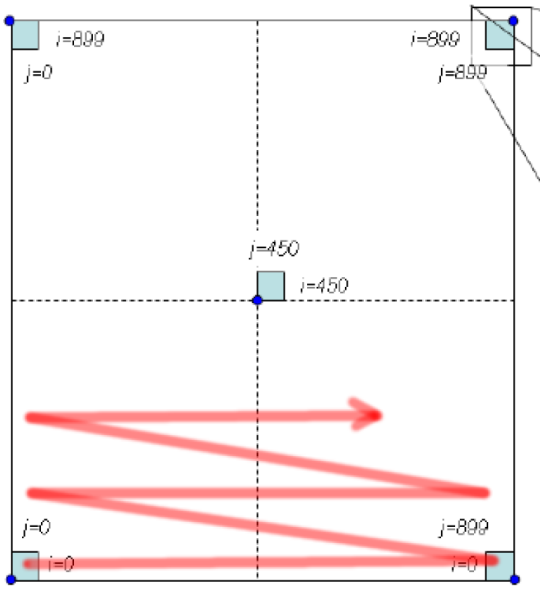


FIGURE A.7: Grid of daily RADOLAN data, based on [DWD, 2018].

A.2 Implementation and experiments

A.2.1 COSMO-DE-EPS available parameter

TABLE A.1: Parameter simulated by COSMO-DE-EPS.

Shortcut	Description
CAPE_ML	Convective Available Potential Energy, mean layer
r	No description available
HTOP_SC	Cloud top above msl, shallow convection
CLCH	Cloud Cover (0 - 400 hPa)
t	Temperature
eva	Evaporation
ALHFL_S	Latent Heat Net Flux (m)
prmsl	No description available
lssrwe	No description available
TQC	Total Column-Integrated Cloud Water
PRR_GSP	Large scale rain rate
TDIV_HUM	Vertical integral of divergence of total water content (s)
CIN_ML	Convective Inhibition, mean layer
2t	No description available
TQI	Total Column-Integrated Cloud Ice
tcols	No description available
SDI_1	Supercell detection index 1 (rot. up+down drafts)
T_SNOW	Snow temperature (top of snow)
TWATER	Total Column integrated water (all components incl. precipitation)
CLCL	Cloud Cover (800 hPa - Soil)
ATHB_S	Net long wave radiation flux
z	No description available
TQG	Total column integrated grauple
TCH	Turbulent transfer coefficient for heat (and Moisture)
W_SO_ICE	Soil ice content (multilayers)
ASOB_T	Net short wave radiation flux
AVMFL_S	Momentum Flux, V-Component (m)
2r	No description available
tp	No description available
u	Zonal wind in 850 hPa
AUMFL_S	Momentum Flux, U-Component (m)

Shortcut	Description
HTOP_DC	Height of top of dry convection above MSL
PRG_GSP	Graupel (snow pellets) precipitation rate
w	Vertical wind in 500 hPa
mn2t6	No description available
VMAX_10M	Maximum Wind 10m
ceil	Ceiling
ASWDIR_S	Downward direct short wave radiation flux at surface
sde	No description available
HBAS_SC	Cloud base above msl, shallow convection
ASOB_S	Net short wave radiation flux (at the surface)
rsn	No description available
RAIN_GSP	Large scale rain (Accumulation)
sd	No description available
tcolr	No description available
cnwat	No description available
APAB_S	Photosynthetically active radiation (m) (at the surface)
GRAU_GSP	Graupel (snow pellets) precipitation (Accumulation)
cd	No description available
tciwv	No description available
W_SO	Column-integrated Soil Moisture (multilayers)
10v	No description available
SDI_2	Supercell detection index 2 (only rot. up drafts)
SNOW_GSP	Large-Scale snowfall - water equivalent (Accumulation)
ATHB_T	Net long wave radiation flux
al	No description available
ASWDIFU_S	Upward diffusive short wave radiation flux at surface
ASHFL_S	Sensible Heat Net Flux (m)
mx2t6	No description available
FRESHSNW	Fresh snow factor
v	Meridional wind in 850 hPa
10u	No description available
CLCM	Cloud Cover (400 - 800 hPa)
ASWDIFD_S	Downward diffusive short wave radiation flux at surface
sp	Wind speed (SP)
c	No description available
QV_S	Specific Humidity (S)

Shortcut	Description
sr	No description available
CLCT	Total Cloud Cover

A.2.2 Example predictions of conducted experiments

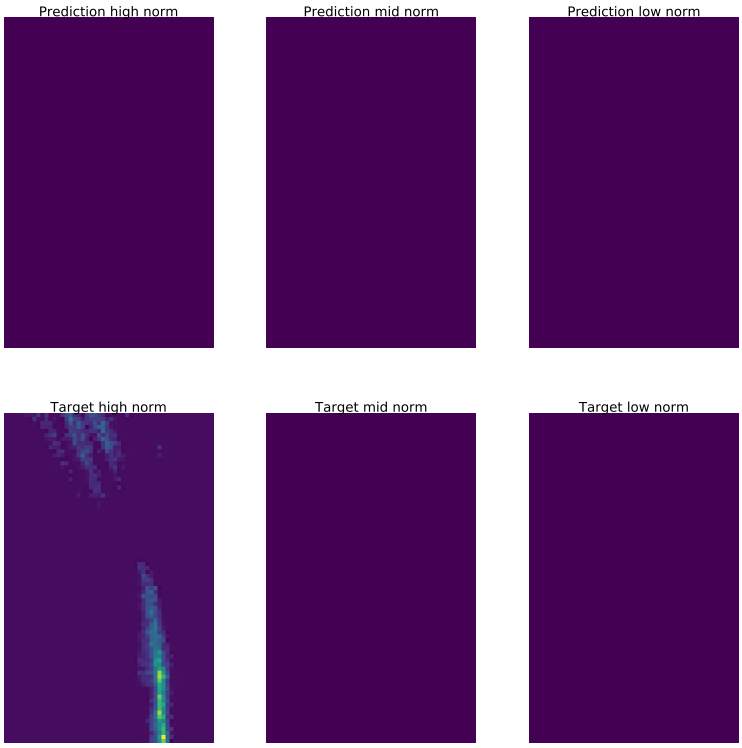


FIGURE A.8: Predictions and target data for neural network combining features last.

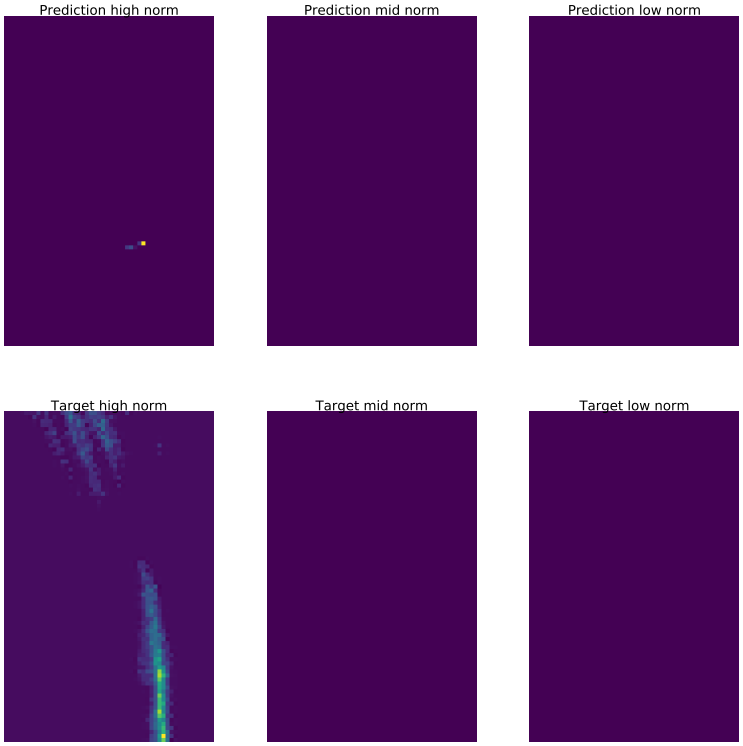


FIGURE A.9: Predictions and target data for neural network combining features first.

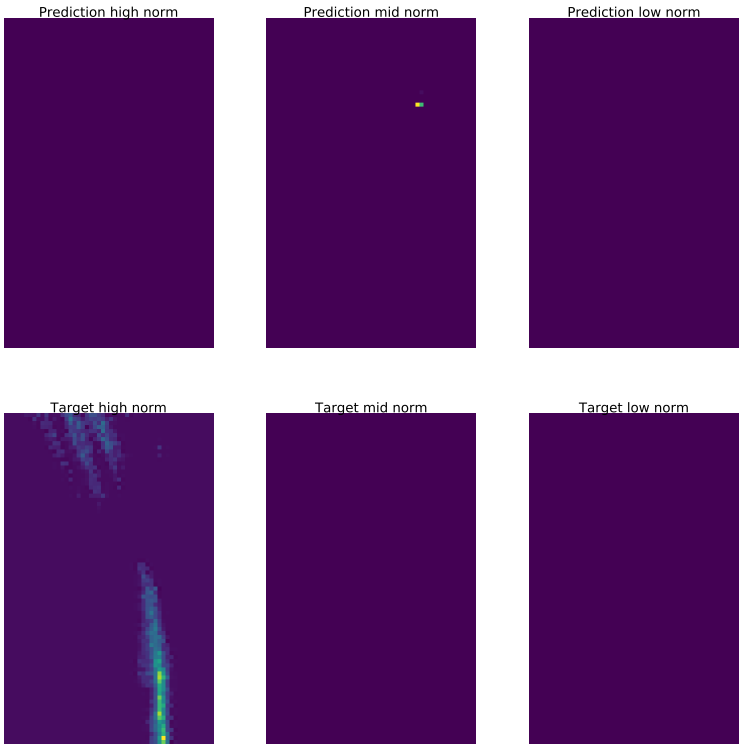


FIGURE A.10: Predictions and target data for neural network combining features first extended version.

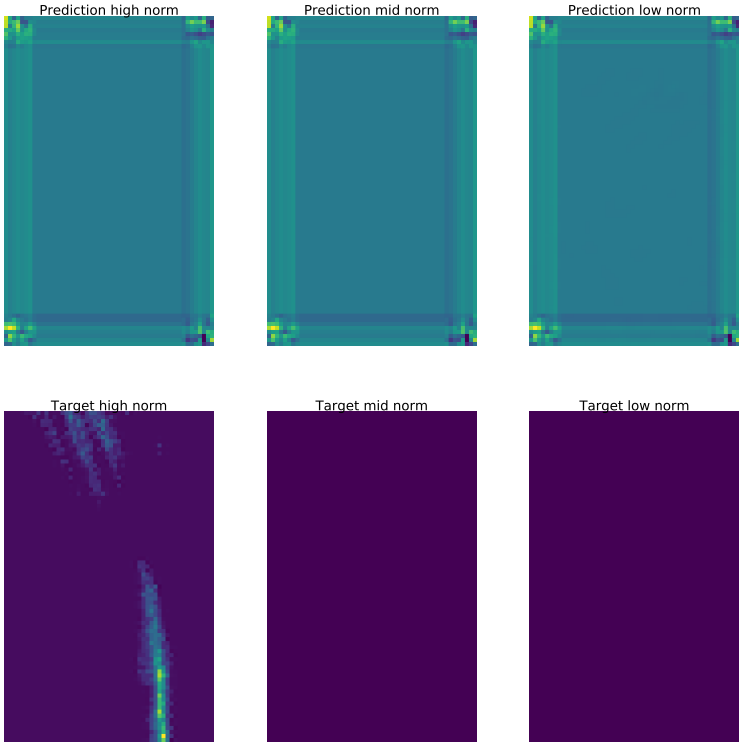


FIGURE A.11: Predictions and target data for simple neural network using mse, tanh and a learning rate of 1e-4.

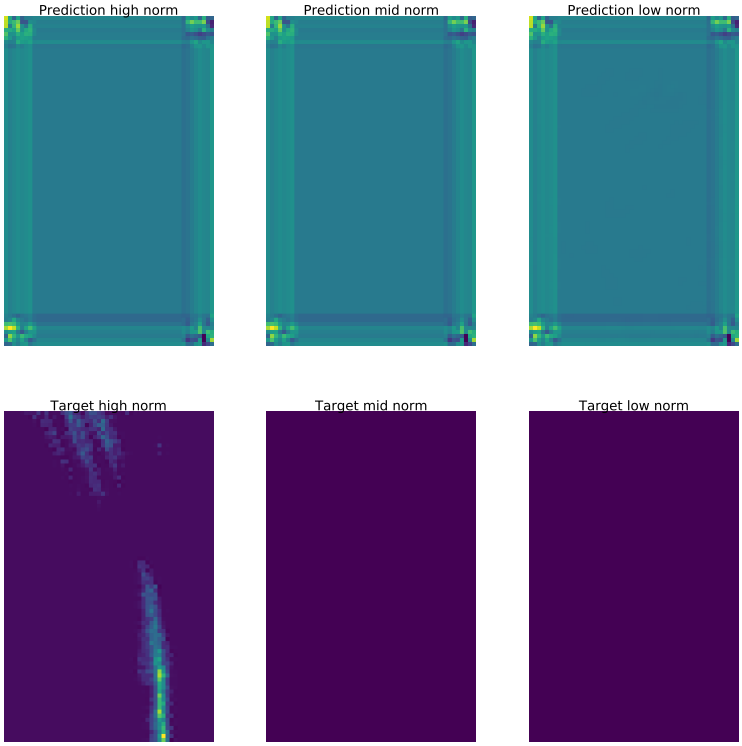


FIGURE A.12: Predictions and target data for simple neural network using mse, tanh and a learning rate of 1e-3.

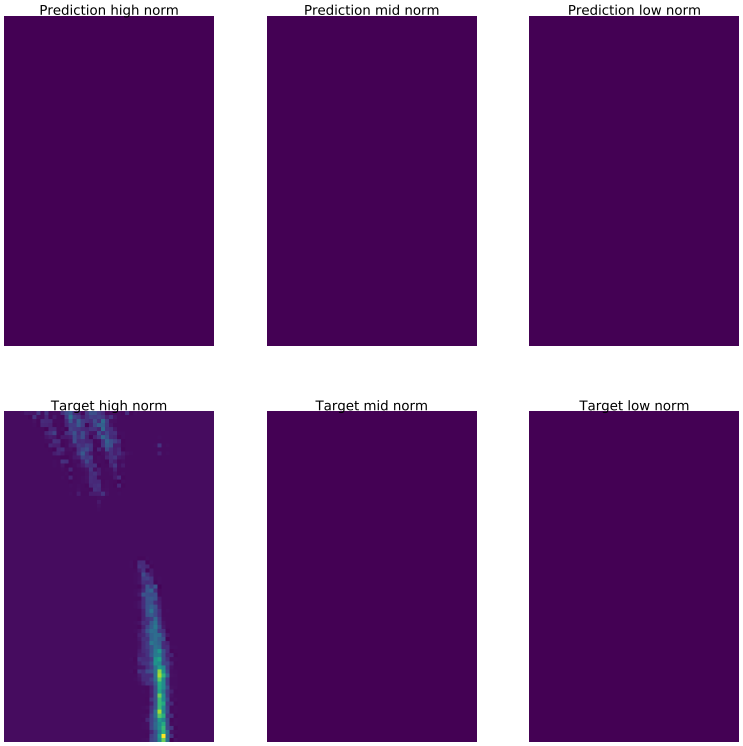


FIGURE A.13: Predictions and target data for simple neural network using mse, tanh and a learning rate of 1e-2.

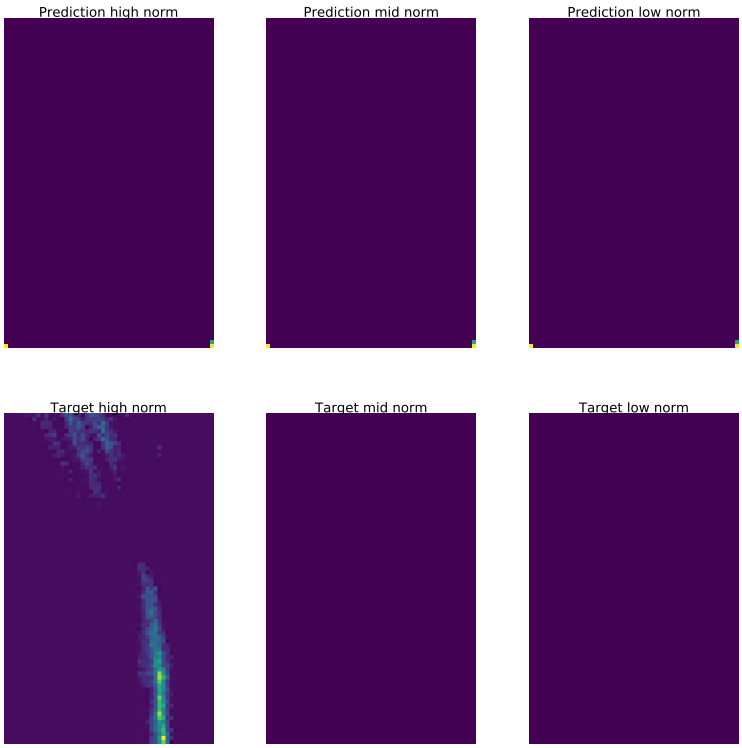


FIGURE A.14: Predictions and target data for simple neural network using mse, relu and a learning rate of 1e-4.

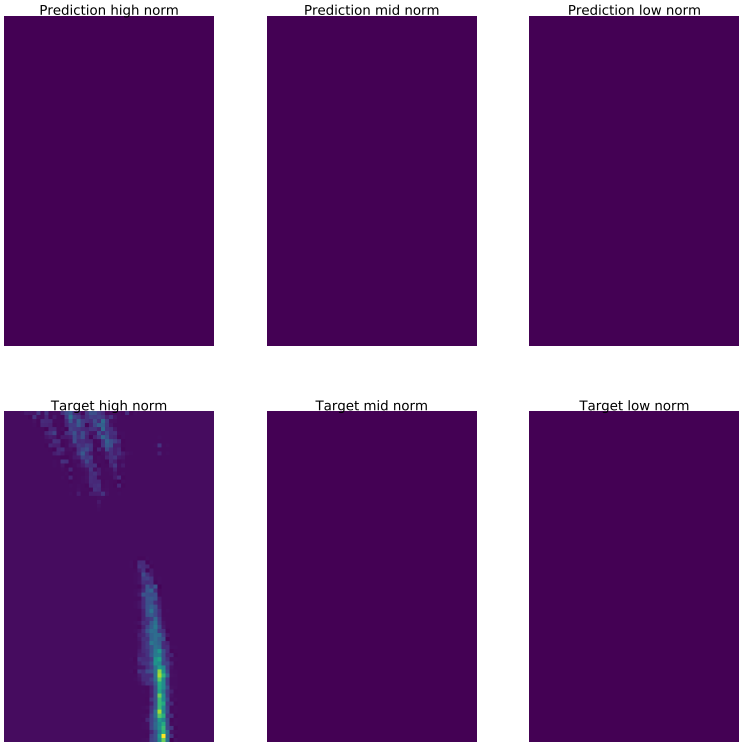


FIGURE A.15: Predictions and target data for simple neural network using mse, relu and a learning rate of 1e-3.

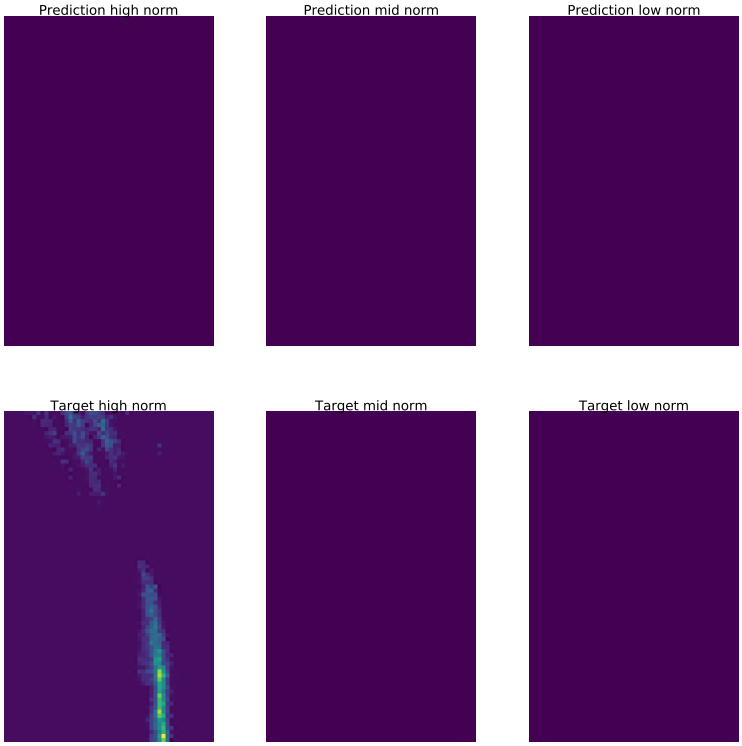


FIGURE A.16: Predictions and target data for simple neural network using mse, relu and a learning rate of 1e-2.

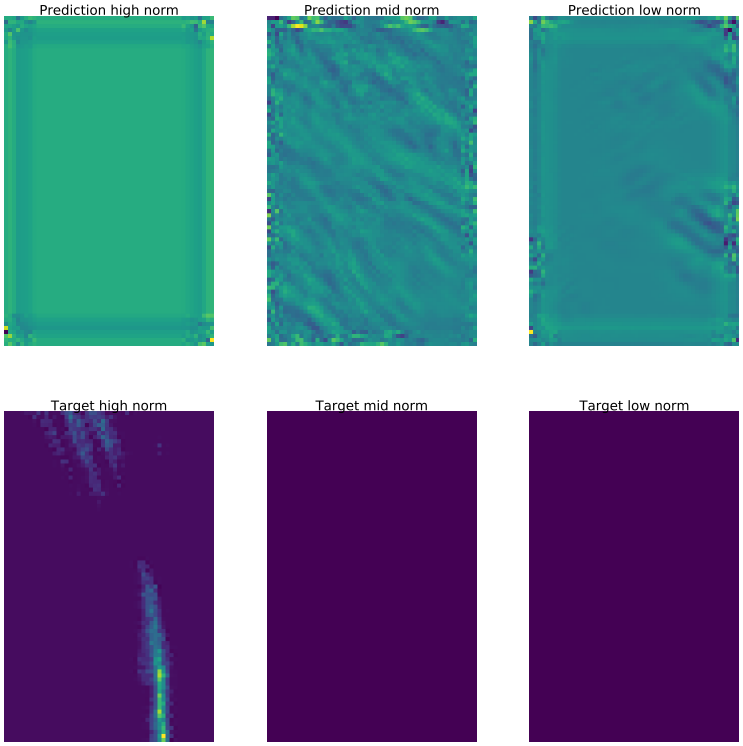


FIGURE A.17: Predictions and target data for simple neural network using mae, tanh and a learning rate of 1e-4.

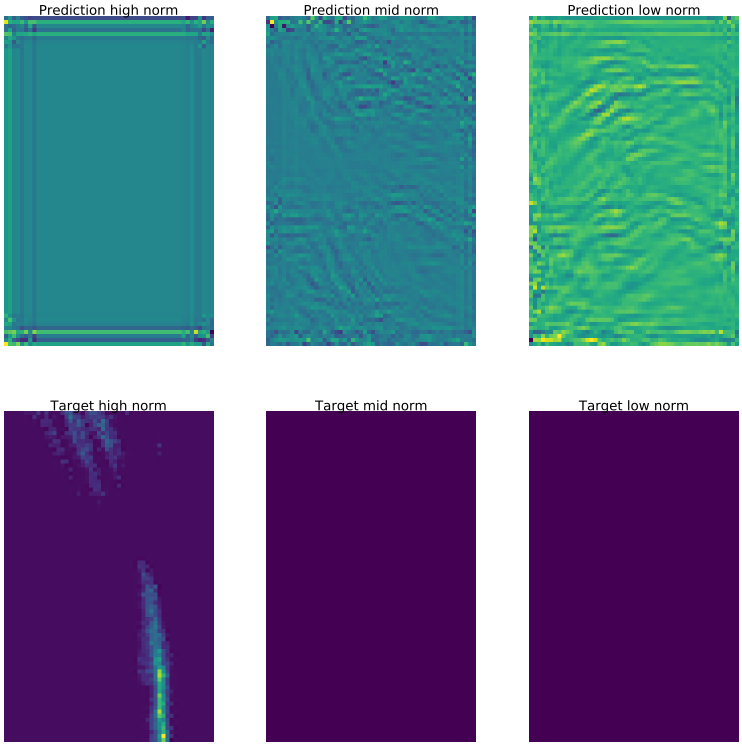


FIGURE A.18: Predictions and target data for simple neural network using mae, tanh and a learning rate of 1e-3.

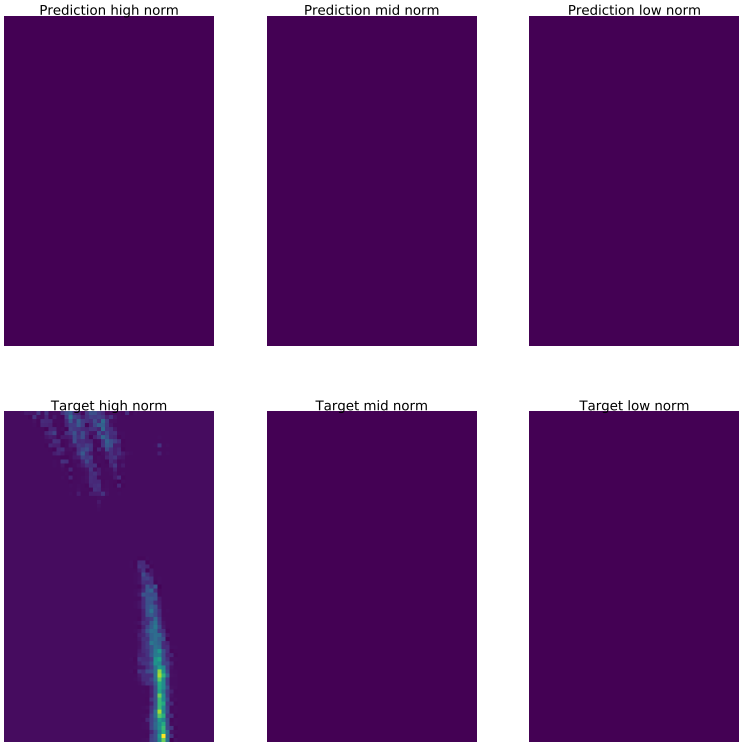


FIGURE A.19: Predictions and target data for simple neural network using mae, tanh and a learning rate of 1e-2.

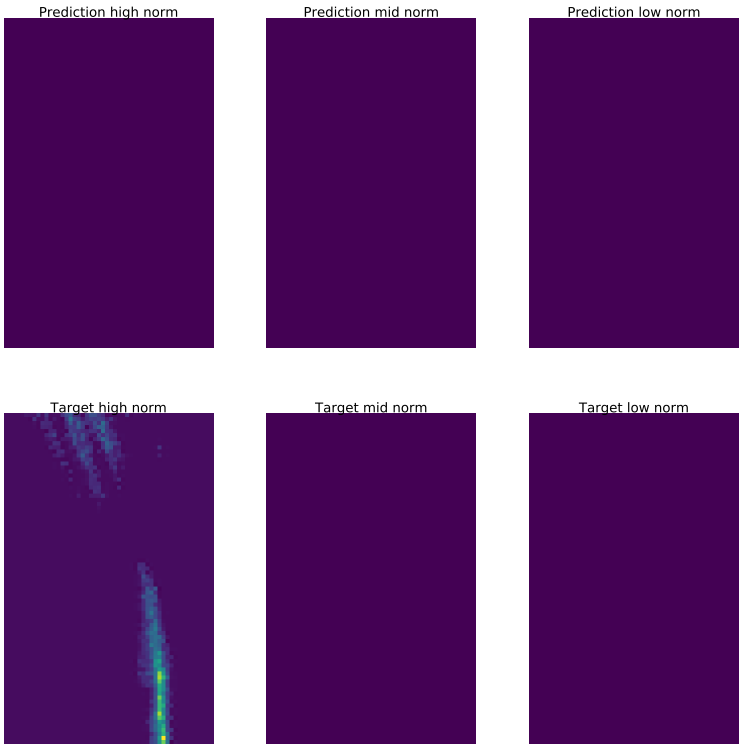


FIGURE A.20: Predictions and target data for simple neural network using mae, relu and a learning rate of 1e-4.

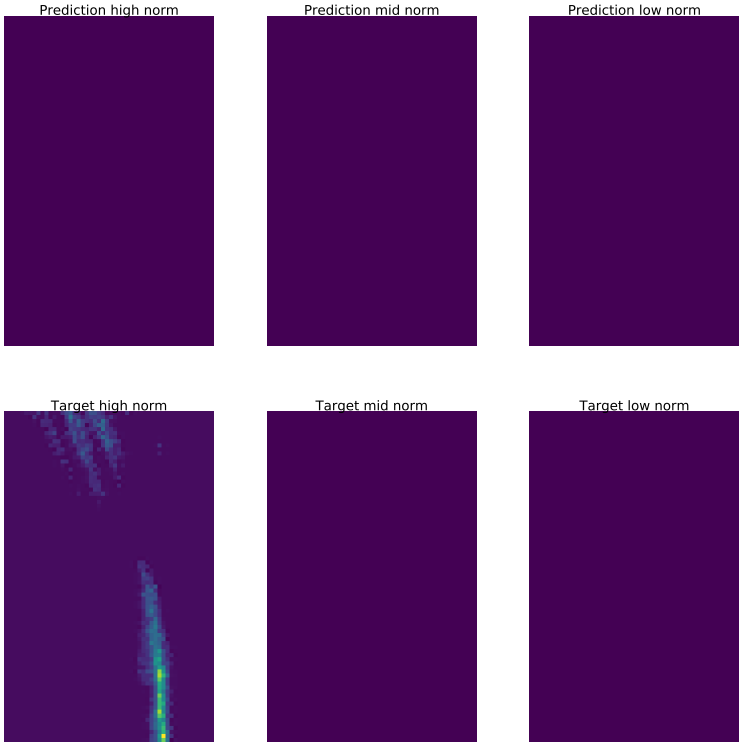


FIGURE A.21: Predictions and target data for simple neural network using mae, relu and a learning rate of 1e-3.

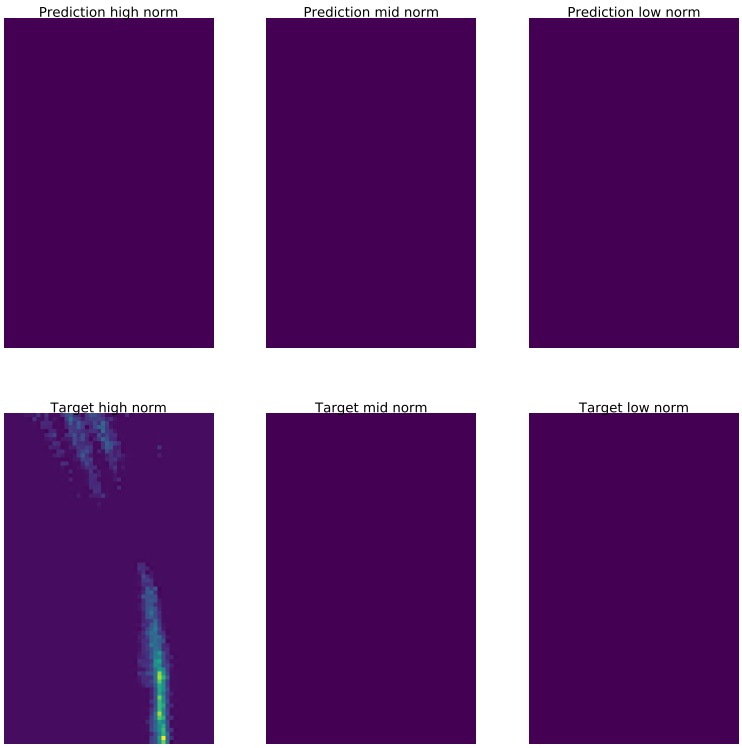


FIGURE A.22: Predictions and target data for simple neural network using mae, relu and a learning rate of 1e-2.

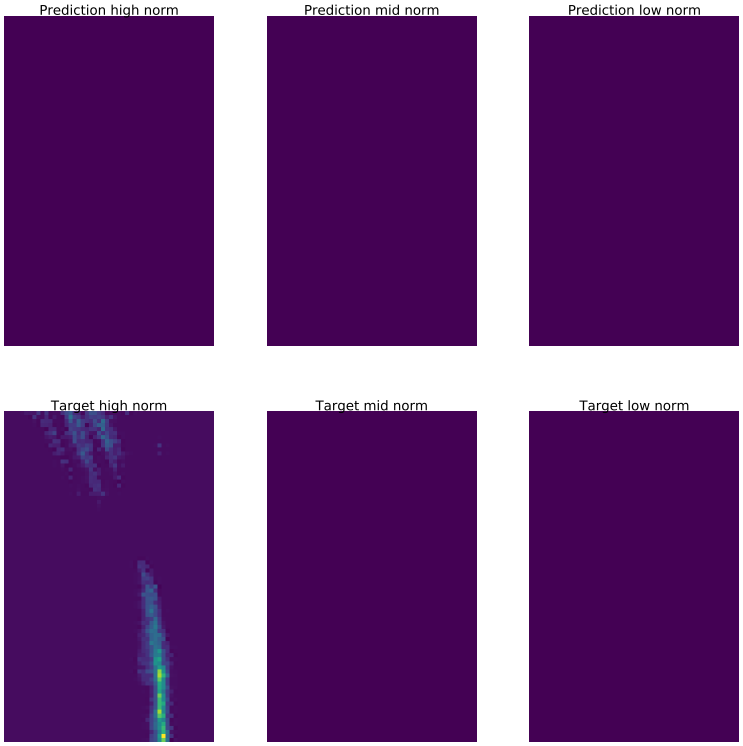


FIGURE A.23: Predictions and target data for convolutional neural network with upsampling in the middle.

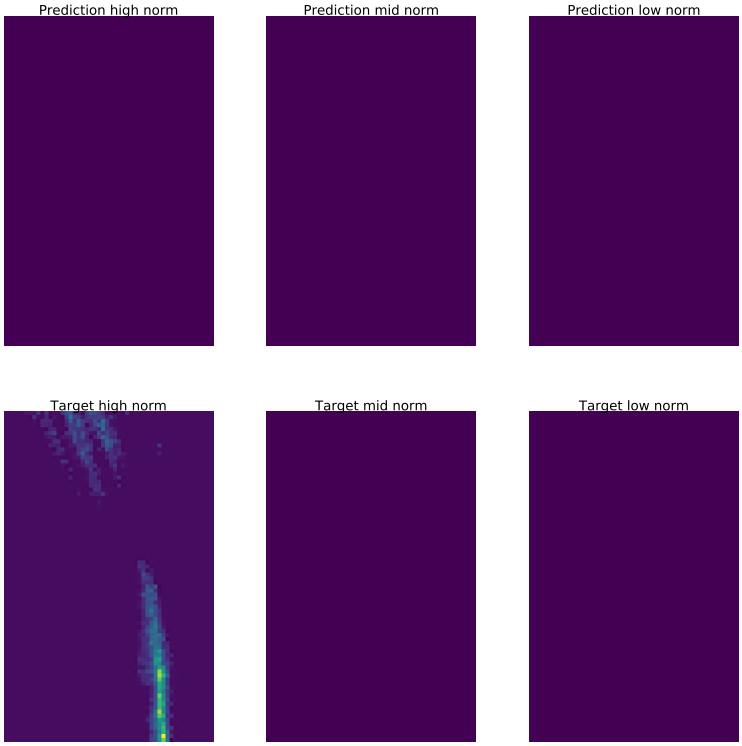


FIGURE A.24: Predictions and target data for convolutional neural network with upsampling at the end.

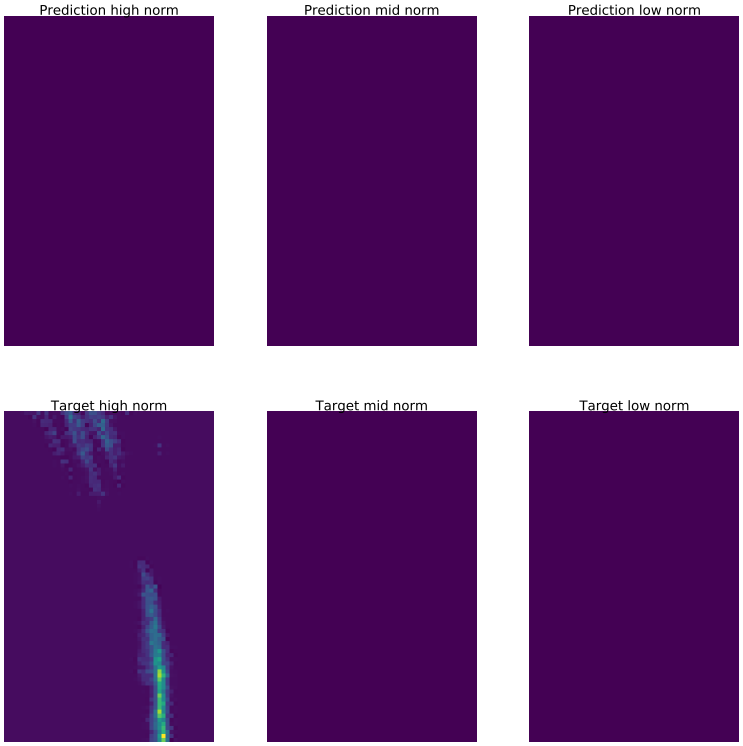


FIGURE A.25: Predictions and target data for convolutional neural network with upsampling at the beginning.

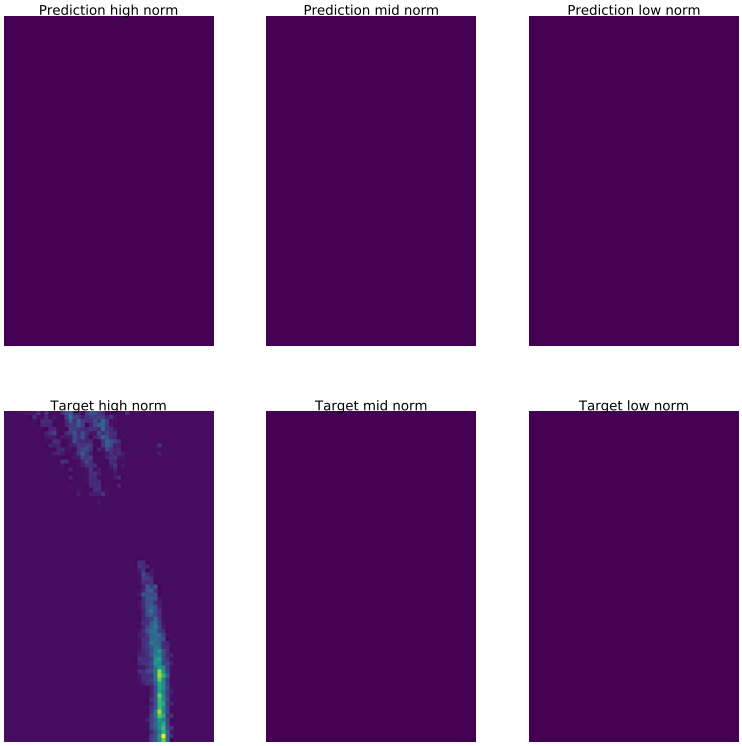


FIGURE A.26: Predictions and target data for deeper convolutional neural network.

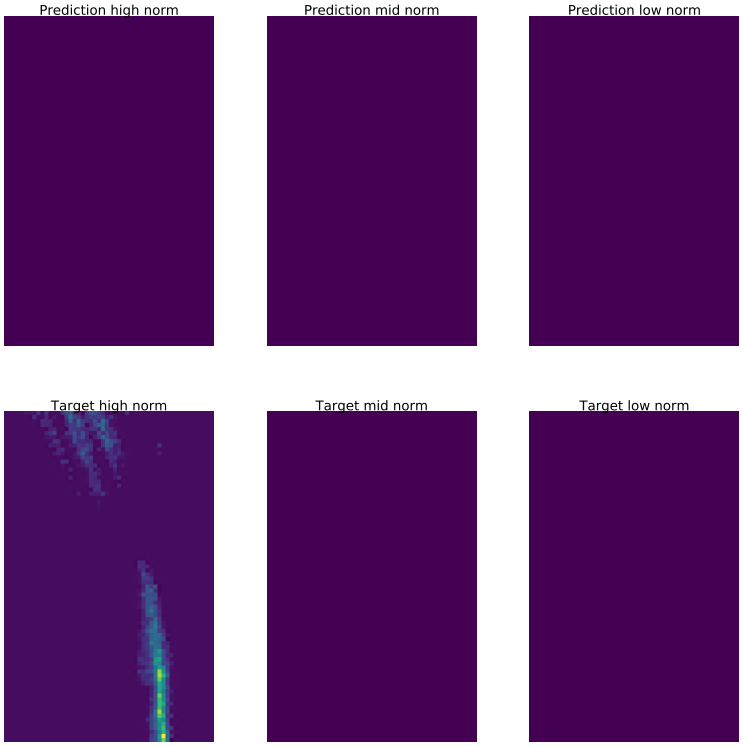


FIGURE A.27: Predictions and target data for deeper convolutional neural network with smaller kernel.

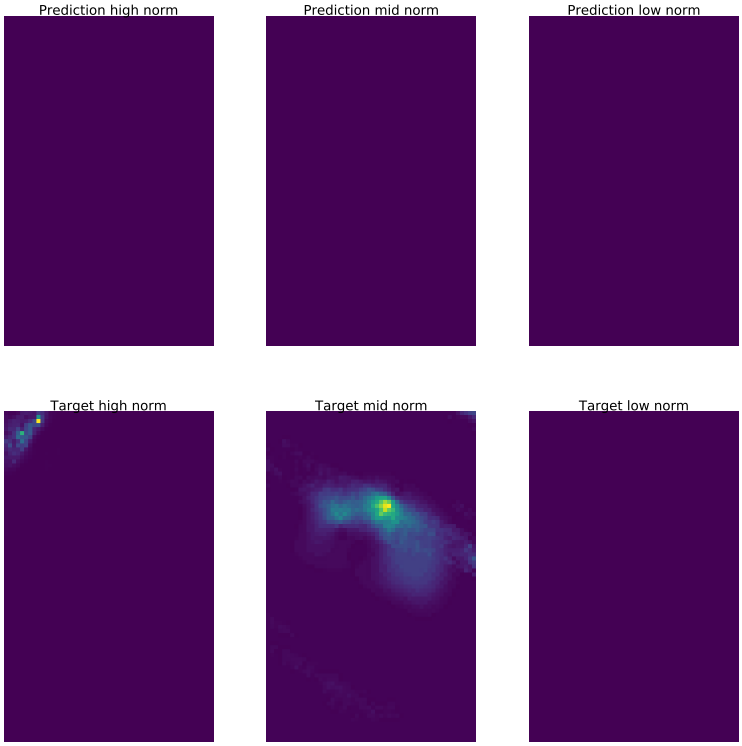


FIGURE A.28: Predictions and target data for neural network combining features last normalized.

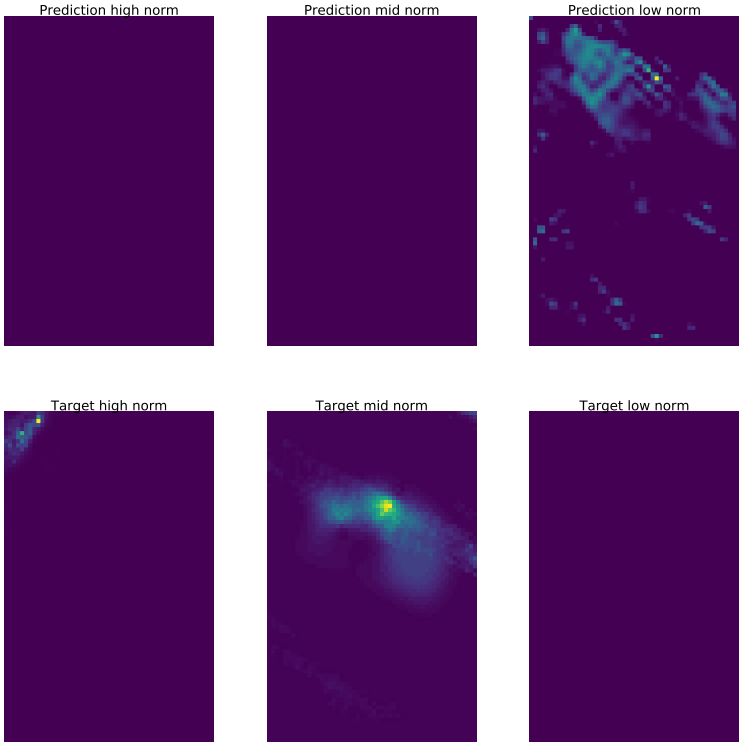


FIGURE A.29: Predictions and target data for neural network combining features first normalized.

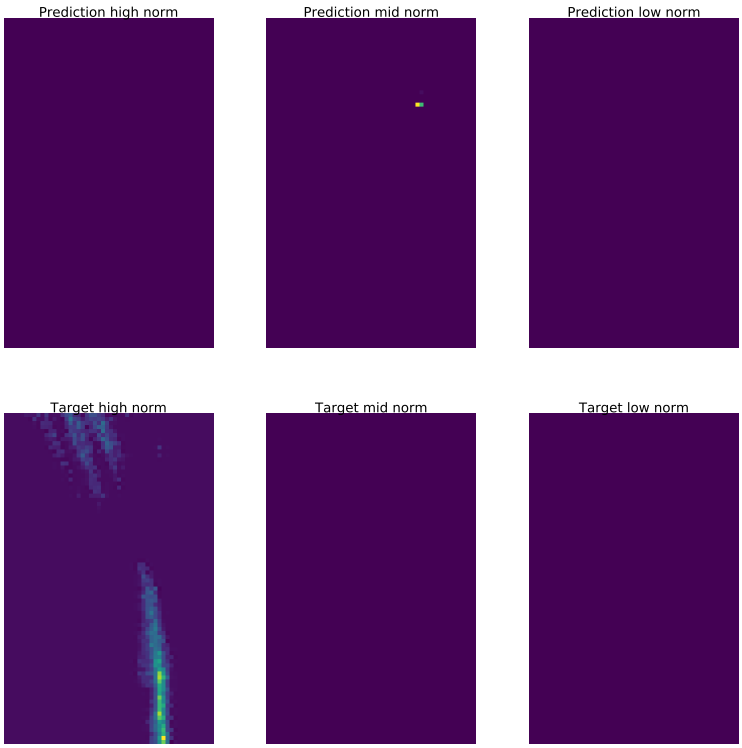


FIGURE A.30: Predictions and target data for neural network combining features first extended version normalized.

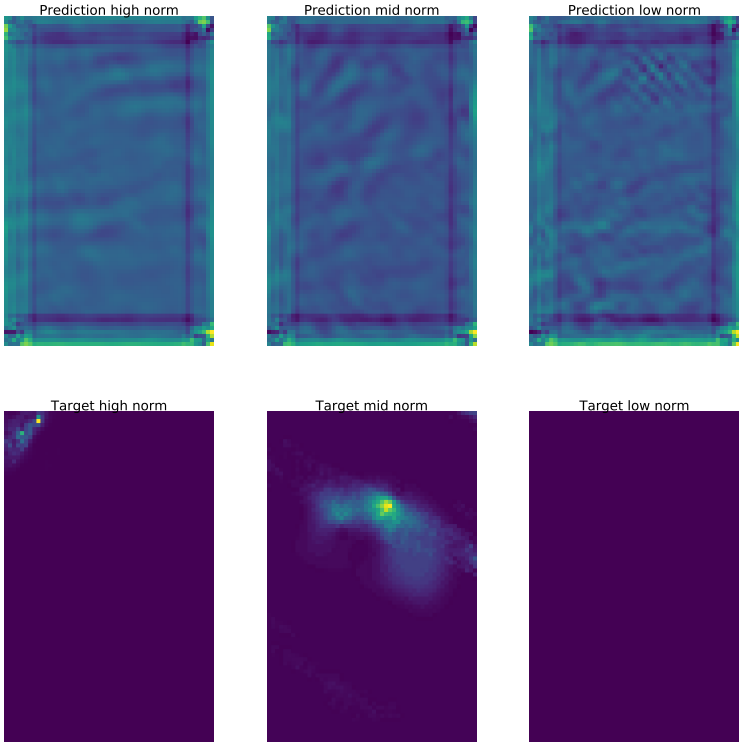


FIGURE A.31: Predictions and target data for simple neural network using mse, tanh and a learning rate of 1e-4 normalized.

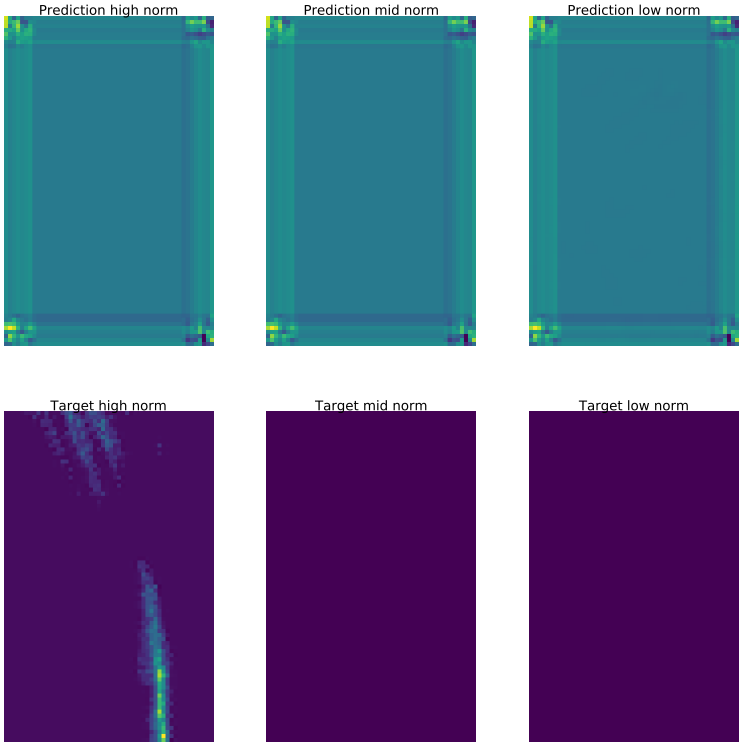


FIGURE A.32: Predictions and target data for simple neural network using mse, tanh and a learning rate of 1e-3 normalized.

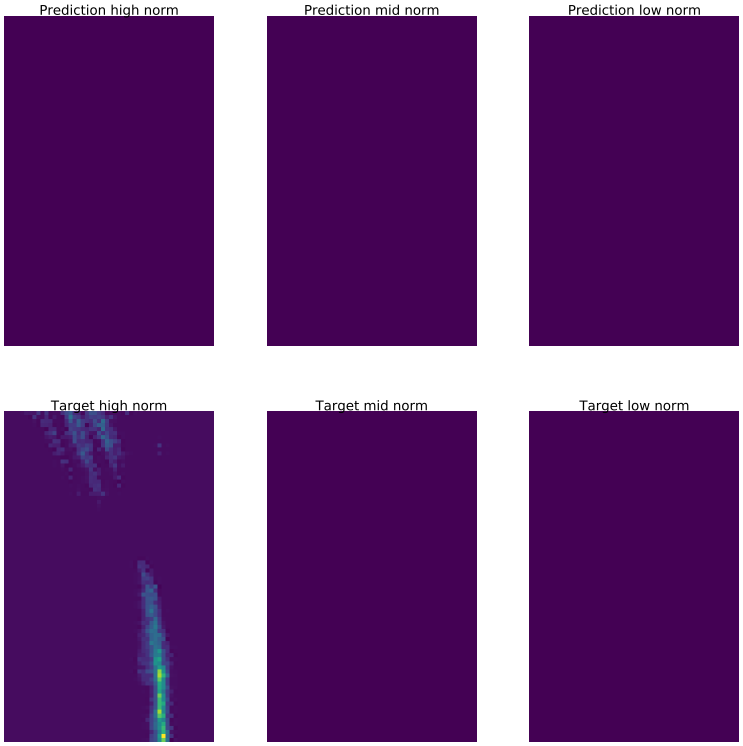


FIGURE A.33: Predictions and target data for simple neural network using mse, tanh and a learning rate of 1e-2 normalized.

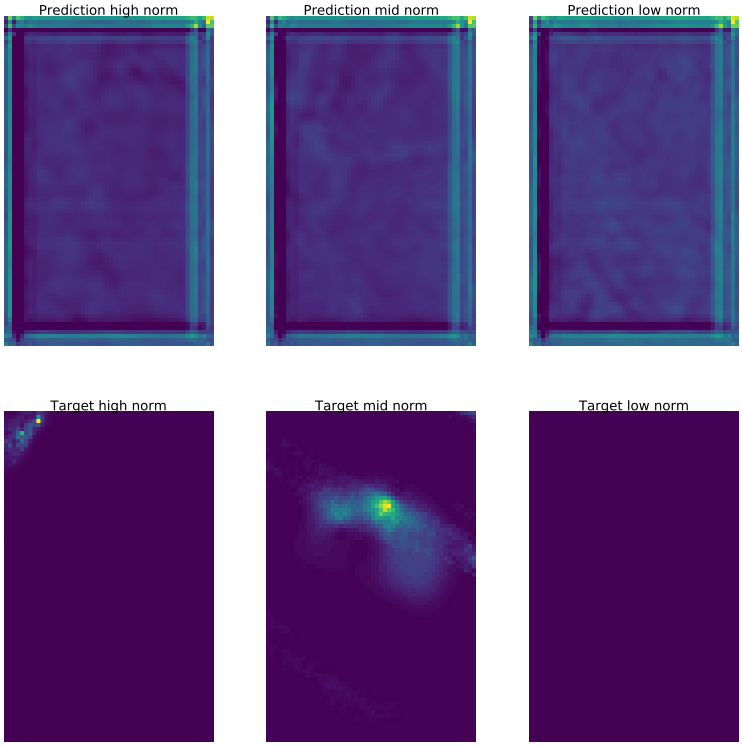


FIGURE A.34: Predictions and target data for simple neural network using mse, relu and a learning rate of 1e-4 normalized.

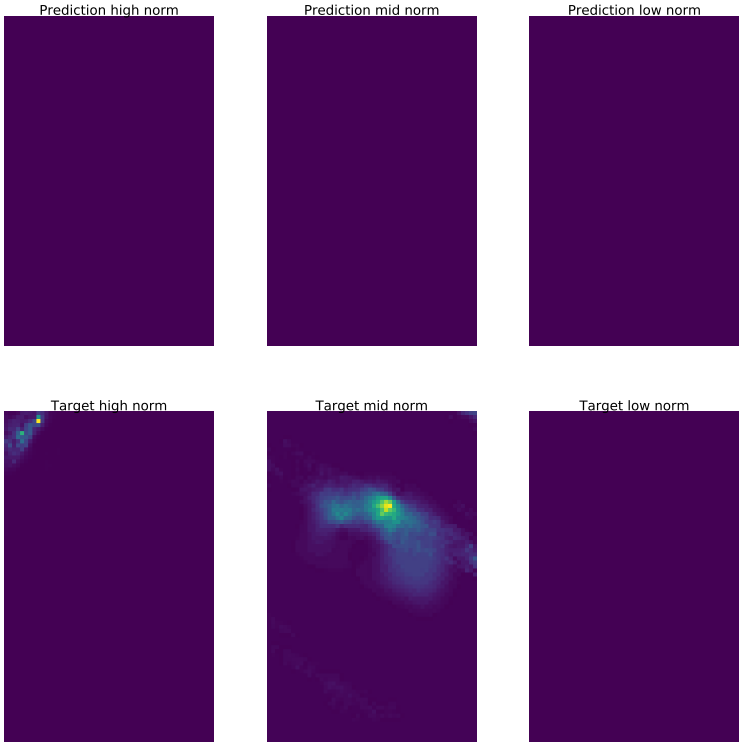


FIGURE A.35: Predictions and target data for simple neural network using mse, relu and a learning rate of 1e-3 normalized.

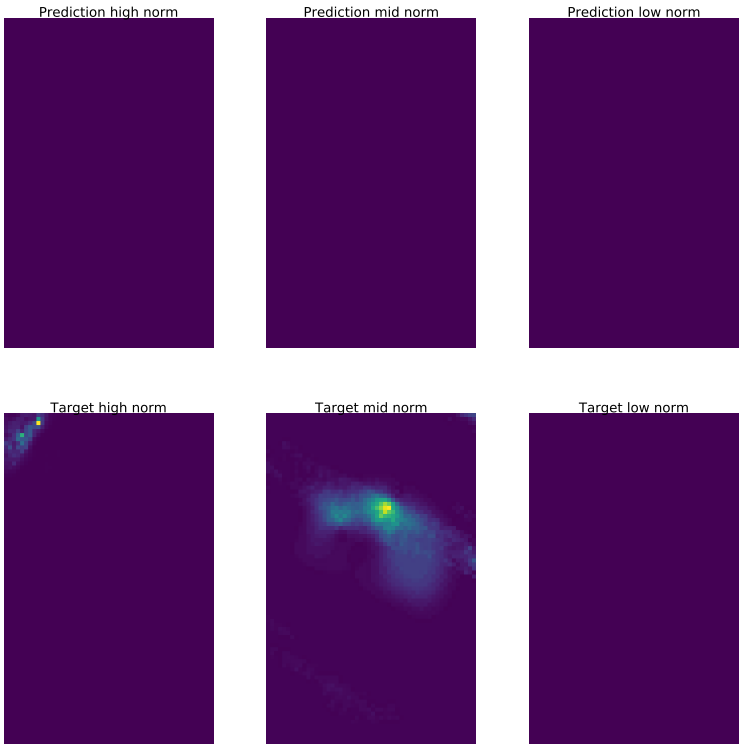


FIGURE A.36: Predictions and target data for simple neural network using mse, relu and a learning rate of 1e-2 normalized.

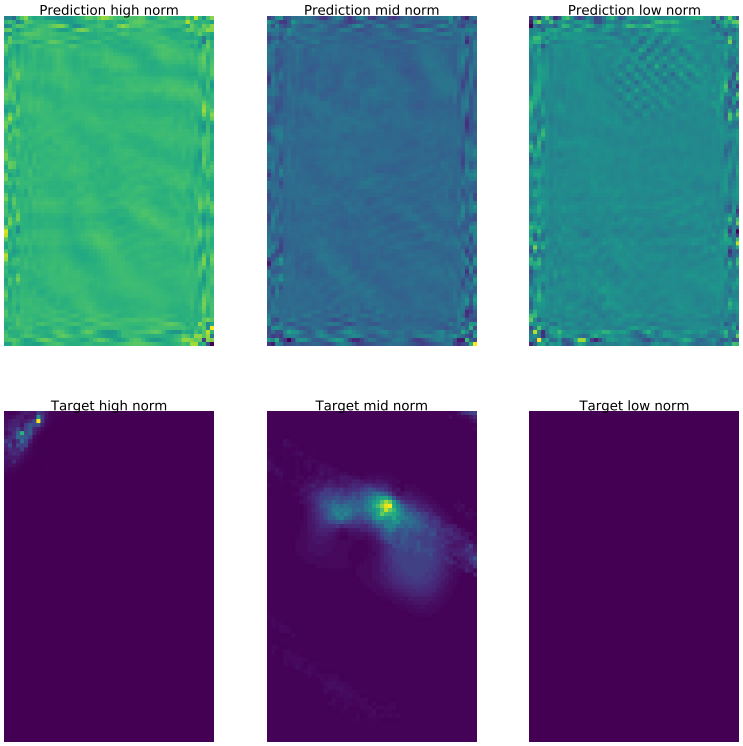


FIGURE A.37: Predictions and target data for simple neural network using mae, tanh and a learning rate of 1e-4 normalized.

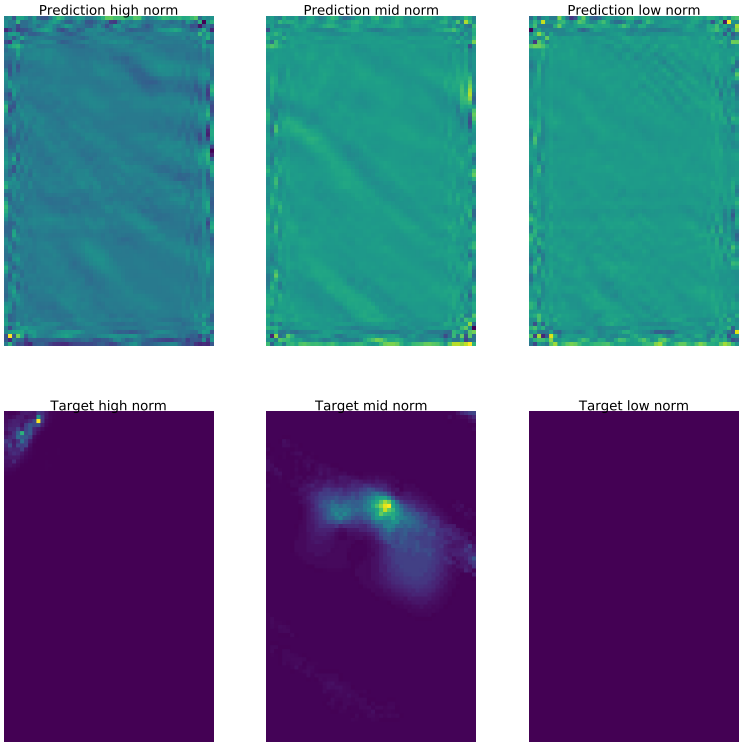


FIGURE A.38: Predictions and target data for simple neural network using mae, tanh and a learning rate of 1e-3 normalized.

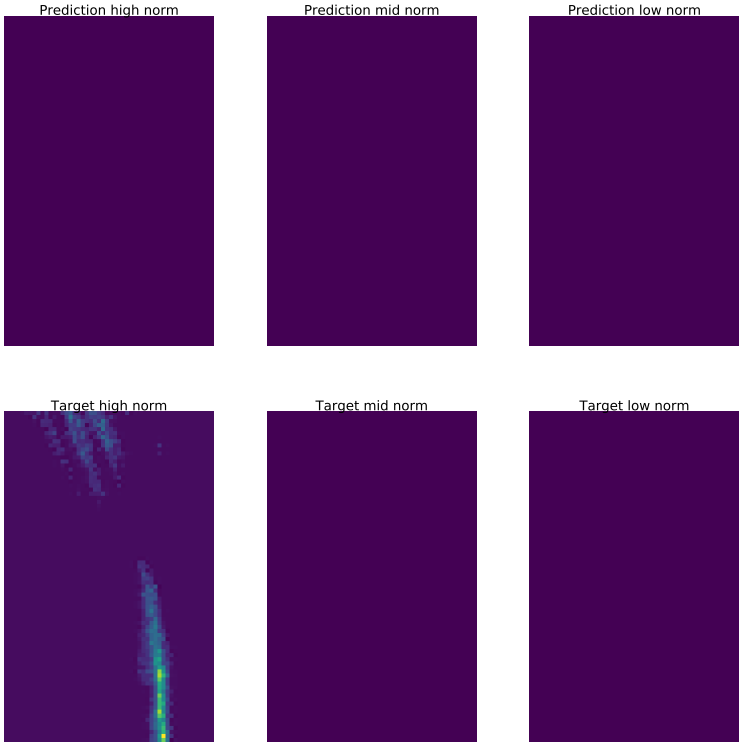


FIGURE A.39: Predictions and target data for simple neural network using mae, tanh and a learning rate of 1e-2 normalized.

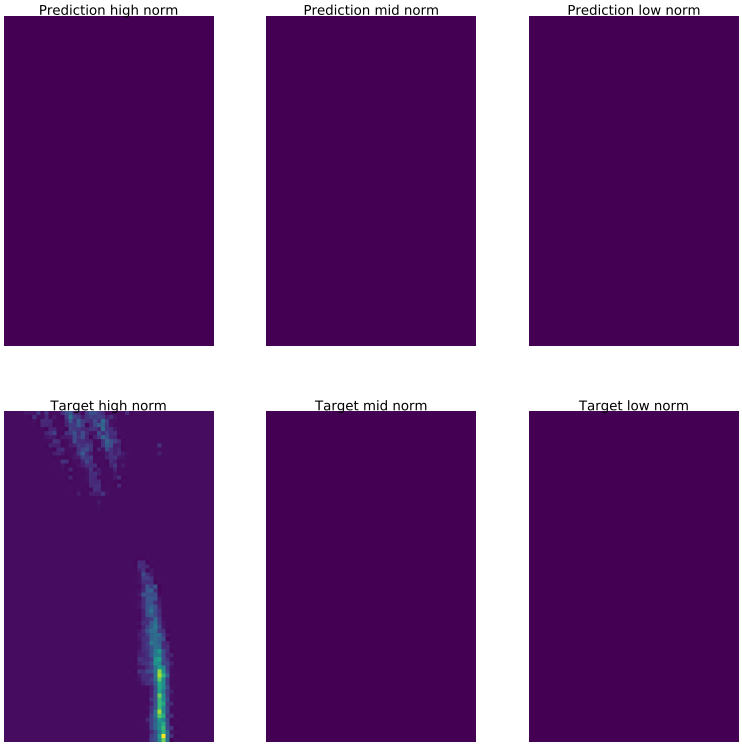


FIGURE A.40: Predictions and target data for simple neural network using mae, relu and a learning rate of 1e-4 normalized.

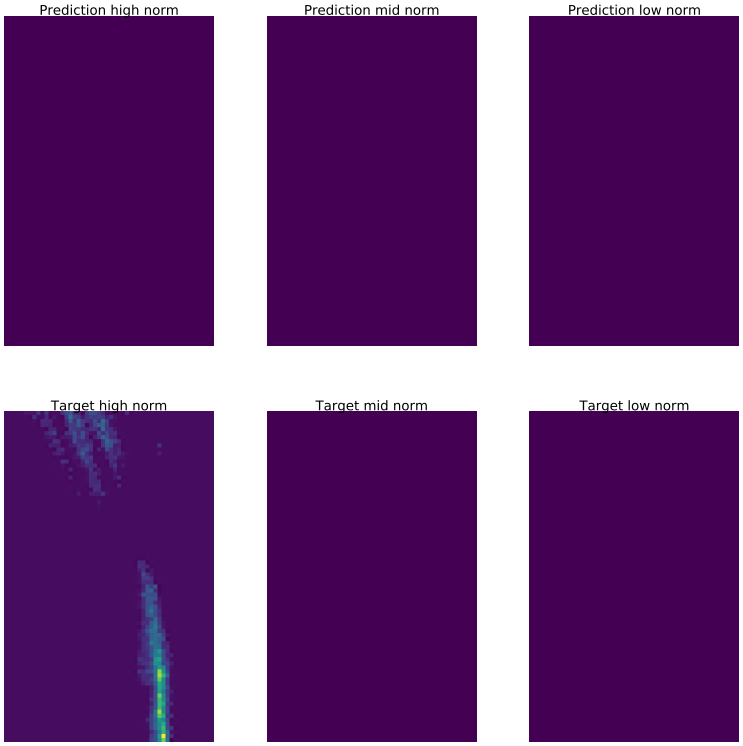


FIGURE A.41: Predictions and target data for simple neural network using mae, relu and a learning rate of 1e-3 normalized.

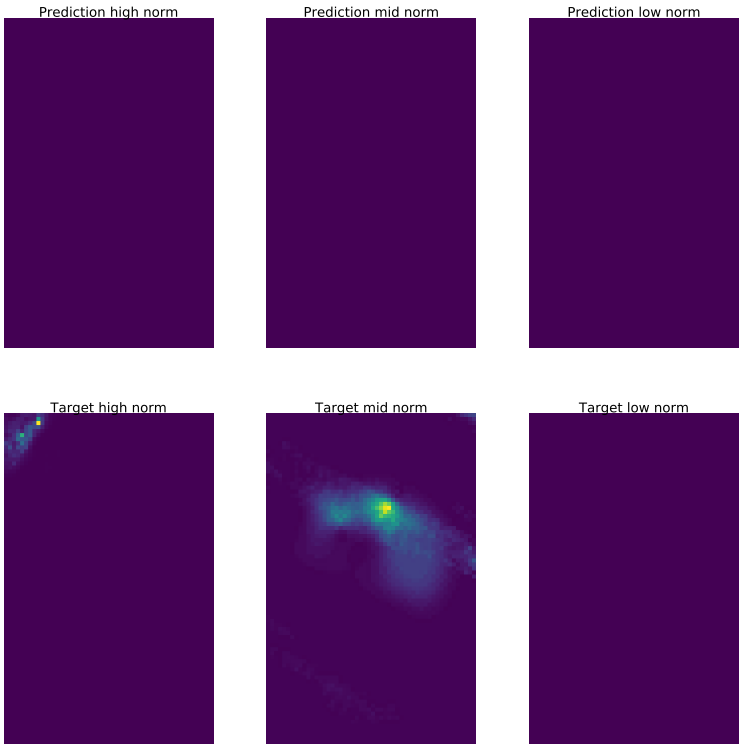


FIGURE A.42: Predictions and target data for simple neural network using mae, relu and a learning rate of 1e-2 normalized.

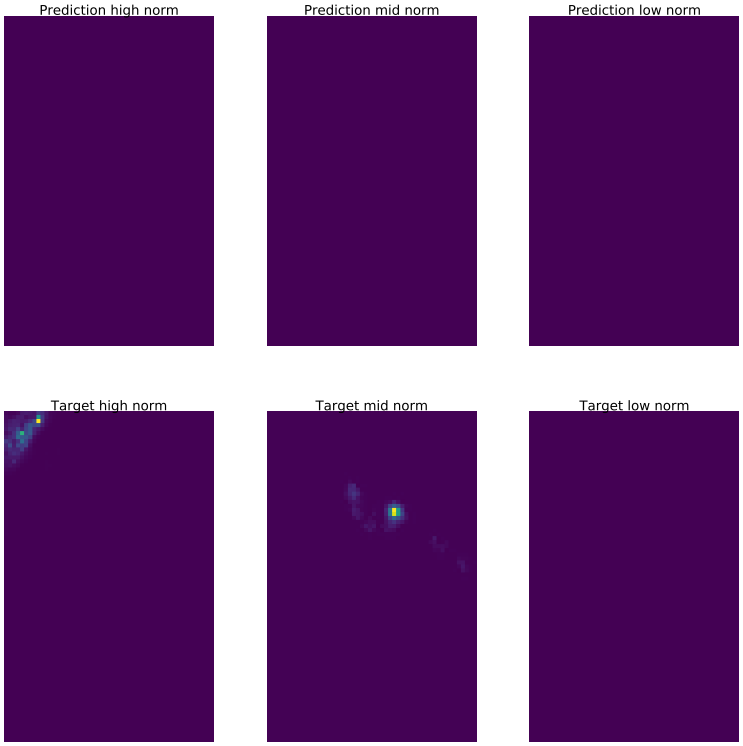


FIGURE A.43: Predictions and target data for convolutional neural network with upsampling in the middle normalized.

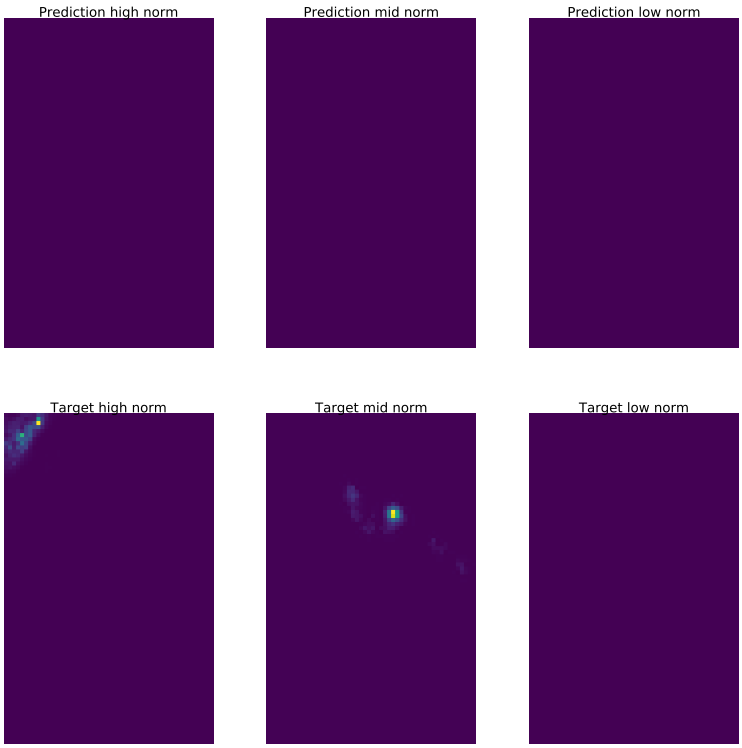


FIGURE A.44: Predictions and target data for convolutional neural network with upsampling at the end normalized.

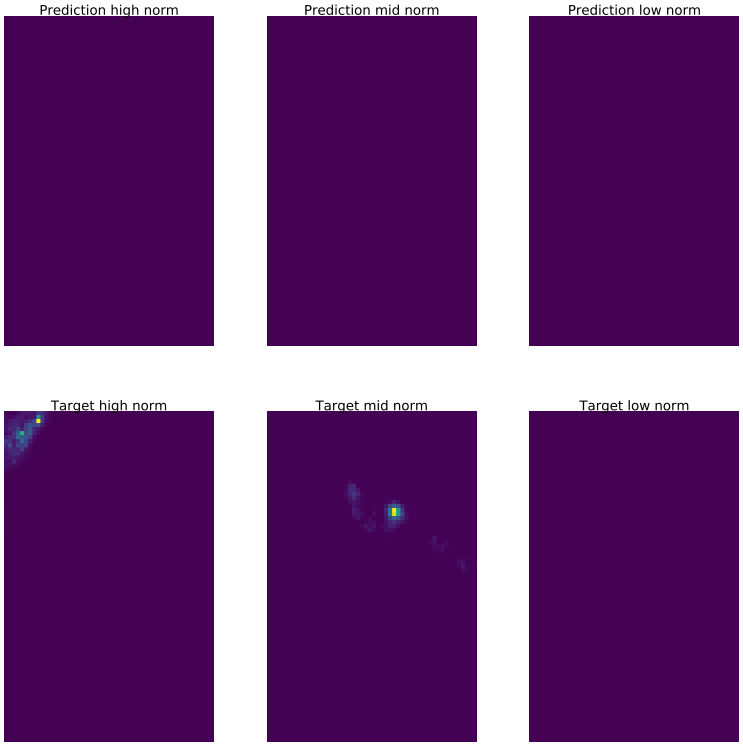


FIGURE A.45: Predictions and target data for convolutional neural network with upsampling at the beginning normalized.

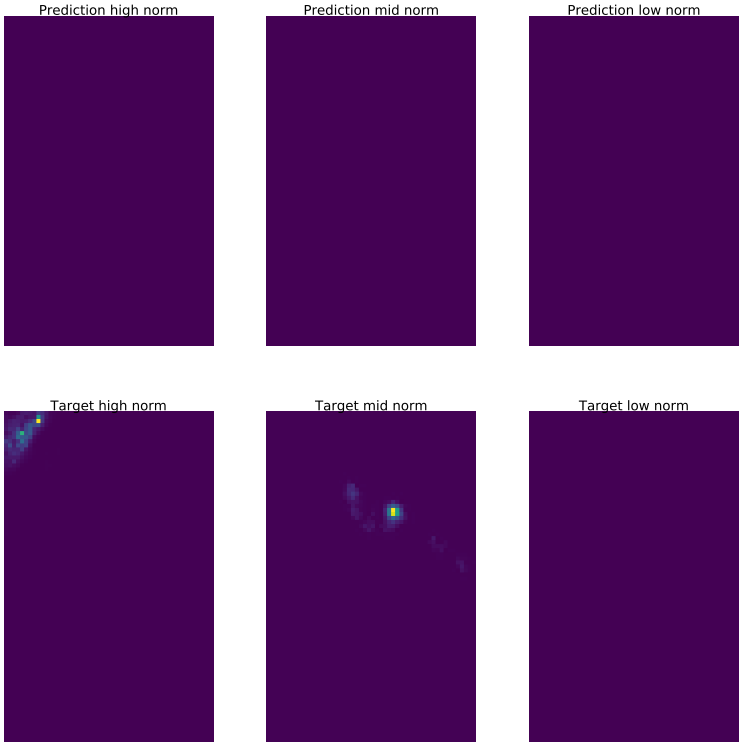


FIGURE A.46: Predictions and target data for deeper convolutional neural network normalized.

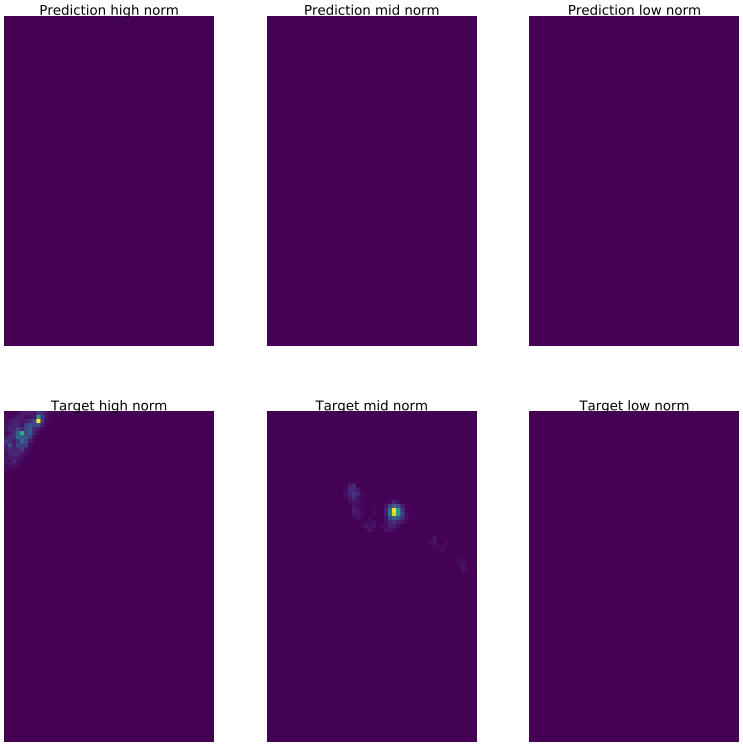


FIGURE A.47: Predictions and target data for deeper convolutional neural network with smaller kernel normalized.

Bibliography

Abadi, Martın et al. (2015). “TensorFlow: Large-scale machine learning on heterogeneous systems”. In:

Ahrens, B and A Beck (2008). “On upscaling of rain-gauge data for evaluating numerical weather forecasts”. In: *METEOROLOGY AND ATMOSPHERIC PHYSICS* 99.3-4, pp. 155–167. ISSN: 0177-7971. DOI: 10.1007/s00703-007-0261-8.

Baldauf, M, S Klink, and T Reinhardt (2016). “Kurze Beschreibung des Lokal-Modells Kürzestfrist COSMO-DE (LMK) und seiner Datenbanken auf dem Datenserver des DWD”. In:

Ben Bouallègue, Zied and Susanne E. Theis (2014). “Spatial techniques applied to precipitation ensemble forecasts: From verification results to probabilistic products”. In: *Meteorological Applications* 21.4, pp. 922–929. ISSN: 14698080. DOI: 10.1002/met.1435.

Berrocal, Veronica J., Alan E. Gelfand, and David M. Holland (2010). “A spatio-temporal downscaler for output from numerical models”. In: *Journal of Agricultural, Biological, and Environmental Statistics* 15.2, pp. 176–197. ISSN: 10857117. DOI: 10.1007/s13253-009-0004-z.

Beven, Keith et al. (2015). “Hyperresolution information and hyperresolution ignorance in modelling the hydrology of the land surface”. In: *Science China Earth Sciences* 58.1, pp. 25–35. ISSN: 16747313. DOI: 10.1007/s11430-014-5003-4.

Daniel Glasner Shai Bagon, Michal Irani (1993). “Super-Resolution from a Single Image”. In:

Dong, Chao, Chen Change Loy, and Xiaoou Tang (2016). “Accelerating the super-resolution convolutional neural network”. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9906 LNCS, pp. 391–407. ISSN: 16113349. DOI: 10.1007/978-3-319-46475-6{_}25.

Dong, Chao et al. (2014). “Learning a Deep Convolutional Network for Image Super-Resolution”. In: *European conference on computer vision*.

— (2016). “Image Super-Resolution Using Deep Convolutional Networks.” In: *IEEE transactions on pattern analysis and machine intelligence* 38.2, pp. 295–

307. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2015.2439281. URL: http://www.ncbi.nlm.nih.gov/pubmed/26761735.

DWD (2018). "Deutscher Wetterdienst Act (DWD Act)". In: 2018.July 2017, pp. 1–8.

DWD, Deutscher Wetterdienst (2017). "RADOLAN : Radar Online Adjustment". In: p. 63067.

— (2018). "RADOLAN Kurzbeschreibung". In: April, pp. 1–5.

— (2019a). *Deutscher Wetterdienst - Geschäftsbereiche*. URL: https://www.dwd.de/DE/derdwd/organisation/geschaeftsbereiche/geschaeftsbereiche_node.html#doc20302bodyText4.

— (2019b). *Deutscher Wetterdienst - Numerische Methoden*. URL: https://www.dwd.de/DE/forschung/wettervorhersage/num_modellierung/01_num_vorhersagemodelle/01a_num_methoden/numerischemethoden_node.html.

— (2019c). *Deutscher Wetterdienst Landing page*. URL: http://www.dwd.de.

Ebtehaj, A M, E Foufoula-Georgiou, and G Lerman (2012). "Sparse regularization for precipitation downscaling". In: *JOURNAL OF GEOPHYSICAL RESEARCH-ATMOSPHERES* 117. ISSN: 2169-897X. DOI: 10.1029/2011JD017057.

ECMWF, European Centre for Medium-Range Weather Forecasts (2019). *Advancing global NWP through international collaboration*. URL: http://www.ecmwf.int/.

Farsiu, Sina et al. (2004). "Fast and robust multiframe super resolution". In: *IEEE Transactions on Image Processing* 13.10, pp. 1327–1344. ISSN: 10577149. DOI: 10.1109/TIP.2004.834669.

Fettke, Peter (2006). "State-of-the-Art des State-of-the-Art Eine Untersuchung der Forschungsmethode „ Review " innerhalb der Wirtschaftsinformatik". In: 48, pp. 257–266.

Folk, Mike et al. (2011). "An Overview of the HDF5 Technology Suite and its Applications". In:

Freedman, Gilad and Raanan Fattal (2011). ""Image and video upscaling from local self-examples". In: *ACM Transactions on Graphics (TOG) 30.2* 12.

Freitag, M. and R. Potthast (2013). "Large Scale Inverse Problems Computational Methods and Applications In the Earth Sciences". In:

Greff, Klaus et al. (2017). "The Sacred Infrastructure for Computational Research". In: Scipy, pp. 49–56. DOI: 10.25080/shinma-7f4c6e7-008.

Gunturk, Bahadir K. et al. (2003). "Eigenface-domain super-resolution for face recognition". In: *IEEE Transactions on Image Processing* 12.5, pp. 597–606. ISSN: 10577149. DOI: 10.1109/TIP.2003.811513.

Ha, Wonsook, Prasanna H. Gowda, and Terry A. Howell (2013). "A review of downscaling methods for remote sensing-based irrigation management: Part I". In: *Irrigation Science* 31.4, pp. 831–850. ISSN: 03427188. DOI: 10.1007/s00271-012-0331-7.

He, Kaiming et al. (2016). "Deep Residual Learning for Image Recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. ISSN: 0366-7022. DOI: 10.1246/cl.2003.428.

HKO, The Government of the Hong Kong Special Administrative Region (2019). *Hong Kong Observatory*. URL: http://www.hko.gov.hk/.

Horé, Alain and Djemel Ziou (2010). "Image quality metrics: PSNR vs. SSIM". In: *Proceedings - International Conference on Pattern Recognition*, pp. 2366–2369. ISSN: 10514651. DOI: 10.1109/ICPR.2010.579.

Huang, Jia-Bin, Abhishek Singh, and Narendra Ahuja (2015). "Single Image Super-resolution from Transformed Self-Exemplars". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Hunt, Brian R., Eric J. Kostelich, and Istvan Szunyogh (2007). "Efficient data assimilation for spatiotemporal chaos: A local ensemble transform Kalman filter". In: *Physica D: Nonlinear Phenomena* 230.1-2, pp. 112–126. ISSN: 01672789. DOI: 10.1016/j.physd.2006.11.008. arXiv: 0511236v2 [physics].

IITM, Indian Institute of Tropical Meteorology (2019). *Indian Institute of Tropical Meteorology - Landing page*. URL: https://www.tropmet.res.in/.

Im, Jungho et al. (2016). "Downscaling of AMSR-E soil moisture with MODIS products using machine learning approaches". In: *Environmental Earth Sciences* 75.15, pp. 1–19. ISSN: 18666299. DOI: 10.1007/s12665-016-5917-6.

Johnson, Justin, Alexandre Alahi, and Li Fei-Fei (2016). "Perceptual losses for real-time style transfer and super-resolution". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9906 LNCS, pp. 694–711. ISSN: 16113349. DOI: 10.1007/978-3-319-46475-6{_}43.

JSC, Jülich Supercomputing Centre (2019). *DeepRain: Machine Learning Methods for Precipitation Forecasts*. URL: https://www.fzj.de/SharedDocs/Meldungen/IAS/JSC/EN/2019/2019-02-deeprain.html.

Kim, Hyunjik et al. (2019). "Attentive Neural Processes". In: pp. 1–18. URL: http://arxiv.org/abs/1901.05761.

Kim, Jiwon, Jung Kwon Lee, and Kyoung Mu Lee (2015a). "Accurate Image Super-Resolution Using Very Deep Convolutional Networks". In: pp. 1646–1654. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2015.2439281. URL: http://arxiv.org/abs/1511.04587.

— (2015b). "Deeply-Recursive Convolutional Network for Image Super-Resolution". In: ISSN: 10636919. DOI: 10.1109/CVPR.2016.181. URL: http://arxiv.org/abs/1511.04491.

Kim, Kwang In and Younghee Kwon (2010). "Single-image super-resolution using sparse regression and natural image prior". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.6, pp. 1127–1133. ISSN: 01628828. DOI: 10.1109/TPAMI.2010.25.

Ledig, Christian et al. (2017). "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network". In: *Cvpr*, pp. 4681–4690. ISSN: 0018-5043. DOI: 10.1109/CVPR.2017.19. URL: http://openaccess.thecvf.com/content_cvpr_2017/papers/Ledig_Photo-Realistic_Single_Image_CVPR_2017_paper.pdf.

Leuenberger, Daniel et al. (2010). "A Generalization of the SLEVE Vertical Coordinate". In: *Monthly Weather Review* 138.9, pp. 3683–3689. ISSN: 0027-0644. DOI: 10.1175/2010mwr3307.1.

Li, Hao et al. (2017). "Visualizing the Loss Landscape of Neural Nets". In: 1.NeurIPS 2018, pp. 1–11. URL: http://arxiv.org/abs/1712.09913.

Li, X et al. (2012). "Geostationary weather radar super-resolution modelling and reconstruction process". In: *International Journal of ...* URL: https://www.inderscienceonline.com/doi/abs/10.1504/IJSPM.2012.047867.

Mahour, Milad et al. (2015). "Integrating super resolution mapping and SEBS modeling for evapotranspiration mapping at the field scale". In: *Precision Agriculture* 16.5, pp. 571–586. ISSN: 15731618. DOI: 10.1007/s11119-015-9395-8. URL: http://dx.doi.org/10.1007/s11119-015-9395-8.

Nayak, Deepak Ranjan (2013). "A Survey on Rainfall Prediction using Artificial Neural Network". In: 72.16.

Nikam, Valmik B and B B Meshram (2013). "Modeling Rainfall Prediction Using Data Mining Method". In: *2013 Fifth International Conference on Computational Intelligence, Modelling and Simulation*, pp. 132–136. DOI: 10.1109/CIMSim.2013.29.

Osendorfer, Christian, Hubert Soyer, and Patrick van der Smagt (2014). "Image Super-Resolution with Fast Approximate Convolutional Sparse Coding". In: Iconip, pp. 250–257. DOI: 10.1007/978-3-319-12643-2{_}31.

Peled, Sharon and Yehezkel Yeshurun (2001). "Superresolution in MRI: application to human white matter fiber tract visualization by diffusion tensor imaging". In: *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine* 45.1, pp. 29–35. ISSN: 0740-3194.

Reinert, D et al. (2019). "Database Reference Manual for ICON and ICON-EPS". In: DOI: 10.5676/DWD.

RMI, Royal Meteorological Institute of Belgium (2019). *Weather Bruessel*. URL: https://www.meteo.be/en.

Russell, Stuart J and Peter Norvig (2016). *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited.

Schraff, C. et al. (2016). "Kilometre-scale ensemble data assimilation for the COSMO model (KENDA)". In: *Quarterly Journal of the Royal Meteorological Society* 142.696, pp. 1453–1472. ISSN: 1477870X. DOI: 10.1002/qj.2748.

Shan, Qi et al. (2008). "Fast image/video upsampling". In: *ACM Transactions on Graphics* 27.5, p. 1. ISSN: 07300301. DOI: 10.1145/1409060.1409106.

Shi, Wenzhe et al. (2013). "Cardiac image super-resolution with global correspondence using multi-atlas PatchMatch". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 8151 LNCS.PART 3, pp. 9–16. ISSN: 03029743. DOI: 10.1007/978-3-642-40760-4{_}2.

Shi, Wenzhe et al. (2016). "Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network". In: *Computer Vision and Patter Recognition*, pp. 1874–1883. DOI: 10.1109/CVPR.2016.207.

Singh, R. S. et al. (2015). "Toward hyper-resolution land-surface modeling: The effects of fine-scale topography and soil texture on CLM4.0 simulations over the Southwestern U.S." In: *Water Resources Research* 51.4, pp. 2648–2667. ISSN: 19447973. DOI: 10.1002/2014WR015686.

Stephan, K, S Klink, and C Schraff (2008). "Assimilation of radar-derived rain rates into the convective-scale model COSMO-DE at DWD". In: 1326.July, pp. 1315–1326. DOI: 10.1002/qj.

Stott, P. (2016). "How climate change affects extreme weather events". In: *Science* 352.6293, pp. 1517–1518. ISSN: 0036-8075. DOI: 10.1126/science.aaf7271.

Sun, Jian, Zongben Xu, and Heung Yeung Shum (2011). "Gradient profile prior and its applications in image super-resolution and enhancement".

In: *IEEE Transactions on Image Processing* 20.6, pp. 1529–1542. ISSN: 10577149. DOI: 10.1109/TIP.2010.2095871.

Theis, S et al. (2017). "Beschreibung des COSMO-DE-EPS und seiner Ausgabe in die Datenbanken des DWD". In:

Thornton, Matt W, Peter M Atkinson, and D A Holland (2006). "Sub-pixel mapping of rural land cover objects from fine spatial resolution satellite sensor imagery using super-resolution pixel-swapping". In: *International Journal of Remote Sensing* 27.3, pp. 473–491. ISSN: 0143-1161.

Timofte, Radu, Vincent De Smet, and Luc Van Gool (2015). "A+: Adjusted anchored neighborhood regression for fast super-resolution". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9006, pp. 111–126. ISSN: 16113349. DOI: 10.1007/978-3-319-16817-3{_}8.

Toth, E, A Brath, and A Montanari (2000). "Comparison of short-term rainfall prediction models for real-time flood forecasting". In: 239, pp. 132–147.

Van Rossum, Guido and Fred L Drake Jr (1995). *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands.

Wang, Shenlong et al. (2012). "Semi-Coupled Dictionary Learning with Applications to Image Super-Resolution and Photo-Sketch Synthesis". In: *Cvpr*, pp. 2216–2223. DOI: 10.1109/CVPR.2012.6247930.

Wang, Zhou et al. (2003). "Multiscale structural similarity for image quality assessment". In: *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers* 2. URL: http://videoclarity.com/wp-content/uploads/2013/05/MS-SSIMAsilomar.pdf.

— (2004). "Image Quality Assessment: From Error Visibility to Structural Similarity". In: *Ieee Transactions on Image Processing* 13.4, pp. 1–14. ISSN: 10577149. DOI: 10.1109/TIP.2003.819861.

Webster, Jane and Richard T Watson (2010). "ANALYZING THE PAST TO PREPARE FOR THE FUTURE : WRITING A REVIEW". In: 26.2.

Yang, Chih-yuan, Chao Ma, and Ming-hsuan Yang (2014). "Single-Image Super-Resolution : A Benchmark". In: pp. 372–386.

Zhang, Xing et al. (2019). "Weather Radar Echo Super-Resolution Reconstruction Based on Nonlocal Self-Similarity Sparse Representation". In: *Atmosphere* 10.5, p. 254. DOI: 10.3390/atmos10050254.

Zhang, Yin, Junjie Wu, and Jianyu Yang (2014). "A sparse sampling strategy for angular superresolution of real beam scanning radar". In: *Eurasip Journal on Advances in Signal Processing* 2014.1, pp. 1–8. ISSN: 16876180. DOI: 10.1186/1687-6180-2014-110.

Zhao, Hang et al. (2016). "Loss Functions for Image Restoration With Neural Networks". In: *IEEE Transactions on Computational Imaging* 3.1, pp. 47–57. DOI: 10.1109/tci.2016.2644865.

Declaration of Authorship

I hereby certify that the work presented here is, to the best of my knowledge and belief, original and the result of my own investigations, except as acknowledged, and has not been submitted, either in part or whole, for a degree at this or any other university.

_____

signature

_____

city, date