EfficientTempNet: Temporal Super-Resolution of Radar Rainfall

Bekir Z Demiray
IIHR — Hydroscience & Engineering
University of Iowa
bekirzahit-demiray@uiowa.edu

Muhammed Sit
IIHR — Hydroscience & Engineering
University of Iowa
muhammed-sit@uiowa.edu

Ibrahim Demir
IIHR — Hydroscience & Engineering
University of Iowa
ibrahim-demir@uiowa.edu

Abstract

Rainfall data collected by various remote sensing instruments such as radars or satellites has different space-time resolutions. This study aims to improve the temporal resolution of radar rainfall products to help with more accurate climate change modeling and studies. In this direction, we introduce a solution based on EfficientNetV2, namely EfficientTempNet, to increase the temporal resolution of radar-based rainfall products from 10 minutes to 5 minutes. We tested EfficientRainNet over a dataset for the state of Iowa, US, and compared its performance to three different baselines to show that EfficientTempNet presents a viable option for better climate change monitoring.

1 Introduction

The importance of environmental data has grown significantly in recent years due to the increased impact of natural disasters. Rainfall data, in particular, plays a crucial role in various climate modeling applications, such as flood forecasting Sit et al. (2021a); Xiang & Demir (2022), monitoring water quality Jha et al. (2007), or managing wastewater Cahoon & Hanke (2017). Given that the spatial and temporal patterns of rain are crucial in these modeling efforts, having reliable and accessible precipitation maps is vital for advancing research on climate related hazards with different objectives like risk assessment Alabbad et al. (2021) or disaster mitigation Alabbad et al. (2022).

Quantitative Precipitation Estimation (QPE) systems provide rainfall data that takes into account three dimensions, which include latitude and longitude as the spatial coordinates and temporal resolution as the third dimension. Weather radars are the primary source used in QPE, and they allow to record the space-time characteristics of precipitation, which is essential for making accurate streamflow predictions in hydrology. Improving rainfall datasets in radar hydrology largely involves addressing uncertainty factors, while the focus is on acquiring more precise precipitation data to enhance our understanding of weather patterns in terms of space and time. However, once the data is obtained, the task of creating better datasets becomes a separate challenge.

The temporal resolution of rainfall data is a critical factor in determining the accuracy of predictive modeling efforts (e.g., Atencia et al. (2011)). This paper aims to address the issue of low temporal resolution rainfall products by proposing a convolutional neural network to enhance the temporal resolution of rainfall data. The proposed CNN model, EfficientTempNet, is based on EfficientNetV2 Tan & Le (2021), and the performance of the network is compared to three different methods: the nearest frame, optical flow, and TempNet Sit et al. (2021b).

1.1 Related Work

Rainfall products aren’t the only data that is subject to temporal interpolation between two 2D maps. Various studies in computer vision literature were presented employing neural networks based approaches for video frame interpolation Niklaus et al. (2017a;b); Liu et al. (2017); Jiang et al. (2018). Conversely, the literature about how to interpolate time in rainfall datasets is limited. In Seo & Krajewski (2015), researchers used advection correction to create 1-minute rainfall maps from 5-minute ones. Building upon Seo & Krajewski (2015), in Sit et al. (2021b), a residual but simple CNN architecture, namely TempNet, was proposed and was shown to be superior to the dense optical flow-based advection correction method and a non-residual CNN over the IowaRain Sit et al. (2021c) dataset. To the best of our knowledge, TempNet is the only study that tackles the problem of temporal super-resolution of radar rainfall products using neural networks, thus forming the baseline of this study. Consequently, in a similar fashion, in this study, the performance of EfficientTempNet will be presented over the IowaRain dataset to show the increment in performance.

The paper is organized as follows: Section 2 introduces the dataset used and provides an overview of the methodology. Section 3 presents and discusses the preliminary results of the comparison between all methods. Finally, Section 4 summarizes the findings.

2 Methods

2.1 Data

IowaRain Sit et al. (2021c); Seo & Krajewski (2020) is a rainfall event dataset that covers the years of 2016 and 2019 with 5-minute temporal resolution. The dataset covers an area that bounds the state of Iowa and has the size of 1088x1760 pixels with 500m of spatial resolution. In order to meet memory bottlenecks and computational complexity drawbacks, we sampled an area within the IowaRain domain from eastern Iowa that is 768x768 in size; then we averaged the values in the area to downscale the rainmap spatial resolution to 3 km, which effectively changed the rainfall map sizes to 128x128. After the subsampling, the event detection criteria of IowaRain were applied to the new area; 82, 83, 90, and 110 events were obtained for each year in 2016 and 2019, respectively. To get a more approximate 70/30 split when dividing the dataset, we used the rainfall events in 2019 as the test set and all the rainfall events before 2019 as the train set, for total set sizes of 255 and 110 events. For each snapshot, or 2D map, t_s from a rain event, a snapshot immediately following it t_{s+5} and immediately preceding it t_{s-5} was converted into its own entry in the dataset. In the end, for each dataset entry or sample, there were three 2D rainfall maps, for t_{s-5}, t_{s+5}, and t_s, the first two being inputs and the last one being the output (Figure 1). In the end, there were 19,264 train and 7,762 test entries used in this study.

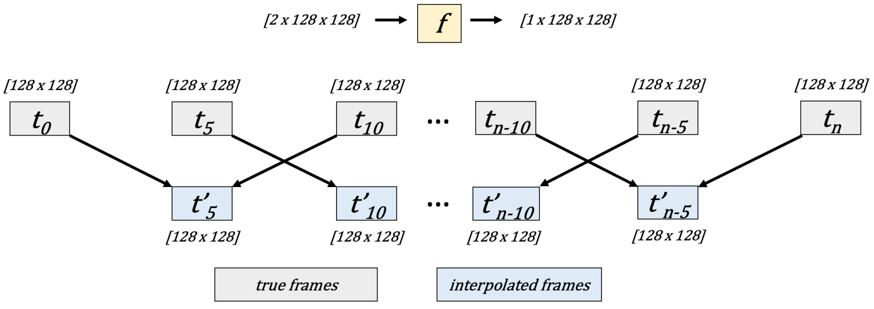

Figure 1: Input/Output shapes and problem definition for EfficientTempNet

2.2 EfficientTempNet

The foundation of this work is based on EfficientNetV2 Tan & Le (2021). In order to create a model that takes two rainfall maps and outputs another one with the same size, EfficientNetV2 is altered by modifying its key component, MBConv block Sandler et al. (2018), which was not

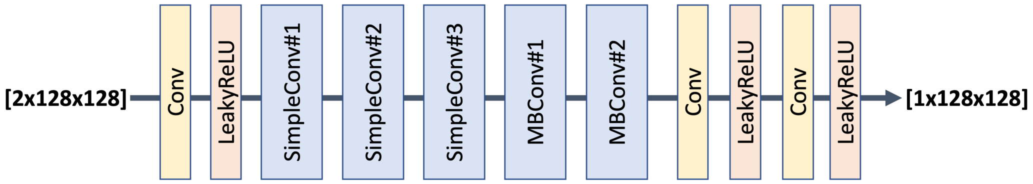

Figure 2: Architecture of Proposed Method

specifically developed with this task in mind. Our model takes two 2D rain map and combines them prior to passing them into a convolutional layer with 24 feature maps. After this layer, multiple SimpleConv and MBConv blocks are used to extract information. In the last section of the model, two convolutional layers help to get the desired output. Our model is depicted visually in Figure 2.

In EfficientNetV2, there are two different blocks, namely MBConv and Fused-MBConv. In the proposed method, we modify the MBConv blocks, and instead of Fused-MBConv blocks, SimpleConv blocks were used, as our experiments during model development favored them over Fused-MBConv blocks. In our MBConv blocks, batch normalization layers are removed, and activation layers are changed with LeakyReLU. The remaining parts of the MBConv are the same as the original implementation, including the SE layer and depthwise convolutions. In our SimpleConv blocks, two convolutional layers with kernel sizes 3 and 1 are used with LeakyReLU in between them. Similar to the MBConv block, SimpleConv blocks take advantage of residual connections. In addition to these, input size doesn't change throughout the model. Details of SimpleConv and MBConv blocks are provided in Table 1.

Table 1: Blocks' details in our model

	SimpleConv#1	**SimpleConv#2**	**SimpleConv#3**	**MBConv#1**	**MBConv#2**
# of Channels	24	48	64	128	256
Expansion ratio	1	4	4	4	6
# of Layers	2	4	3	4	8

The RAdam optimizer Liu et al. (2019) was employed for training with a learning rate of 0.001, and the Mean Absolute Error as the loss function and for evaluation. The implementation of the network was done using Pytorch 1.9 Paszke et al. (2019), and the ReduceLROnPlateau scheduler was utilized to adjust the learning rate downward if there was no progress in reducing the model's loss over three consecutive epochs. The training was performed using NVIDIA Titan V GPUs.

3 Results

This section outlines the metrics used to evaluate performance and presents the results of our methods as well as three baselines: nearest frame, optical flow, and TempNet. First, we will give information about compared methods, then describe the metrics, and finalize the section with scores and discussion.

Nearest Frame - The nearest frame involves assuming that the interpolated frame is equal to the closest frame in time to the forecasted frame. In our case, we decided to select the predecessor frame as the closest, although both frames are in the same proximity.

Optical Flow - Although a variety of optical flow calculation algorithms can be found in the computer vision literature, the Gunnar-Farneback optical flow was used in this paper Farnebäck (2003). Each value's pixel intensity was determined using the Gunnar-Farneback optical flow. For the rain map scenario, this would entail figuring out the shifts in each measurement over the course of the two-dimensional rain map's 3km by 3km grid. Once the optical flow is computed, all measurements are shifted between frames based on their position in the first frame and their motion vectors in the optical flow. NumPy Harris et al. (2020), a library for numerical computation, and

OpenCV Bradski (2000), a library for computer vision, were utilized in implementation.

TempNet - TempNet has done a similar job to ours. The study is powered by three components, and the design is completed with residual connections. More details on the TempNet can be found here Sit et al. (2021b).

We measured the performance of each of the previously mentioned methods using four metrics, namely, Mean Absolute Error (MAE), Probability of Detection (POD), and False Alarm Ratio (FAR) and Critical Success Index (CSI). The MAE calculates the average of the absolute differences between the estimated and actual 2D rain maps in the test dataset. The POD 1, FAR 2, and CSI 3 metrics are calculated using the number of hits (H), false alarms (F), and misses (M), respectively, in a binary manner. H represents the number of correctly estimated rainfall cells, meaning the number of elements in the 2D map that were correctly estimated as non-zero values. F represents the number of wrongly estimated rainfall cells, where the cells were estimated to have rain, but the corresponding indices in the ground truth 2D map had zero. M represents the number of rainfall cells that were estimated as zero but had non-zero values in the ground truth. It's important to note that a value of 1.0 is best for POD and CSI, whereas a value of 0.0 is best for FAR. All metrics were calculated using a threshold value of 0.0001 over the estimated rainfall maps, since the neural networks would produce small non-zero values throughout the estimated 2D rainfall maps.

$$POD = \frac{H}{H+M} \quad (1) \qquad FAR = \frac{F}{H+F} \quad (2) \qquad CSI = \frac{H}{H+F+M} \quad (3)$$

Table 2: Performance summary of tested methods for predicting intermediate frame

Methodology	**MAE(mm/h)↓**	**CSI ↑**	**POD ↑**	**FAR ↓**
Nearest Frame	0.630	0.840	0.911	0.0878
Optical Flow	0.324	0.857	**0.978**	0.127
TempNet	0.272	0.898	0.919	**0.0252**
EfficientTempNet	**0.208**	**0.923**	0.972	0.0524

Table 2 presents the performance of the baselines as well as the CNN-based models on the test set in terms of described metrics for the interpolation of the frame at t_s from frames at t_{s-5} and t_{s+5}. As shown in Table 2, both TempNet and EfficientTempNet outperform the baseline methods in terms of the MAE metric, which was used to train them. Between EfficientTempNet and TempNet, EfficientTempNet shows more strong results with a significant margin in MAE scores as a result of an increase in model size. As for the POD, optical flow provides the best result with a small margin in comparison to EfficientTempNet. However, with consideration of FAR and POD together, it can be argued that optical flow tends to generate non-zero values that results in correctly identifying true positives but also causes a higher number of false positives compared to EfficientTempNet. Overall, in improving temporal resolution of radar rainfall products, the EfficientTempNet offers a better solution compared to other employed methods since its scores are either best or runner-up over the used metrics.

4 Conclusion

In this study, the EfficientNetV2-based CNN model, EfficientTempNet, was introduced for improving the temporal resolution of radar rainfall products and compared to three additional methods. The results showed that EfficientTempNet outperformed the other approaches in terms of various performance metrics, including MAE. This work represents significant progress in creating improved rainfall maps for various purposes, such as flood forecasting Xiang et al. (2021); Sit & Demir (2019) and climate change modeling Rolnick et al. (2022); Sit et al. (2020).

REFERENCES

Yazeed Alabbad, Jerry Mount, Ann Melissa Campbell, and Ibrahim Demir. Assessment of transportation system disruption and accessibility to critical amenities during flooding: Iowa case study. *Science of the total environment*, 793:148476, 2021.

Yazeed Alabbad, Enes Yildirim, and Ibrahim Demir. Flood mitigation data analytics and decision support framework: Iowa middle cedar watershed case study. *Science of The Total Environment*, 814:152768, 2022.

A Atencia, L Mediero, MC Llasat, and Luis Garrote. Effect of radar rainfall time resolution on the predictive capability of a distributed hydrologic model. *Hydrology and Earth System Sciences*, 15 (12):3809–3827, 2011.

Gary Bradski. The opencv library. *Dr. Dobb's Journal: Software Tools for the Professional Programmer*, 25(11):120–123, 2000.

Lawrence B Cahoon and Marc H Hanke. Rainfall effects on inflow and infiltration in wastewater treatment systems in a coastal plain region. *Water Science and Technology*, 75(8):1909–1921, 2017.

Gunnar Farnebäck. Two-frame motion estimation based on polynomial expansion. In *Image Analysis: 13th Scandinavian Conference, SCIA 2003 Halmstad, Sweden, June 29–July 2, 2003 Proceedings 13*, pp. 363–370. Springer, 2003.

Charles R Harris, K Jarrod Millman, Stéfan J Van Der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J Smith, et al. Array programming with numpy. *Nature*, 585(7825):357–362, 2020.

Manoj K Jha, Philip W Gassman, and Jeffrey G Arnold. Water quality modeling for the raccoon river watershed using swat. *Transactions of the ASABE*, 50(2):479–493, 2007.

Huaizu Jiang, Deqing Sun, Varun Jampani, Ming-Hsuan Yang, Erik Learned-Miller, and Jan Kautz. Super slomo: High quality estimation of multiple intermediate frames for video interpolation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9000–9008, 2018.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. On the variance of the adaptive learning rate and beyond. *arXiv preprint arXiv:1908.03265*, 2019.

Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *Proceedings of the IEEE international conference on computer vision*, pp. 4463–4471, 2017.

Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 670–679, 2017a.

Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *Proceedings of the IEEE international conference on computer vision*, pp. 261–270, 2017b.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

David Rolnick, Priya L Donti, Lynn H Kaack, Kelly Kochanski, Alexandre Lacoste, Kris Sankaran, Andrew Slavin Ross, Nikola Milojevic-Dupont, Natasha Jaques, Anna Waldman-Brown, et al. Tackling climate change with machine learning. *ACM Computing Surveys (CSUR)*, 55(2):1–96, 2022.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Bong-Chul Seo and Witold F Krajewski. Correcting temporal sampling error in radar-rainfall: Effect of advection parameters and rain storm characteristics on the correction accuracy. *Journal of Hydrology*, 531:272–283, 2015.

Bong-Chul Seo and Witold F Krajewski. Statewide real-time quantitative precipitation estimation using weather radar and nwp model analysis: Algorithm description and product evaluation. *Environmental Modelling & Software*, 132:104791, 2020.

Muhammed Sit and Ibrahim Demir. Decentralized flood forecasting using deep neural networks. *arXiv preprint arXiv:1902.02308*, 2019.

Muhammed Sit, Bekir Z Demiray, Zhongrun Xiang, Gregory J Ewing, Yusuf Sermet, and Ibrahim Demir. A comprehensive review of deep learning applications in hydrology and water resources. *Water Science and Technology*, 82(12):2635–2670, 2020.

Muhammed Sit, Bekir Demiray, and Ibrahim Demir. Short-term hourly streamflow prediction with graph convolutional gru networks. *arXiv preprint arXiv:2107.07039*, 2021a.

Muhammed Sit, Bong-Chul Seo, and Ibrahim Demir. Tempnet–temporal super resolution of radar rainfall products with residual cnns. *arXiv preprint arXiv:2109.09289*, 2021b.

Muhammed Sit, Bong-Chul Seo, and Ibrahim Demir. Iowarain: A statewide rain event dataset based on weather radars and quantitative precipitation estimation. *arXiv preprint arXiv:2107.03432*, 2021c.

Mingxing Tan and Quoc Le. Efficientnetv2: Smaller models and faster training. In *International conference on machine learning*, pp. 10096–10106. PMLR, 2021.

Zhongrun Xiang and Ibrahim Demir. Fully distributed rainfall-runoff modeling using spatial-temporal graph neural network. 2022. doi: 10.31223/X57P74.

Zhongrun Xiang, Ibrahim Demir, Ricardo Mantilla, and Witold F Krajewski. A regional semi-distributed streamflow model using deep learning. 2021. doi: 10.31223/X5GW3V.