

RNN-based Pedestrian Crossing Prediction using Activity and Pose-related Features

J. Lorenzo¹, I. Parra¹, F. Wirth², C. Stiller², D. F. Llorca¹ and M. A. Sotelo¹

Abstract—Pedestrian crossing prediction is a crucial task for autonomous driving. Numerous studies show that an early estimation of the pedestrian’s intention can decrease or even avoid a high percentage of accidents. In this paper, different variations of a deep learning system are proposed to attempt to solve this problem. The proposed models are composed of two parts: a CNN-based feature extractor and an RNN module. All the models were trained and tested on the JAAD dataset. The results obtained indicate that the choice of the features extraction method, the inclusion of additional variables such as pedestrian gaze direction and discrete orientation, and the chosen RNN type have a significant impact on the final performance.

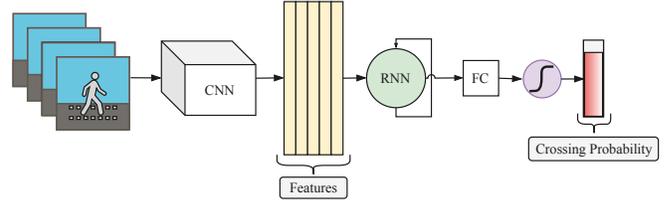


Fig. 1: Diagram describing the proposed method.

I. INTRODUCTION & RELATED WORK

According to the World Health Organization (WHO) [1], several efforts have been made in order to improve road safety, keeping the number of road deaths constant concerning the increase in both population and motorization.

The report additionally states that more than half of the reported fatalities are of Vulnerable Road Users (VRUs), with pedestrians and cyclists representing 26% of all deaths. According to the EU [2], 37% of road fatalities were in urban environments, and an additional two billion people are expected to be living in those areas by 2045, aggravating the problem [1].

Over the last decade, autonomous driving systems have evolved mainly due to the advent of Deep Learning. However, while tasks such as object classification and localization [3] have been significantly developed and improved, the understanding of the environment continues to be a challenging problem. Referring to pedestrians, the ability to predict pedestrian crossing action in urban scenarios can help in the planning strategy, achieving a smoother and more human-like autonomous driving. Moreover, as it is explained in [4], an improvement in the anticipation time can lead to a considerable reduction of possible pedestrian injuries.

There are two main approaches related to pedestrian crossing prediction. The first one is the human motion-based approach. These methods try to infer pedestrian intention employing dynamics, whether using information extracted from 3D pose [5], from image data and 3D position [6] or using optical flow information [7]. For a detailed overview of human motion methods, see the survey by Rudenko et

al. [8]. Nonetheless, due to pedestrian complex dynamic behavior, position forecasting must be supported by additional information such as context-related. In [9], authors based prediction on dynamic context variables such as distance to the car, and distance to the curb, in order to cope with sudden changes in dynamics. However, even if the prediction error decreases, the position forecasting does not take into account the majority of environmental variables which could affect the decision of the pedestrian. A higher-level approach, closer to the way the driver’s mind works, tries to simplify the problem and, at the same time, capture information about the context and environment in a non-supervised way. This second approach, based on action classification, tries to simplify the problem by getting closer to the driver’s way of inferring pedestrian intention. One way to pursue this approach is by using Convolutional Neural Networks (CNNs). For example, in [10] a pre-trained CNN model is fine-tuned for this task. In the field of video action recognition, 3D CNNs have recently become also popular [11]. As an example of the use of this architecture, in [12], a 3D CNN spatio-temporal model is used together with an object detector and a tracking algorithm achieving real-time performance at 20 fps.

In this paper, we propose a method for prediction of pedestrian crossing intention in one or more timesteps in the future, using data extracted from color videos recorded from inside a vehicle. Furthermore, additional information related to orientation, looking/gaze direction, movement state and image coordinates is used and compared with the model based solely on video input to observe possible improvements. The rest of the article is organized as follows: section II describes the proposed algorithms and the different architectures used. Experimental setup, including dataset selection, preprocessing methodology and experiments description, is detailed in section III. In section IV, experimental results are presented and discussed. Conclusions and future works are described in section V.

¹ Department of Computer Engineering, Universidad de Alcalá, Madrid, Spain {javier.lorenzod, ignacio.parra, david.fernandezl, miguel.sotelo}@uah.es

² Institut für Mess- und Regelungstechnik, Karlsruher Institut für Technologie, Karlsruhe, Germany {stiller, florian.wirth}@kit.edu

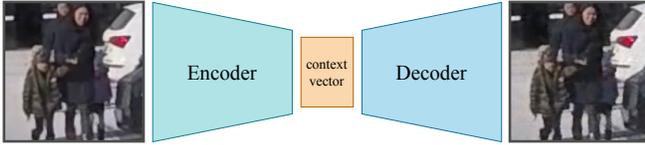


Fig. 2: High-level diagram describing a Convolutional Autoencoder. Output image example corresponds to SegNet-based autoencoder.

II. SYSTEM DESCRIPTION

The significant development of Deep Learning during the last decade has propelled the use of several variants of neural networks. In this work, two of these variants have been used: CNNs, used to extract features from pedestrian image sequences and Recurrent Neural Networks (RNNs), utilized to extract temporal information from these features. The proposed deep learning systems try to answer the question “Is the pedestrian going to cross the street?” by approaching it as a sequence binary classification problem where we try to infer the intention in a future time horizon given an input sequence. In the next subsections, both the proposed problem and the architecture of the developed models will be discussed.

A. Problem description

The purpose of the proposed Deep Learning system is to predict the crossing intention of pedestrians by using temporal information provided by image sequences and other categorical variables.

The input sequences are defined as a set of features $\mathbf{X}_t = \{X_{t-N}, X_{t-N+1}, \dots, X_t\}$, where N is the number of past frames and t the current frame. The output is defined as a binary label Y_{t+M} where $t+M$ is the index of the frame to be predicted. Thus, each pedestrian track with length P is divided in $S = P - N - M$ subsequences, with $t \in \mathbb{N} : t \in [N, P - M - 1]$. The remaining section will discuss the architecture followed by our model, explaining the role of each module separately.

B. General model architecture

The proposed system is composed by two main modules: a **feature extractor**, used to get useful information from image data and a **many-to-one RNN module**. At a high level, features extracted are introduced to the RNN module. Output of the RNN module is introduced in a fully-connected layer, and its output is passed through a sigmoid in order to get the predicted probability of crossing action in the trained time horizon. This architecture is represented in Fig. 1.

C. Feature extraction

Input features are extracted from color video sequences using three alternative techniques:

- Pretrained CNN models from ResNet family [13] and from ResNeXt family [14]. All models are pretrained on ImageNet [15]). The network was modified by cutting

off the last fully connected layer and obtaining the features from the average pool layer output.

- Convolutional autoencoder with previous pre-trained ResNet34 used as encoder [16]. An autoencoder is a type of encoder-decoder variant which is trained for the task of input reconstruction in a self-supervised manner. After the training process, the encoder is separated from the decoder and used as a feature extraction method. A high-level diagram of this architecture is shown in Fig. 2. The network was trained with a learning rate of 10^{-3} and using Binary Cross Entropy (BCE) loss.
- SegNet-based autoencoder [17]. This method is pre-trained on Cars Dataset [18] and obtained from [19].

No fine-tuning has been applied to any of the feature extractors during the RNN models training.

Following the same pooling strategy as in the pretrained ResNet34, output features of both encoders extracted from trained autoencoders, with size $512 \times 7 \times 7$, are averaged with a pooling layer with a 7×7 kernel, obtaining a $512 \times 1 \times 1$ tensor. The obtained tensor is flattened in order to obtain a one-dimensional vector of size 512.

In some experiments, categorical variables are used as inputs along with images. These variables are embedded in order to learn their multidimensional relationship between their categories. These embeddings are learned during training, and their dimension for each category is established following the heuristic proposed in the course imparted by Jeremy Howard [20]: $\min(\text{Int}(N_c/2 + 1), 50)$ where N_c is the number of categories of the variable (cardinality).

D. RNN module

For the recurrent module of the system, two variants of RNNs are used: Long Short-Term Memory (LSTM) [21] and Gated Recurrent Unit (GRU) [22]. These variants help in the fight of RNNs vanishing gradient problem. The main difference between GRUs and LSTMs is that GRUs are computationally more efficient and according to [23], they achieve similar results in sequence modeling problems.

Bidirectional variants of LSTMs and GRUs are also used on the experiments in order to test if the additional information of the reversed sequence can improve the understanding of the problem.

III. EXPERIMENTAL SETUP

In the following section, all experiments carried out will be detailed separately. Unless otherwise noted, LSTM module with the pre-trained ResNet50 used for feature extraction is the selected choice for the tests.

A. Image Data Preparation

All models have been trained on the JAAD [24], a naturalistic dataset focused on the behavior of pedestrians during their road-crossing action. It comprises 346 videos filmed inside a vehicle of duration ranging from 5 to 10 seconds. Their format varies both in frame rate and in resolution. There are 8 videos at 60 fps and 10 videos in HD resolution



Fig. 3: Two examples of JAAD pedestrian sequences: a crossing (top) and not crossing (down) situation.

(1280×720). The rest of the videos are filmed at 30 fps in FHD resolution (1920×1080)

Default split sets for training and testing suggested by the authors have been used in order to encourage possible future comparisons with other algorithms. In this split, HD videos are excluded, in addition to another set that presents low visibility (night scenes, heavy rain), totaling 323 videos. The videos at 60 fps included in these splits have been lowered to 30 fps.

The input to the model is composed of image sequences and, in some variants, categorical variables. Image sequences are extracted using the ground truth 2D bounding box annotations of pedestrians with crossing behavior. The height and width of them are equalized in order to avoid image deformation. All sequences are filtered by occlusion level and the bounding box height. Fully occluded samples and bounding boxes with height lower than 50 pixels have been removed only in the training set, leaving all the other sets unchanged, in order to see the behavior of the model in challenging situations.

Finally, in order to meet the input restrictions of feature extraction methods, images are resized to 224×224 (size used in training) and standardized using the per-channel mean and deviation of ImageNet.

B. Feature extraction method importance

Various tests were performed changing the feature selection method, one for each option on the list in subsection II-C. The reason for using autoencoders is to test if features extracted with a method specialized on the reconstruction of images help the network in its training process more than a classification pre-trained network.

C. Rescaling image features and normalization

Output data of the average pooling layer have a range between 0 and a maximum value which depends on the input

image and also on the feature extractor. We tried a rescaling approach to test if there is any improvement in the results. Rescaling is performed by dividing the sequence of image features between the maximum value in the batch, obtaining data between 0 and 1.

D. Influence of additional variables

Three categorical variables related to pedestrians and extracted from ground truth annotations have been used to study their influence on predictions: looking/gaze direction, orientation and state of movement. Looking direction is a binary variable, whose value is 1 if the pedestrian looks at the vehicle and 0 otherwise. The orientation variable has the following categories and are defined relative to the car: *front* (0), *back* (1), *left* (2) and *right* (3). State of movement has two possible values: *standing* and *moving*. Another variable used is the bounding box center (u_c, v_c), extracted from groundtruth annotations and divided by the maximum of each dimension in order to achieve independence from the camera sensor used.

The output of each embedding layer and the center of the bounding box are concatenated to the feature vector. As a result, the input vector used in the RNN module increases its size from 3 to 9.

E. LSTM versus GRU

As mentioned in subsection II-D, we perform a study on the influence of the type of RNN chosen, and their bidirectional variants. With this objective in mind, four RNN models are compared with the same hyperparameter configuration: LSTM, GRU, Bidirectional Long Short Term Memory (BDLSTM) and Bidirectional Gated Recurrent Unit (BDGRU).

F. Hyperparameter search

After an ablation study using grid search, the configuration used for the model is the following:

- RNN hidden dimension: 4
- Number of stacked RNN layers: 1
- Dropout (applied to RNN output): 0.5

The simplicity of the network is due to the trend towards overfitting of more complex networks.

G. Training configuration

PyTorch [25] has been the framework chosen to carry out the experiments. All experiments have been trained and tested on a single NVIDIA GTX TITAN X GPU. We have used Adam [26] as an optimizer with a learning rate of 10^{-4} . The loss function used for training is the BCE loss. To make computations deterministic, a fixed random seed has been established in all pseudorandom number generators. Finally, to avoid unnecessary processing, validation patience with a value of five has been set, i.e., if validation losses stop improving during five epochs, the training will end.

IV. RESULTS

The metrics used to compare these results are *accuracy*, *precision*, *recall* and, finally Average Precision (AP) score, calculated as a weighted sum, following equation 1, where R is recall, P is precision and n refers to the threshold number. All metric values are percentages.

$$AP = \sum_n (R_n - R_{n-1})P_n \quad (1)$$

A. Feature extraction method Importance

TABLE I: Different feature extraction methods results

Method	Acc.	P	R	AP
ResNet18	62.68	62.87	98.82	69.08
ResNet34	63.32	65.71	86.75	74.33
ResNet50	65.75	69.63	80.43	75.62
ResNet101	68.95	71.29	84.49	77.16
ResNet152	62.53	64.26	90.59	75.85
ResNeXt50	70.04	74.96	78.39	79.87
ResNeXt101	69.45	74.14	78.73	81.20
ConvAE-ResNet	62.67	62.67	100.00	61.64
ConvAE-SegNet	62.67	62.67	100.00	68.46

Results per method are given in table I, pre-trained models from ResNet and ResNeXt families, obtain better results than self-trained ones. The increase in the complexity of the network is directly related to the increase in all performance metrics. One possible reason is the difference in training data size and diversity between ImageNet [15] and JAAD [24] or CARS dataset [18].

Although the images are reconstructed quite accurately in the self-trained extractors, the output features of the encoder lack useful information for the RNN module. This is shown in the recall value of 100%, which means that the model has converged in predicting that every pedestrian will cross. This problem may be caused by the use of an average pooling layer after training since, in pre-trained models, average pooling is used during the training stage.

B. Rescaling image features and normalization

Rescaling input image features contributes to an improvement in the results (see table II). These results show that the high variation in input features penalized the learning process.

TABLE II: Rescaling image features results

Normalization type	Acc.	P	R	AP
None	65.75	69.63	80.43	75.62
Rescaling	65.89	70.75	77.70	76.84

C. Influence of additional variables

The incorporation of all additional variables improves the AP from 75.62% to a 80.00% (see table III). This result shows that the incorporation of meaningful data can act as a regulatory factor to allow greater learning generalization. Individually, orientation and looking direction are the variables with more weight followed by the state of movement. Those variables are also used by drivers when they infer the pedestrians' crossing intentions (e.g. a pedestrian walking towards the road and a pedestrian at the curb looking at the driver's car are more likely to cross than a pedestrian walking parallel to the car and suddenly stopping). In the case of the bounding box center in the image, it has less weight. This is probably due to its relativeness and high variation as it belongs to the image coordinate system.

TABLE III: Influence of additional variables results

Variable	Acc.	P	R	AP
None	65.75	69.63	80.43	75.62
Looking	65.13	67.28	86.37	76.94
Orientation	67.12	69.96	83.30	77.71
Bbox center	64.78	66.59	87.91	76.15
Movement	68.76	72.71	80.30	76.75
All	68.82	74.20	77.03	80.00

D. LSTM versus GRU

TABLE IV: RNN selection results

Method	Acc.	P	R	AP
LSTM	65.75	69.53	80.43	75.62
GRU	62.65	62.84	98.88	64.25
BDLSTM	67.33	69.20	86.28	79.07
BDGRU	67.62	76.00	70.66	80.19

According to table IV, in this problem, additional temporal information provided by bidirectionality can improve the results of an LSTM-based network. GRU obtains worse results than LSTM and in the case of the bidirectional variants, both RNNs improve the results, with BDGRU performing slightly better than the BDLSTM. This can be due to the high dropout used and the fixed seed used for reproducibility.

TABLE V: Best model for each improvement configuration

all add. var.	reescalng	best RNN	best feat. extr.	AP
-	-	-	-	75.62
✓	-	-	-	80.00
-	✓	-	-	76.84
-	-	✓	-	80.19
-	-	-	✓	81.20
✓	✓	✓	✓	83.34

E. Final results

To see the effect of the above experiments together, a model has been trained including all of the previous upgrades. According to AP scores in table V, improvements work well together with an increase of more than 8% concerning the simpler model.

F. Qualitative results

In figure 4, two example sequences are shown with the input image sequence at the left and the output crossing probability at the right. The model used in this experiment is the best model from table V with a change in the output dimension. Instead of outputting the crossing probability one second in the future, the output is split into eight equispaced time steps between 0 and 1 second. Both sequences belong to the same pedestrian. In the top one, the pedestrian is not going to cross in one second in the future, and in the bottom one, the pedestrian is beginning to cross. As the graphs show, the probability of crossing is low in the first time step of the top graph, but this value is doubled at the end of the prediction, indicating a possible future crossing, which becomes more likely in the bottom case.

G. Dataset limitations

JAAD dataset is one of the few datasets focused on pedestrian behavior. However, it is composed of short videos. Besides there are challenging situations that affect training: windshield wipers occlusion, bad weather conditions (raining, snowing) and reflections on the windshield. Additionally, small pedestrians are a problem that can be filtered easily, but this is not the case for non-relevant pedestrians i.e., pedestrians who are crossing or not but are not in the path of the vehicle. Filtering out these problems can lead to better training convergence, but at the same time, it leads to a loss of training data. In Fig. 5 some examples of those challenging situations discussed before are shown.

V. CONCLUSIONS & FUTURE WORK

A method based on a CNN feature extractor and a RNN many-to-one module has been proposed to predict pedestrian crossing action in the future. Image and categorical data have been the chosen sources of information for the model. Experiments carried out have shown that pre-trained networks can provide better temporal information than autoencoders. The inclusion of additional data can improve the results, as well as the use of bidirectional LSTM. Applying all improvements at the same time rises AP score more than 8%. These

results are encouraging, and they show a way ahead in the development of more reliable and secure intention prediction systems.

As stated in the discussion, the JAAD dataset is useful for tasks such as detection and tracking, but not for video understanding. For this reason, the PIE dataset [27] will be considered in future work in order to develop models. New context and local variables could also be studied, such as 3D pose, kinematics, relative distances and presence of traffic lights or zebra crossings. Concerning hyperparameter optimization, non-exhaustive search methods (e.g., Bayesian Optimization methods) could be applied to the training process. Finally, different strategies can be followed by the feature extractor output instead of averaging all output channels in a single vector (e.g., attention mechanisms or the use of convolutional LSTMs [28]).

VI. ACKNOWLEDGEMENTS

This work was funded by Research Grants S2018/EMT-4362 (Community Reg. Madrid), DPI2017-90035-R (Spanish Min. of Science and Innovation), BRAVE Project, H2020, Contract #723021 and by Universidad de Alcal, via a pre-doctoral grant to the first author (FPI-UAH). It has also received funding from the Electronic Component Systems for European Leadership Joint Undertaking under grant agreement No 737469 (AutoDrive Project). This Joint Undertaking receives support from the European Unions Horizon 2020 research and innovation programme and Germany, Austria, Spain, Italy, Latvia, Belgium, Netherlands, Sweden, Finland, Lithuania, Czech Republic, Romania, Norway.

REFERENCES

- [1] “WHO | Global status report on road safety 2018.”
- [2] European Commission, “Road Safety In The European Union,” vol. 29, no. 3, pp. 359–367, 2018.
- [3] Z. Zou, Z. Shi, Y. Guo, and J. Ye, “Object detection in 20 years: A survey,” *CoRR*, vol. abs/1905.05055, 2019.
- [4] C. G. Keller and D. M. Gavrila, “Will the pedestrian cross? a study on pedestrian path prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 2, pp. 494–506, April 2014.
- [5] R. Quintero Mínguez, I. Parra Alonso, D. Fernández-Llorca, and M. A. Sotelo, “Pedestrian path, pose, and intention prediction through gaussian process dynamical models and pedestrian activity recognition,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 5, pp. 1803–1814, May 2019.
- [6] E. Rehder, F. Wirth, M. Lauer, and C. Stiller, “Pedestrian prediction by planning using deep neural networks,” *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1–5, 2017.
- [7] O. Styles, A. Ross, and V. Sanchez, “Forecasting pedestrian trajectory with machine-annotated training data,” in *2019 IEEE Intelligent Vehicles Symposium (IV)*, June 2019, pp. 716–721.
- [8] A. Rudenko, L. Palmieri, M. Herman, K. M. Kitani, D. M. Gavrila, and K. O. Arras, “Human motion trajectory prediction: A survey,” *CoRR*, vol. abs/1905.06113, 2019.
- [9] J. F. P. Kooij, F. Flohr, E. A. I. Pool, and D. M. Gavrila, “Context-Based Path Prediction for Targets with Switching Dynamics,” *International Journal of Computer Vision*, vol. 127, no. 3, pp. 239–262, Mar. 2019.
- [10] A. Rasouli, I. Kotseruba, and J. K. Tsotsos, “Are they going to cross? a benchmark dataset and baseline for pedestrian crosswalk behavior,” in *The IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.
- [11] K. Hara, H. Kataoka, and Y. Satoh, “Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet?” *CoRR*, vol. abs/1711.09577, 2017.



Fig. 4: Two examples in test set. The top one represents a non-crossing sequence and at the bottom, a crossing one. Left graphics show the output crossing probability at eight future time steps between 0 and 1 seconds (0 and 30 frames).



Fig. 5: Some defiant cases from JAAD dataset

[12] K. Saleh, M. Hossny, and S. Nahavandi, "Real-time intent prediction of pedestrians for autonomous ground vehicles via spatio-temporal densenet," in *2019 International Conference on Robotics and Automation (ICRA)*, May 2019, pp. 9704–9710.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[14] S. Xie, R. B. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," *CoRR*, vol. abs/1611.05431, 2016.

[15] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A Large-Scale Hierarchical Image Database," in *CVPR09*, 2009.

[16] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[17] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *CoRR*, vol. abs/1511.00561, 2015.

[18] J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3d object representations for fine-grained categorization," in *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[19] F. Liu, "Autoencoder," <https://github.com/foamliu/Autoencoder>, 2020.

[20] "fastai course v3," <https://github.com/fastai/course-v3>, 2020.

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[22] K. Cho, B. van Merriënboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *CoRR*, vol. abs/1406.1078, 2014.

[23] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *CoRR*, vol. abs/1412.3555, 2014.

[24] I. Kotseruba, A. Rasouli, and J. K. Tsotsos, "Joint attention in autonomous driving (JAAD)," *CoRR*, vol. abs/1609.04741, 2016.

[25] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 2019, pp. 8024–8035.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.

[27] A. Rasouli, I. Kotseruba, T. Kunic, and J. K. Tsotsos, "Pie: A large-scale dataset and models for pedestrian intention estimation and trajectory prediction," in *International Conference on Computer Vision (ICCV)*, 2019.

[28] S. Xingjian, 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 *Advances in neural information processing systems*, 2015, pp. 802–810.