EasyRain: A User-Friendly Platform for Comparing Precipitation Nowcasting Models

Ji Cheng*[1], Guimu Guo#[2], Da Yan#[3], Xiaotian Hao*[4], Wilfred Ng*[5]

*Department of Computer Science and Engineering, The Hong Kong University of Science and Technology
{[1]jchengac, [4]xhao, [5]wilfred}@cse.ust.hk

#Department of Computer Science, The University of Alabama at Birmingham
{[2]guimuguo, [3]yanda}@uab.edu

Abstract—**Precipitation nowcasting, which predicts rainfall intensity in the near future, has been studied by meteorologists for decades. Currently, computer vision techniques, especially optical flow based methods, are widely adopted by observatories since they deliver reasonable performance without the need of model training. However, their performance is highly sensitive to model parameters which require a lot of empirical knowledge to optimize. With the recent success of deep learning (DL), machine learning researchers have started to explore the use of spatiotemporal DL models for precipitation nowcasting, which have demonstrated a better performance than optical flow based methods. However, DL models are not easy to configure for non-DL experts such as meteorologists. In this poster, we introduce EasyRain, a platform with a user-friendly web interface to help users without domain knowledge (in DL and/or meteorology) to efficiently build DL and optical flow based models. We will demonstrate the efficiency and usability of EasyRain for training, tuning, and comparing precipitation nowcasting models.**

I. Introduction

The goal of precipitation nowcasting is to predict the future rainfall intensity in a local region over a relatively short period of time based on radar echo maps [4]. The radar echo maps are Constant Altitude Plan Position Indicator (CAPPI) images that encode the rainfall density information at each local region and can be converted to rainfall intensity maps using either Marshall-Palmer relationship or Z-R relationship [2]. As illustrated in Figure 1, the problem takes several consecutive frames of radar echo maps as the input, and output the predicted future radar echo maps. The obtained radar echo maps can then be further converted into rainfall intensity maps.

This problem is of great significance since heavy rain may cause natural disasters such as flooding and landslide, which may in turn lead to infrastructure failures, traffic congestion, and mass casualties in extreme cases.

Two types of methods have been proposed by meteorology and machine learning research communities, respectively. **(1) Optical flow based models,** as represented by the *Real-time Optical flow by Variational methods for Echoes of Radar* (ROVER) [7] algorithm. **(2) Deep learning models,** where the state-of-the-arts are sequence-to-sequence DL models with novel RNN (recurrent neural network) components such as ConvLSTM [3], ConvGRU and TrajGRU [4].

As the abstract has explained, both methods have their pros and cons. It is challenging for meteorologists to compare the

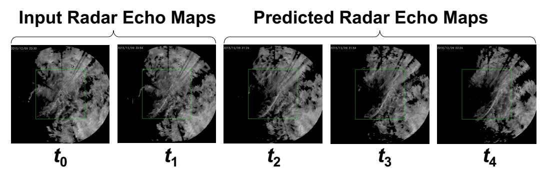


Fig. 1. Illustration of Precipitation Nowcasting

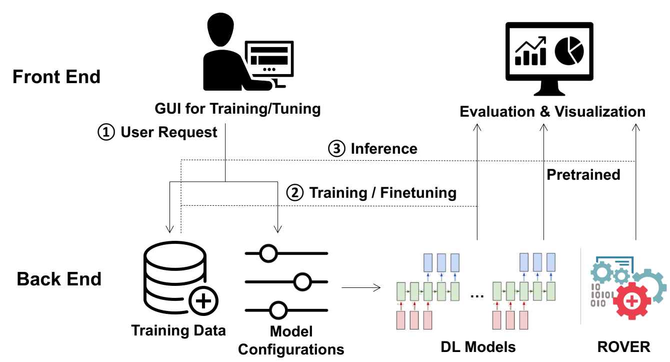


Fig. 2. EasyRain System Architecture

performance of optical flow based methods with deep learning (DL) based methods, as it is not a trivial task for scientists without DL experience to configure and run deep learning models. To help users efficiently build and tune precipitation nowcasting models, we propose a platform called EasyRain with a user-friendly browser-based graphical user interface (GUI). EasyRain allows meteorologists to easily train and tune DL models, to tune parameters of optical flow based methods like ROVER, and to compare different models. Users can dynamically change the model parameters with performance feedback of interactive speed; users can also compare the performance of selected models in both quantitative and qualitative manners using visualization plots. A rich set of models have been pretrained on the HKO-7 dataset [4] for users to choose from; users may further finetune a pretrained DL model on their own dataset by initializing the parameters to be those of the pretrained model, to achieve faster convergence.

Figure 2 summarizes our EasyRain platform consisting of (1) a front-end web interface for users to easily train and make inference with DL models and ROVER, and to visualize the results for model comparison; (2) a back end that automatically manages data splitting (into training and validation sets), efficient training of DL models, and result evaluation.

978-1-7281-0858-2/19/$31.00 © 2019 IEEE

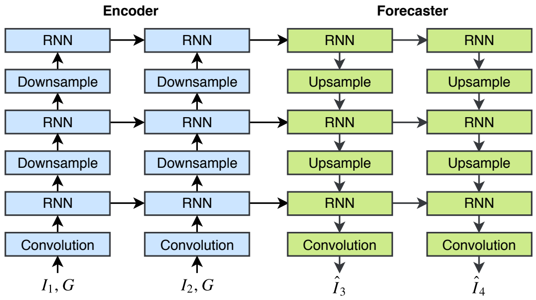


Fig. 3. The Encoding-Forecasting Network Structure [4]

II. THE EASYRAIN PLATFORM

As Figure 2 shows, at the front end of EasyRain, a user may upload their datasets for training and/or inference, and specify/tune model hyperparameters via a friendly browser-based graphical interface. EasyRain maintains three DL models: ConvGRU, TrajGRU and 3D CNN which are pretrained on the HKO-7 dataset [4], and users may directly use a pretrained DL model to make inference on their uploaded dataset. Optionally, users may train a new model from their uploaded training dataset, to be used subsequently for making inference on their uploaded test dataset.

In the latter scenario, one may train a new DL model from scratch, e.g., by specifying weight initialization methods such as random Gaussian weights, Xavier/He initialization. Users may also choose to finetune the parameters of a pretrained DL model over their own training dataset to allow faster convergence, in which case the model parameters start from those already trained over HKO-7. Our user interface provides timely feedback on the current training loss and epoch/iteration number for users to track the training progress.

In contrast, ROVER directly takes user-specified parameters for inference and does not require training; and therefore, users can directly tune the model parameters in the browser, and visualize the dynamic feedback of the model performance.

EasyRain also provides an intuitive interface to compare the quantitative and qualitative model performance results among DL models and ROVER. In the rest of this section, we will introduce in Section II-A our built-in models that users can select for precipitation nowcasting; then, Section II-B explains how the model performance are evaluated and compared.

A. Built-in Models

ConvLSTM and ConvGRU. The pioneering work of [3] adapts the sequence-to-sequence RNN-based prediction model to conduct precipitation nowcasting, by extending RNN to have convolutional structures in both the input-to-state and state-to-state transitions so as to accommodate radar echo maps as model inputs.

As an illustration, Figure 3 shows the adopted encoding-forecasting network structure. In the figure, three layers of RNNs are used to predict two future frames $\hat{I}_3, \hat{I}_4$ given two input frames I_1, I_2. The encoding network compresses the input sequence into hidden state tensors and the forecasting network then unfolds them to give the predicted frames.

Shi et al. adopt convolutional LSTM[1] (ConvLSTM) [3] as the RNN component in Figure 3; later, Shi et al. [4] further replaces LSTM with another RNN variant GRU[2]. Both LSTM and GRU overcomes the problem of vanishing gradients in RNN, but GRU has less parameters and is thus computationally more efficient; GRU extended with convolutional structures produces ConvGRU which is computationally more efficient than ConvLSTM and is thus adopted in EasyRain.

TrajGRU. The limitation of ConvLSTM and ConvGRU is that the connection structure and weights are fixed for all the locations in a receptive field. However, natural motion and transformation (e.g., rotation) in radar echo maps are location-variant in general. To overcome this limitation, [4] proposed a location-variant convolutional RNN component called TrajGRU where the recurrent connections between consecutive frames are dynamically determined. TrajGRU is demonstrated in [4] to give better performance without increasing the number of parameters.

3D CNN[3]. CNN-based models are known to be faster to train than RNN-based models. Tran et al. [5] used 3D CNN to encode temporal information as depth of the input in the task of video action recognition. In 3D CNN, convolution and pooling operations are performed spatio-temporally, which is in contrast to 2D CNN where they are done only spatially. 3D CNN is demonstrated to be effective in [5] in modeling temporal information; 3D CNN has also been used to perform prediction task on videos with fractionally-strided convolutions in [6], which is similar to our task of precipitation nowcasting (i.e., a video of radar echo maps).

ROVER. Conventional precipitation nowcasting methods use optical flow[4] to accurately extrapolate the future radar echo maps. One such example is ROVER [7] which is the state-of-the-art optical flow based method currently used at the Hong Kong Observatory (HKO). ROVER first generates flow fields through optical flow calculation [1] and converts them to the prediction through semi-Lagrangian advection. ROVER requires six optical flow parameters and the choice of the parameters could significantly affect the quality of a predicted radar echo map. While empirically-set parameters that work well for Hong Kong are provided as default in EasyRain, parameter tuning is often required to achieve optimal performance on radar echo maps from a different region.

Pretrained Models. EasyRain provides 3 pretrained DL models ConvGRU, TrajGRU, and 3D CNN to users, whose hyperparameters have been tuned to demonstrate good performance on the HKO-7 dataset. For ROVER, we use widely accepted parameters currently used in Hong Kong Observatory.

B. Model Evaluation

Qualitative Performance. Radar echo maps are usually recorded every several minutes. EasyRain allows users to view

[1]LSTM stands for long short-term memory.
[2]GRU stands for Gated Recurrent Unit.
[3]CNN stands for convolutional neural network.
[4]Optical flow is a technique in computer vision.

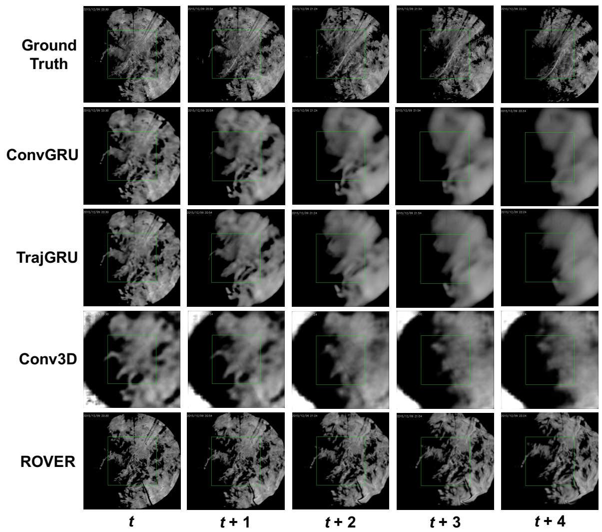
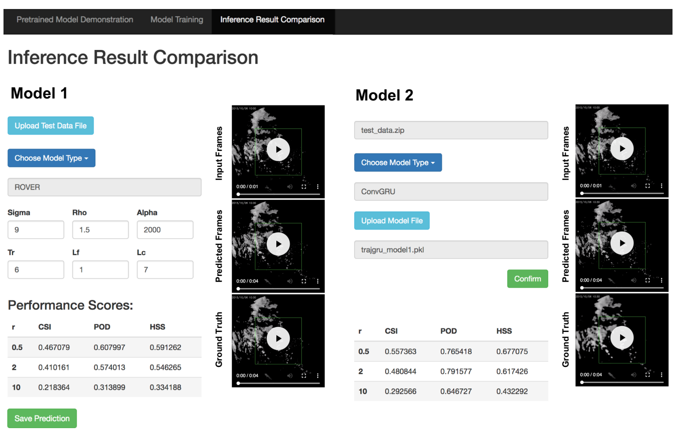


Fig. 4. Video Frames from Different Models and the Ground Truth



r	CSI	POD	HSS
0.5	0.467079	0.607997	0.591262
2	0.410161	0.574013	0.546265
10	0.218364	0.313899	0.334188

r	CSI	POD	HSS
0.5	0.557363	0.765418	0.677075
2	0.480844	0.791577	0.617426
10	0.292566	0.646727	0.432292

Fig. 5. GUI for Inference Result Comparison

a sequence of radar echo maps (or simply, frames) as a video; the predicted frames of different models can be juxtaposed as videos along with the video of ground-truth frames for comparison, as illustrated in Figure 4. When the videos are played together, their frames are aligned in time which gives users an intuitive feeling of how the prediction differs from the ground truth as time goes by.

Quantitative Performance. Each predicted (or ground-truth) frame will be converted into a rainfall density map by applying Marshall-Palmer relationship [2]. Then, each pixel in the rainfall density map is converted into 0 (for not raining) or 1 (for raining) using a cutoff threshold which is typically set as 0.5 mm/hr. In a nutshell, each predicted frame is converted into a 0/1 matrix. The 0/1 matrix computed from a predicted frame at time t can be evaluated against that computed from the ground-truth frame at time t by calculating well-established quantitative evaluation metrics [3], [4] including Critical Index Score CSI, Probability of Detection POD, and Heidke Skill Score HSS. Figure 5 illustrates how users may compare two different models in EasyRain, where the 3 quantitative scores CSI, POD and HSS are presented.

III. User Web Interface

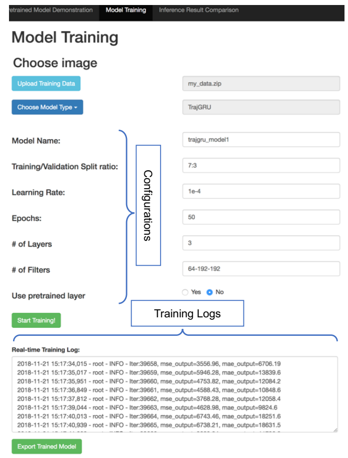


Fig. 6. GUI for Model Training

Figure 6 shows our DL model training interface on the left. To start training, a user needs to upload their own training dataset, select a DL model, and specify its hyperparameters. Finally, the user may click the "Start Training!" button to begin the training process. When the training finishes, the user may further click the "Export Trained Model" button to save the trained model to a Python pickle file. During training, the current loss value will be reported after each epoch.

As Figure 5 shows, to start a comparison, a user first needs to upload a test dataset containing a sequence of radar echo maps, and then to select the types of models to compare. If the model is a DL one, the user needs to specify the trained model which is stored as a pickle file; otherwise, ROVER is used, and the user needs to input the 6 parameters of ROVER. Finally, the user may click the "Confirm" button to start the inference and evaluation. Once inference is completed, quantitative scores CSI, POD and HSS will be presented in a table, and the input frames (e.g., before time t), the predicted frames (e.g., since time t) and their corresponding ground-truth frames will be presented as videos for intuitive comparison.

Instead of using a DL model trained by users themselves, EasyRain also provides pretrained models on the HKO-7 dataset. In this case, users choose a pretrained model rather than specify a model pickle file.

IV. Conclusion

EasyRain allows users to easily train and make inferences with DL models and ROVER for precipitation nowcasting. Models are compared in quantitative and qualitative manners.

Acknowledgment. This project is partially supported by NSF OAC-1755464 and NSF DGE-1723250.

References

[1] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. In *ECCV*, volume 3024 of *Lecture Notes in Computer Science*, pages 25–36, May 2004.

[2] J. S. Marshall and W. M. Palmer. The Distribution of Raindrops with Size. *Journal of Atmospheric Sciences*, 5:165–166, Aug. 1948.

[3] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NIPS*, pages 802–810. 2015.

[4] X. Shi, Z. Gao, L. Lausen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. WOO. Deep learning for precipitation nowcasting: A benchmark and a new model. In *NIPS*, pages 5617–5627. 2017.

[5] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, pages 4489–4497, 2015.

[6] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *NIPS*, pages 613–621, 2016.

[7] W.-c. Woo and W.-k. Wong. Operational application of optical flow techniques to radar-based rainfall nowcasting. *Atmosphere*, 8(3):48, 2017.