Radar-Based Volumetric Precipitation Nowcasting: A 3D Convolutional Neural Network with U-Net Architecture

Peter Pavlík[1,2], Viera Rozinajová[2,3] and Anna Bou Ezzeddine[2]

[1]Faculty of Information Technology, Brno University of Technology, Božetěchova 1/2, Brno-Královo Pole, 612 00, Czechia
[2]Kempelen Institute of Intelligent Technologies, Mlynské Nivy II. 18890/5, Bratislava, 821 09, Slovakia
[3]Slovak Centre for Research of Artificial Intelligence - slovak.AI, Slovakia

Abstract

In recent years – like in many other domains – deep learning models have found their place in the domain of precipitation nowcasting. Many of these models are based on the U-Net architecture, which was originally developed for biomedical segmentation, but is also useful for the generation of short-term forecasts and therefore applicable in the weather nowcasting domain. The existing U-Net-based models use sequential radar data mapped into a 2-dimensional Cartesian grid as input and output. We propose to incorporate a third - vertical - dimension to better predict precipitation phenomena such as convective rainfall and present our results here. We compare the nowcasting performance of two comparable U-Net models trained on two-dimensional and three-dimensional radar observation data. We show that using volumetric data results in a small, but significant reduction in prediction error.

Keywords

precipitation nowcasting, radar imaging, U-Net

1. Introduction

Accurate precipitation nowcasting is important for planning various human activities and tasks such as agriculture, construction building or winter road maintenance. Nowcasting is defined by the World Meteorological Agency as forecasting with local detail, by any method, over a period from the present to six hours ahead, including a detailed description of the present weather [1].

In practice, simpler - and therefore faster - models outperform complex Numerical Weather Prediction (NWP) models at the task of precipitation nowcasting because NWP models cannot consider the latest observations due to their long inference time. The highly sophisticated NWP models usually need hours to produce their forecasts and so they are not able to take into consideration the latest data observations. Even a simple model that can quickly output a prediction will outperform the NWP models at the task of precipitation nowcasting simply by the fact that it can consider the present data. Nowcasting models can work in conjunction with NWP models and use their long-term forecasts as additional inputs to further refine their nowcasts [1].

Precipitation nowcasting is usually performed using temporal extrapolation of past data from weather radar systems because it requires highly accurate and constantly updated data about precipitation fields, i.e. the location of storms, wind, fog, snow etc. Weather radar systems are essential for nowcasting because they directly observe precipitation particles with an update rate of a few minutes [1]. See Figure 1 for an example of a radar precipitation map.

In the last few years, deep learning precipitation nowcasting approaches, such as convolutional neural networks (CNN), started to gain attention. From the initial ConvLSTM model [2], through encoder-decoder U-Net architectures [3, 4], to the recently-introduced GAN-based approaches [5, 6], the CNN models proved to consistently outperform the operational state-of-the-art methods in the domain [6].

Most precipitation nowcasting models only use the radar data mapped to a 2D Cartesian grid, aggregating the vertical dimension, even though the raw output of weather radar systems consists of multiple measurements at different elevation angles and polar coordinates that capture the precipitation phenomena in 3-dimensional space around the radar.

We propose using volumetric data from multiple altitudes to give the model as much data about the observation as possible. Providing information about the vertical motion of precipitation particles, as well as their vertical extension, could potentially be valuable for the model, as they are an important factor in predicting the behavior of convective storms [7].

We compare two models - a reference U-Net architecture based on existing research [3, 4] and an alternative with 2D convolutional layers replaced by 3D convolution. We evaluate their performance in the task of predicting

CDCEO 2022: 2nd Workshop on Complex Data Challenges in Earth Observation, July 25, 2022, Vienna, Austria
peter.pavlik@kinit.sk (P. Pavlík); viera.rozinajova@kinit.sk (V. Rozinajová); anna.bou.ezzeddine@kinit.sk (A. B. Ezzeddine)
0000-0002-7468-5503 (P. Pavlík); 0000-0003-1302-6261 (V. Rozinajová); 0000-0002-3341-6059 (A. B. Ezzeddine)

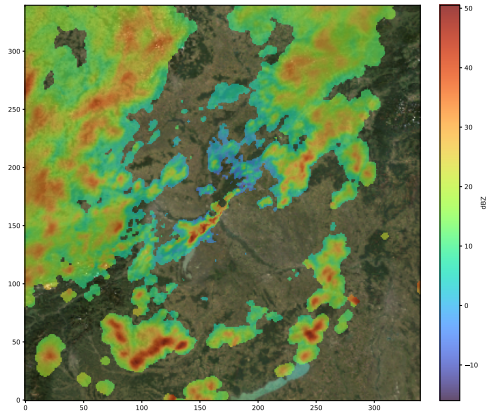

Figure 1: A single radar echo observation. The shown reflectivity values represent reflectivity captured at 2 km above radar (CAPPI). The reflectivity map is overlaid over a satellite image of the appropriate area centered on the Malý Javorník radar station generated using Google Earth Engine [8]. Landsat-8 image courtesy of the U.S. Geological Survey.

a single constant-altitude radar reflectivity observation 30 minutes into the future.

Our experiments show that providing volumetric data from multiple altitude levels results in small, but statistically significant reduction of prediction error.

2. Related Work

Many automated nowcasting systems that employ various inputs and computation approaches are in use today [9, 10, 11, 12, 13]. These systems are generally based on extrapolating past observed rainfall data forwards in time. They typically estimate the future advection based on motion observed in the most recent radar images using cross-correlation or optical flow techniques [1].

Some nowcasting systems use the cell tracking approach. They firstly identify storms in the radar scan and then locate the corresponding object in the consecutive scans to track its motion. Cell tracking is useful for tracking severe storms and is useful for generating early warnings [1].

The shortcoming of these advection nowcasting methods is the assumption that the observed precipitation field will not change, only move elsewhere. Therefore, they lack the capability to predict beginning of new precipitation phenomena such as convective initiation (start of a storm triggered by rising moist warm air) or the decaying of the storm at the end of its lifecycle [1, 14].

In the past years, data-driven approaches using deep learning to construct precipitation nowcasting models to mitigate these limitations have started to gain attention [2, 3, 6].

The first deep learning approach applied to the task of precipitation nowcasting was a ConvLSTM model presented in [2] that outperformed the operational optical-flow-based ROVER nowcasting system. Experiments with other CNN architectures started, such as a ConvGRU model from [15] or a U-Net-based architecture introduced in [16]. The U-Net architectures, originally developed for segmentation of medical images [17], proved to be quite popular with models such as RainNet[3] and SmaAt-U-Net[4] further exploring this approach.

The previously mentioned neural network regression models trying to nowcast the future state of precipitation fields were affected by blurring. When using traditional gridpoint-based verification statistics such as Mean Squared Error (MSE) as the training loss function, we face the so-called "double penalty problem". A forecast of a precipitation feature that is correct in terms of intensity, size, and timing, but incorrect concerning location, results in very large mean square error [18]. This causes the model to produce blurry outputs to mitigate the penalisation caused by spatially incorrect precipitation features.

The blurry predictions pose one of the biggest challenges for anyone trying to develop a nowcasting model based on machine learning as such predictions have difficulties predicting extreme events due to the smoothing. Recently, this problem started to be addressed by training models using the Generative Adversarial Network (GAN) approach, the most prominent being DGMR[6]. They introduced a GAN framework[19] to solve the problem of blurry predictions present in other deep learning precipitation nowcasting models such as RainNet. Model is trained using a combination of two discriminators inspired by existing research in video generation and a regularization term that comprise the loss function. The first discriminator, spatial, discourages blurry predictions while the second one, temporal, discourages jumpy predictions. The regularization term penalizes deviations between the observed radar sequences and the model prediction. The DGMR model can be currently considered the state-of-the-art in the precipitation nowcasting domain.

2.1. Motivation for Volumetric Nowcasting

The application of deep learning models for precipitation nowcasting is the focus of many research works. However, the vast majority of the models use 2-dimensional aggregate radar products and thus throw away any information which can be gained from processing the vertical structure of precipitation objects captured by the radar.

When reviewing the existing works in the precipitation nowcasting domain, we identified a need to explore the effect of working with 3-dimensional volumetric

radar data. By processing the data into a 2D aggregated map, we lose all information about the vertical structure of the precipitation particles detected by the radar. The model trained in this way cannot consider the vertical movement of particles caused by updraft or downdraft and predict the future precipitation accordingly.

Compared to 2-dimensional precipitation nowcasting, volumetric models are much less prevalent. One such model was presented in [20], where a ConvLSTM model was used to predict future radar reflectivity. The model input shape is 18×18×20 (18×18 km with 1 km resolution, 10 km above at 500 m resolution) provided at multiple time steps, each one is processed by a 3D-CNN first, then passed on to ConvLSTM sequential network. The output is a classification for the central region of 6 × 6 km predicting whether the reflectivity in the next 30 and 60 minutes will exceed a set threshold. The final result is a binary map with resolution of 6 × 6 km. The problem with this approach is that the model cannot consider any fast moving precipitation particles, since it cannot see more than 6 km past its target region. Also, the target region size of 6 × 6 km can hardly be considered a high spatial resolution, which is one of the defining traits of nowcasting.

One other work worth mentioning is a 3D-CNN+GAN hybrid model from [5]. This model is quite sophisticated. It uses the GAN-based approach to predict plausible data and a weighted MSE loss function to give more importance to high reflectivity values, resulting in better ability to predict extreme precipitation events and reduce output blurring. However, the third data dimension is not actually the altitude above radar we want to consider, but time - i.e. the past observations are not as separate channels, but form a 3D volume. Nevertheless, the model drives the development of 3D-CNN models for precipitation nowcasting.

3. Radar Reflectivity Dataset

To explore the effect of volumetric precipitation nowcasting, we collaborated with the Slovak Meteorological Institute that provided us a dataset of roughly 3.5 years of reflectivity data from Malý Javorník weather radar station. The data is captured in 5 minute intervals. The dataset consists of 355 761 separate observations in the ODIM HDF5 format.

The radar captures the precipitation particles in the air by measuring returned radar wave power (echo) after hitting precipitation particles. This value is called reflectivity, measured in logarithmic dimensionless units called decibels (dBZ). The data consists of reflectivity values at the so-called reflectivity gates in multiple elevation angles distributed around the radar station and encoded in polar coordinates. See Figure 2 for a vertical slice of a single radar observation.

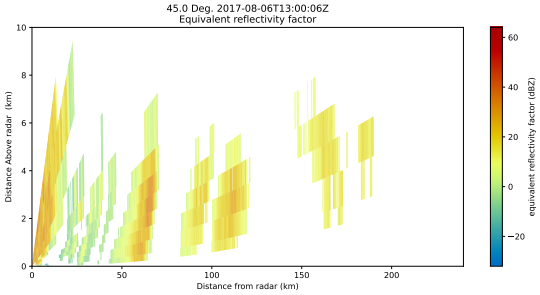

Figure 2: Vertical slice of a single radar reflectivity observation at a set azimuth. The separate "rays" at different elevation angles are identifiable.

Since the convolutional neural network models cannot process the data in polar coordinates, we need to convert them into Cartesian maps. We processed the data using the Py-ART Python library [21]. The radar echo observations are typically aggregated into precipitation maps in two forms. The first one is Constant Altitude Plan Position Indicator (CAPPI), which displays reflectivity gate values at certain altitude slice above radar. The other is CMAX, which aggregates the vertical dimension and displays the maximum value in the vertical column for each data point. If a 3D volume is created from multiple CAPPI maps at different altitude levels, the product is called MCAPPI.

The reflectivity maps can be converted to rainfall rate maps using the Marshall-Palmer Z-R relationship[22]:

$$Z = 200R^{1.6} \tag{1}$$

where Z is the reflectivity factor and R is the rainfall rate in mm/h.

3.1. Training data selection

The dataset requires filtering before training since the majority of the observations are of clear skies with nothing to learn from. Most of the observations from the dataset therefore have no value for training the model and could even negatively affect the training by biasing the outputs toward clear sky prediction, while we are mostly interested in non-trivial cases with high precipitation. We filtered the images as follows:

1. Create a CAPPI radar reflectivity map at 2 km altitude above radar at 1 × 1 km resolution and select a center slice of size 336 × 336 km.
2. Convert reflectivity to rainfall rate according to Marshall-Palmer Z-R relationship (1).
3. Compute the ratio of rainy to clear pixels (threshold 0.05 mm/5 min or 0.6 mm/h - corresponds to slight rain).

4. If the rainfall map contains at least 20% of rainy pixels and 11 previous observations are available, add it to the target observation set.

Each selected target observation was included in the training dataset, along with a set number of previous observations to serve as inputs and non-target intermediary outputs. For our models, we decided to use 6 observations as input and 6 as output, effectively predicting the precipitation half an hour in advance based on the last half hour of data. This means that for each target observation, we also needed to include 11 leading observations in the dataset. This process returned 9 018 suitable target images which together with the necessary leading images represent 3.18% of the original dataset.

It should be noted that the data converted to rainfall described above was not used for training, only for filtering the target observations based on the ratio of rainy pixels. The actual training data used reflectivity directly for both 2D images and 3D volumes. The 2D dataset was a collection of CAPPI radar reflectivity maps at 2 km altitude above radar. A 3D dataset was a collection of CAPPI radar reflectivity maps at 8 altitude levels above radar, from 500 m.a.r to 4000 m.a.r. The extent of the data was set to 336×336 km centered on the radar station with spatial resolution of 1×1 km for both 2D and 3D data, resulting in images of size 336×336 pixels and $8 \times 336 \times 336$ voxels respectively for a single observation.

4. Model Architectures

To compare the impact of adding a vertical dimension as fairly as possible, we chose a basic U-Net architecture inspired by models developed in [3, 4] as a reference model. As U-Net is a fully convolutional neural network, converting it to process volumetric data is a trivial task - mostly just a matter of replacing 2D convolutional layers with 3D convolutions. Besides this, the model only required replacing 2D max-pooling layers in the encoder for 3D max-pooling and bilinear upsample in the decoder for trilinear. See Figure 3 for the specific number of channels and kernel sizes at each layer of the model. Both were implemented using the PyTorch library [23].

The conversion of the model from 2D to 3D convolutions was mostly straightforward and resulted in increasing the number of trainable parameters 3-fold from roughly 17 to 52 million. The three-fold increase is based on the fact that the model uses convolution kernels of size 3 at every convolutional layer, therefore each kernel has 27 ($3 \times 3 \times 3$) instead of 9 (3×3) weights (disregarding bias and multiple channels). Other architectural parameters of the model such as number of kernels at each layer were kept the same for the comparison between these models to be fair and dependent solely on the provided data as much as possible. The GPU processing time (disregarding the time to move the data to memory) was not affected, with both models needing around 6 ms of GPU time to generate a single output on our hardware.

Set	No. of obs.	% of original
Full Dataset	355761	100
Target Observations	9018	2.53
Target + Lead Obs.	11310	3.18
Training Set Targets	6515	1.83
Validation Set Targets	1150	0.32
Test Set Targets	1353	0.38

Table 1
The observation count of the full dataset, the subset selected for training according to the training data selection described in Section 3.1 and the sizes of train, test, validation splits.

4.1. Training and Evaluation

To train and evaluate the models, the training dataset was split into training, validation and test subsets in chronological order. The last 15% of target observations were selected for the test set, the rest was chosen for training. Out of these, the last 15% of target observations were again selected for validation and the rest was used as training samples. See Table 1 for the exact number of observations in each set.

Adam optimizer was used for training the model. To find the optimal training model hyperparameters - starting learning rate, optimizer learning rate scheduler parameters and gradient clipping threshold - we utilized the Bayesian sweep search provided by Weights & Biases[24]. We trained 20 models with 2D CNN architecture and 5 with 3D CNN architecture. The best performing model of each architecture variant was selected for performance evaluation. See Table 2 for all the possible hyperparameter values and the best performing ones for both 2D and 3D models. Early stopping after 15 non-improving epochs was utilized.

Choosing the right metric to evaluate the performance of precipitation nowcasting models is not simple. The correct method depends on a model's use-case and no single composite measure is currently able to objectively evaluate performance of precipitation nowcasting models [1]. While we outlined the shortcomings of using MSE to evaluate precipitation nowcasting models above in Section 2, we are using MSE as the loss function and the primary evaluation metric despite the double penalization effect that occurs since it is still the most commonly used metric in this domain. Additionally, to provide more insight into model performance, we are also computing mean model accuracy, precision, recall and F1 scores on binarized precipitation maps using a threshold value of

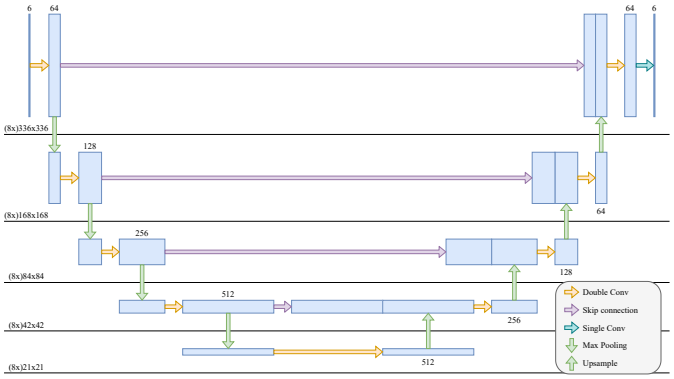

Figure 3: Diagram of the used U-Net model encoder-decoder architectures and the feed-forward process for both the 2D and 3D variant of the model. Each rectangle represents a multi-channel feature map with the number of channels shown above (or below in the decoder part). The spatial resolution of the feature maps at each level is shown at the left side of the diagram (the vertical dimension size of the 3D model is in brackets). Each arrow represents an operation with the data, see legend at the bottom right. The kernels of the double convolution operation are of size 3×3 or $3 \times 3 \times 3$, the kernels of the final single convolution operation are of size 1×1 or $1 \times 1 \times 1$ and the kernels of the max pooling operation are of size 2×2 or $1 \times 2 \times 2$ for 2D and 3D models respectively. All the convolutional layers used the ReLU activation function.

Hyperparameter	2D U-Net	3D U-Net
Batch size	**32**	**4**
Learning rate	5×10^{-5}, 7.5×10^{-5}, 1×10^{-4}, $\mathbf{2.5 \times 10^{-4}}$, 5×10^{-4}	5×10^{-5}, 7.5×10^{-5}, $\mathbf{1 \times 10^{-4}}$, 2.5×10^{-4}, 5×10^{-4}
Opt. LRS Factor	0.5, 0.7, **0.9**	0.5, 0.7, **0.9**
Opt. LRS Patience	3, 5, **7**	**3**, 5, 7
Grad. Clip. Thres.	**0.2**, 1, 5	0.2, 1, **5**

Table 2
The hyperparameters values searched through during the training of the models using the Weights & Biases bayesian search. The values used for training the best performing models are in bold. The batch size used was the highest possible based on our GPU memory limit. The optimizer learning rate scheduler parameters are functionally meaningless, as both of the models achieved the best performance before the optimizer was triggered to lower the learning rate. Gradient clipping was added to prevent exploding gradient behavior occurring sometimes when a large starting learning rate was selected.

20 dBZ (corresponding to light rain) to differentiate between rain and no rain areas. This way, we can evaluate only the shape of precipitation features and disregard the intensity, which can serve as another valuable metric. Our experiments have shown that higher threshold values corresponding to extreme precipitation events show larger differences between model metrics during evaluation, however the informative value would be lower due to such events occurring only in the small minority of the test set observations.

5. 2D vs. 3D: A Comparison

The impact of providing a vertical dimension to the model was evaluated by comparing the error rate when predicting a single reflectivity map at constant altitude above radar. We trained the 2D model to output the next CAPPI radar reflectivity maps at 2 km above radar 30 minutes into the future based on past radar reflectivity maps at the same altitude. Subsequently, we trained a 3D model to predict equivalent 3D reflectivity maps at 8 altitude levels based on recent volumetric observation data. To evaluate which model is better at precipitation nowcasting, we evaluate the prediction error on a single CAPPI map at 2 km above radar from the target observation (nowcast 30 minutes in the future). This can be done because one slice of the output volume of the 3D model matches the altitude level the 2D model was trained on (2000 m.a.r.).

A simple euclidean persistence was used as a benchmark. This benchmark method simply copies the last input observation as the prediction output. Despite the method being trivial, the precipitation data is highly dependent on previous observations and so it provides a good performance benchmark. Using this benchmark, we can also evaluate the rate of change in the data and therefore see how ”difficult” it is to make an accurate

Model	MSE ↓	MAE ↓	Accuracy ↑	Precision ↑	Recall ↑	F1 ↑
Persistence	55.4110	4.7534	0.8307	0.6529	0.6426	0.6457
2D U-Net	22.6510	3.2623	0.8969	**0.8257**	0.7282	0.7696
3D U-Net	**22.0340**	**3.2124**	**0.9000**	0.8022	**0.7833**	**0.7894**

Table 3
Comparison of model results on the test set for each of the chosen metric scores. The ↓ symbol means it is a lower-is-better score, while the ↑ symbolizes a higher-is-better score. The best result for each score is bolded.

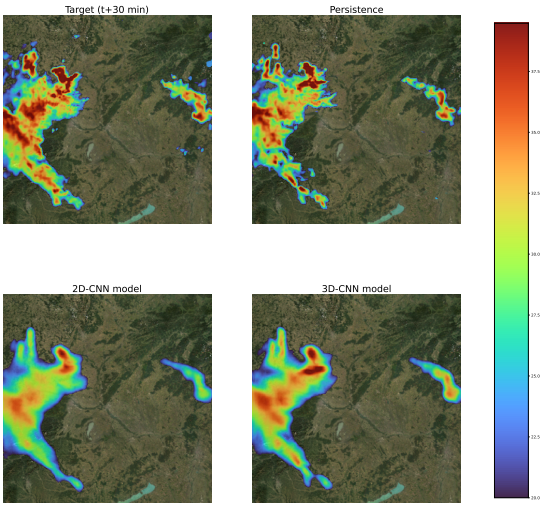

Figure 4: A visual comparison of nowcasts produced by the models for a random observation from the test set. Upper left image shows the target observation at 2 km CAPPI. Upper right is the benchmark persistence nowcast. Bottom left is the reference 2D-CNN U-Net model nowcast. Bottom right is the corresponding slice of our 3D-CNN volumetric U-Net model nowcast. While both U-Net models show the expected blurring, the volumetric model is affected less, with larger areas of high reflectivity (shown in dark red). This is desirable, as the model is better at predicting extreme events.

prediction for each sample.

The results in Table 3 show that the best 3D-CNN U-Net model slightly outperformed the best 2D-CNN counterpart. On average, the 3D model achieved lower prediction error on the test set, in both MSE and MAE metrics. The improvement is small, but statistically significant (paired t-test at 0.99 confidence level on test set MSE scores rejected the null hypothesis that the means of 2D and 3D model error scores are the same, p-value is very close to zero). The area-based metrics also show small improvements, with accuracy and F1 scores being slightly higher. Based on considerably higher recall and lower precision, we can assume the 3D model predicts larger precipitation bodies on average. See Figure 4 for a visual comparison of the model outputs.

6. Conclusion

Our research shows that providing additional information from multiple altitude levels has the potential to increase the nowcasting accuracy, as compared to the currently standard approach of using only 2-dimensional precipitation maps. The improvements in error metrics, while not groundbreaking, were statistically significant and show that providing more data is worth it, if we can afford the increase in model complexity and training time. Even a small reduction in prediction error can be beneficial in many applications and our preliminary results show that volumetric nowcasting can have a positive impact.

Additionally, volumetric nowcasts undoubtedly provide more value to the operators of these nowcasting systems. Reflectivity at different altitudes affects the true rainfall rate on the ground in different ways, which cannot be taken into account from simple 2-dimensional precipitation nowcasts. 3-dimensional predictions of future reflectivity observations can serve as a more valuable input to the consecutive models mapping the observed reflectivity to actual the rainfall rate on the ground.

While the field of precipitation nowcasting using neural networks is not new, there are still more uncertainties regarding best practices that should be comprehensively explored and compared. There are several open questions to answer, e.g.: Is it better to train the model directly on the captured reflectivity data or the data converted to rainfall rate? How many previous observations should be provided to the model? How to convert radar observations to actual rainfall on the ground as accurately as possible? These are just some of the interesting problems that need to be explored in the future.

Acknowledgments

This research was partially supported by TAILOR, a project funded by EU Horizon 2020 research and innovation programme under GA No 952215; by The Ministry of Education, Science, Research and Sport of the Slovak Republic under the Contract No. 0827/2021; and by Life Defender - Protector of Life, ITMS code: 313010ASQ6, co-financed by the European Regional Development Fund (Operational Programme Integrated Infrastructure).

References

[1] F. Schmid, Y. Wang, A. Harou, Nowcasting guidelines–a summary, Bulletin n° 68 (2019) 2.

[2] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, W.-c. Woo, Convolutional lstm network: A machine learning approach for precipitation nowcasting, Advances in neural information processing systems 28 (2015).

[3] G. Ayzel, T. Scheffer, M. Heistermann, Rainnet v1. 0: a convolutional neural network for radar-based precipitation nowcasting, Geoscientific Model Development 13 (2020) 2631–2644.

[4] K. Trebing, T. Stanczyk, S. Mehrkanoon, Smaat-unet: Precipitation nowcasting using a small attention-unet architecture, Pattern Recognition Letters 145 (2021) 178–186. URL: https://www.sciencedirect.com/science/article/pii/S0167865521000556. doi:https://doi.org/10.1016/j.patrec.2021.01.036.

[5] C. Wang, P. Wang, P. Wang, B. Xue, D. Wang, Using conditional generative adversarial 3-d convolutional neural network for precise radar extrapolation, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 14 (2021) 5735–5749.

[6] S. Ravuri, K. Lenc, M. Willson, D. Kangin, R. Lam, P. Mirowski, M. Fitzsimons, M. Athanassiadou, S. Kashem, S. Madge, et al., Skilful precipitation nowcasting using deep generative models of radar, Nature 597 (2021) 672–677.

[7] C. A. Doswell, Severe convective storms—an overview, Severe convective storms (2001) 1–26.

[8] USGS/Google, Usgs landsat 8 collection 1 tier 1 toa reflectance, 2022. URL: https://developers.google.com/earth-engine/datasets/catalog/LANDSAT_LC08_C01_T1_TOA.

[9] M. Dixon, G. Wiener, Titan: Thunderstorm identification, tracking, analysis, and nowcasting—a radar-based methodology, Journal of atmospheric and oceanic technology 10 (1993) 785–797.

[10] A. Hering, C. Morel, G. Galli, S. Sénési, P. Ambrosetti, M. Boscacci, Nowcasting thunderstorms in the alpine region using a radar based adaptive thresholding scheme, in: Proceedings of ERAD, volume 1, 2004.

[11] E. Ruzanski, V. Chandrasekar, Y. Wang, The casa nowcasting system, Journal of Atmospheric and Oceanic Technology 28 (2011) 640–655.

[12] T. Haiden, A. Kann, C. Wittmann, G. Pistotnik, B. Bica, C. Gruber, The integrated nowcasting through comprehensive analysis (inca) system and its validation over the eastern alpine region, Weather and Forecasting 26 (2011) 166–183.

[13] N. E. Bowler, C. E. Pierce, A. W. Seed, Steps: A probabilistic precipitation forecasting scheme which merges an extrapolation nowcast with downscaled nwp, Quarterly Journal of the Royal Meteorological Society: A journal of the atmospheric sciences, applied meteorology and physical oceanography 132 (2006) 2127–2155.

[14] S. Pulkkinen, D. Nerini, A. A. Pérez Hortal, C. Velasco-Forero, A. Seed, U. Germann, L. Foresti, Pysteps: an open-source python library for probabilistic precipitation nowcasting (v1.0), Geoscientific Model Development 12 (2019) 4185–4219. URL: https://gmd.copernicus.org/articles/12/4185/2019/. doi:10.5194/gmd-12-4185-2019.

[15] X. Shi, Z. Gao, L. Lausen, H. Wang, D.-Y. Yeung, W.-k. Wong, W.-c. Woo, Deep learning for precipitation nowcasting: A benchmark and a new model, Advances in neural information processing systems 30 (2017).

[16] S. Agrawal, L. Barrington, C. Bromberg, J. Burge, C. Gazen, J. Hickey, Machine learning for precipitation nowcasting from radar images, CoRR abs/1912.12132 (2019). URL: http://arxiv.org/abs/1912.12132. arXiv:1912.12132.

[17] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, CoRR abs/1505.04597 (2015). URL: http://arxiv.org/abs/1505.04597. arXiv:1505.04597.

[18] C. Keil, G. C. Craig, A displacement and amplitude score employing an optical flow technique, Weather and Forecasting 24 (2009) 1297 – 1308. URL: https://journals.ametsoc.org/view/journals/wefo/24/5/2009waf2222247_1.xml. doi:10.1175/2009WAF2222247.1.

[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Advances in neural information processing systems 27 (2014).

[20] W. Zhang, R. Zhang, H. Chen, G. He, Y. Ge, L. Han, A multi-channel 3d convolutional-recurrent neural network for convective storm nowcasting, in: 2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS, IEEE, 2021, pp. 363–366.

[21] J. J. Helmus, S. M. Collis, The python arm radar toolkit (py-art), a library for working with weather radar data in the python programming language, Journal of Open Research Software 4 (2016).

[22] J. S. Marshall, W. M. K. Palmer, The distribution of raindrops with size, Journal of Atmospheric Sciences 5 (1948) 165 – 166. doi:10.1175/1520-0469(1948)005<0165:TDORWS>2.0.CO;2.

[23] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, Pytorch: An impera-

tive style, high-performance deep learning library, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[24] L. Biewald, Experiment tracking with weights and biases, 2020. URL: https://www.wandb.com/, software available from wandb.com.