MOVING TARGET CLASSIFICATION IN AUTOMOTIVE RADAR SYSTEMS USING CONVOLUTIONAL RECURRENT NEURAL NETWORKS

Sangtae Kim[*], *Seunghwan Lee*[*], *Seungho Doo*[†], *and Byonghyo Shim*[*]

[*]Department of Electrical and Computer Engineering, INMC
Seoul National University
[†] Hyundai Mobis Co. DAS Sensor Engineering Team
Email : *{stkim, shlee, bshim}@islab.snu.ac.kr, † seungho.doo@mobis.co.kr

ABSTRACT

Moving target classification is a key ingredient to avoid accident in autonomous driving systems. Recently, fast chirp frequency modulated continuous wave (FMCW) radar has been popularly used to recognize moving targets due to its ability to discriminate moving objects and stationary clutter. In order to protect vulnerable road users such as pedestrians and cyclists, it is essential to identify road users in a very short period of time. In this paper, we propose a deep neural network that consists of convolutional recurrent units for target classification in automotive radar system. In our experiment, using the real data measured by the fast chirp FMCW-based high range resolution radar, we show that the proposed network is capable of learning the dynamics in time-series image data and outperforms the conventional classification schemes.

Index Terms— convolutional neural networks, recurrent neural networks, classification, fast chirp FMCW radar

1. INTRODUCTION

Moving target classification is a key ingredient to avoid accident in autonomous driving systems. In recent years, there is a growing interest to implement target classifier using 24GHz and 77GHz radars [1]. In classifying the human activities, the radar cross section (RCS) has been popularly used. The RCS of target, which corresponds to the effective area of target, depends on various factors such as posture, body shape, clothing type, size, orientation, to name just a few. However, the RCS-based approach is computationally inefficient and also is not easy to model the various behaviors of moving targets. As an alternative approach, fast chirp frequency modulated continuous wave (FMCW) radar has been popularly used in recent years. The fast chirp FMCW radar has a simple structure and also is very easy to extract range-velocity (RV) image used to discriminate moving objects and stationary clutter [2]. Since the protection of vulnerable road users such as pedestrians and cyclists is key to the successful commercialization of autonomous driving systems, extracting the radar features to identify the road users is an important problem. For example, high resolution radar characteristics of pedestrians and cyclists have been presented in [3].

One major bottleneck in the target classification of the automotive radar systems is very stringent processing time constraint. Since the acquisition of data snapshot and classification should be completed in a very short period of time (e.g., less than 0.5 sec), it is technically not possible to use the data accumulated for a long period. Clearly, if only a few snapshots are used in the target classification, one might not ontain the accurate classification results. Fortunately, recent studies demonstrate that the deep learning based classifiers achieve significant performance improvement over the conventional approaches such as linear classifier or support vector machine (SVM). In particular, convolutional neural network (CNN) plays a key role in the front-end implementation and has shown great promise in the field of image recognition tasks such as object classification and target detection when the underlying data has 2-dimensional structure [4, 5]. CNN has been widely used in computer vision tasks such as video representation and classification of human activity. Besides, recurrent neural networks (RNN) have been popularly used in time-series analysis. Well-known limitation of RNN models is the vanishing gradient problem [6]. As a means to overcome this problem, long short-term memory (LSTM) [7] and gated recurrent unit (GRU) [8] have been introduced. LSTM and GRU-based architectures are used in various tasks such as speech recognition and video captioning due to their ability to preserve information over long period of time.

An aim of this paper is to present a convolutional recurrent unit-based deep neural network for automotive radar systems. In the proposed scheme, time-series radar snapshots are transformed into RV images via 2-dimensional discrete Fourier transform (2-D DFT). In order to handle time-series RV images, we exploit the convolutional recurrent neural network (CRNN), in particular, convolutional LSTM (C-LSTM)

This work was partly supported by Hyundai Mobis Co. and the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIP) (2016R1A2B3015576)

[9] and convolution GRU (C-GRU) [10]. The key ingredient of the proposed classifier is a convolutional recurrent unit-based feature extractor to obtain the dynamics of input RV images used to perform the target classification.

In our experiments, we focus on the lateral movement of vulnerable road users, which reflects the most frequent peril scenario. Using the real data measured by the fast chirp FMCW-based high range resolution radar, we show that the proposed network can learn the dynamics of the time-series image data. Also, we show that the proposed network outperforms the conventional classification schemes.

2. FAST CHIRP FMCW RADAR-BASED MODEL

2.1. Fast Chirp FMCW Radar

The concept of the fast FMCW chirp modulation has been proposed in [11] and applied to the automotive radar system in [12]. The main idea behind the fast chirp FMCW radar is to scale down the chirp modulation time in the microsecond (μsec) range. As a result, the conventional FMCW chirp is replaced by a sequence of short chirps. Using this radar, 2-D data matrix $\mathbf{S} \in \mathbb{C}^{M \times N}$ can be measured. Note that N is the number of single chirps and $M = f_s T_{\text{chirp}}$ is the number of sample points in a single chirp (T_{chirp} is the single chirp modulation time and f_s is the sampling frequency). For the data matrix $\mathbf{S}$, the absolute value of 2-D discrete Fourier transform (DFT) is expressed as

$$\mathbf{X}[k,l] = \frac{1}{\sqrt{M \cdot N}} \left| \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} \left[\mathbf{S}[m,n] e^{-2j\pi \frac{mk}{M}} \right] e^{-2j\pi \frac{nl}{N}} \right| .$$

Note that $\mathbf{X}$ is referred to as the RV image and used as an input data of the classifier.

2.2. Recurrent Neural Networks

In this subsection, we briefly review the conventional RNN and LSTM. RNN-based model can be applied to a variable length of sequence data. Specifically, when the sequence $\mathcal{X} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T)$ is given, then the activation $\mathbf{h}_t$ of a vanilla RNN is given by

$$\mathbf{h}_t = \tanh(\mathbf{W}_{xh}\mathbf{x}_t + \mathbf{W}_{hh}\mathbf{h}_{t-1})$$

where $\mathbf{W}_{xh}$ and $\mathbf{W}_{hh}$ are the weight matrices.

Due to the vanishing gradient problem, it is difficult to train the network to learn the long-term dynamics. As an alternative to the RNN, LSTM and GRU have been popularly used [7, 8]. The LSTM structure can be extended using peephole connections [13]. The peephole LSTM used in our work has been shown to be a promising approach for the applications where the precise duration of intervals between relevant events matters. In the sequel, we call the peephole LSTM as LSTM for brevity. In essence, LSTM incorporates memory units that allow the network to learn when to forget previous hidden states and when to update hidden states given new information. The activation $\mathbf{h}_t$ of LSTM can be expressed as the following set of equations:

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{W}_{ci} \odot \mathbf{c}_{t-1}) \\
\mathbf{f}_t &= \sigma(\mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{W}_{cf} \odot \mathbf{c}_{t-1}) \\
\mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}_{xc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1}) \\
\mathbf{o}_t &= \sigma(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{W}_{co} \odot \mathbf{c}_{t-1}) \\
\mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t)
\end{aligned}$$

where $\sigma(\cdot)$ is the sigmoid function $\left(\sigma(x) = \frac{e^x}{1+e^x}\right)$, $\tanh(\cdot)$ is the hyperbolic-tangent function, and $\odot$ is the element-wise product. LSTM includes the gates (the input gate $\mathbf{i}_i$, the forget gate $\mathbf{f}_t$, and the output gate $\mathbf{o}_t$) which decide the amount of information to pass through. Note that $\mathbf{i}_i$, $\mathbf{f}_t$, and $\mathbf{o}_t$, have the same equations but with different parameters. The cell state $\mathbf{c}_t$ which corresponds to the internal memory of the unit represents the output indirectly. The activation $\mathbf{h}_t$ is the nonlinear version of $\mathbf{c}_t$ (*tanh* is popularly used as a nonlinear function).

The overall idea behind GRU is quite similar to that of LSTM. It has been shown that GRU networks perform similar to LSTM but have simpler structure [14]. The hidden state $\mathbf{h}_t$ of the GRU is defined by the following set of equations:

$$\begin{aligned}
\mathbf{z}_t &= \sigma(\mathbf{W}_z\mathbf{x}_t + \mathbf{U}_z\mathbf{h}_{t-1} + \mathbf{W}_{cz} \odot \tilde{\mathbf{h}}_{t-1}) \\
\mathbf{r}_t &= \sigma(\mathbf{W}_r\mathbf{x}_t + \mathbf{U}_r\mathbf{h}_{t-1} + \mathbf{W}_{cr} \odot \tilde{\mathbf{h}}_{t-1}) \\
\tilde{\mathbf{h}}_t &= \tanh(\mathbf{W}\mathbf{x}_t + \mathbf{r}_t \odot (\mathbf{U}\mathbf{h}_{t-1})) \\
\mathbf{h}_t &= \mathbf{z}_t \odot \mathbf{h}_{t-1} + (\mathbf{1} - \mathbf{z}_t) \odot \tilde{\mathbf{h}}_t
\end{aligned}$$

Although there is neither internal memory nor output gate in GRU, the hidden state $\mathbf{h}_t$ represents both the internal memory and the output. The reset gate $\mathbf{r}_t$ and the update gate $\mathbf{z}_t$ of GRU operate in a similar way to the input and forget gates of LSTM. We also applied peephole connection to GRU.

3. DEEP NEURAL NETWORK WITH CONVOLUTIONAL RECURRENT UNITS

The overall moving target classification system is illustrated in Fig. 1. The proposed network consists of one convolutional recurrent layer, one convolutional layer, and one fully connected layer. We use a recurrent unit whose input is 2-D time-series signals obtained from the radar. As a convolutional recurrent unit for extracting features of moving targets, we apply C-LSTM [9] and C-GRU [10]. One main issue in the automotive radar systems is to understand the dynamics of targets in a reasonable time with only a few number of images. One main advantage of the CRNN-based feature extractor is that the network can learn the dynamics of moving targets. Also, by applying the convolution operators, the number of parameters can be reduced significantly. Clearly, this will be very helpful in memory-constrained applications such

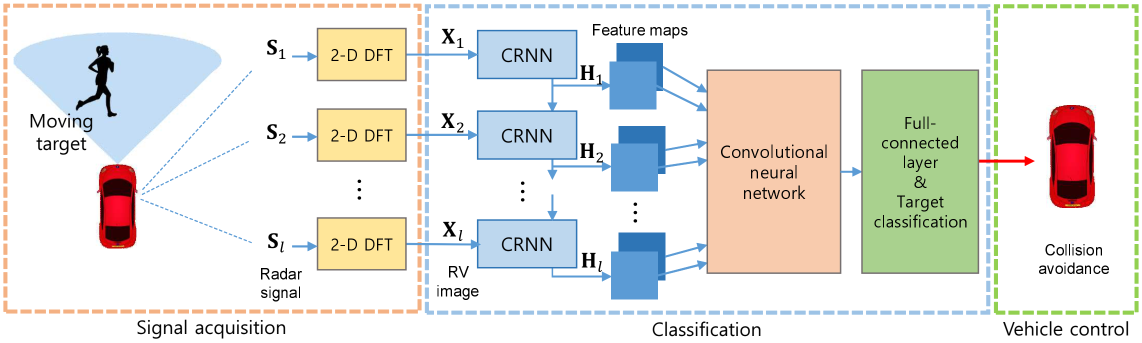

Fig. 1: Proposed moving target classification system; The time-series radar signals is measured and transformed into RV images by 2-D DFT. The CRNN-based proposed network is applied to extract the features of moving targets. The convolutional layer and fully connected layer are followed.

as the proposed system. Another important benefit is that the spatial information of an input image can be maintained and the higher level of spatial features can be learned through the deep layers.

By applying convolutional operators, the activations $\mathbf{H}_t$ of C-LSTM can be re-written as

$$\begin{aligned}
\mathbf{I}_t &= \sigma(\mathbf{W}_{xi} * \mathbf{X}_t + \mathbf{W}_{hi} * \mathbf{H}_{t-1} + \mathbf{W}_{ci} \odot \mathbf{C}_{t-1}) \\
\mathbf{F}_t &= \sigma(\mathbf{W}_{xf} * \mathbf{X}_t + \mathbf{W}_{hf} * \mathbf{H}_{t-1} + \mathbf{W}_{cf} \odot \mathbf{C}_{t-1}) \\
\mathbf{C}_t &= \mathbf{F}_t \odot \mathbf{C}_{t-1} + \mathbf{I}_t \odot \phi(\mathbf{W}_{xc} * \mathbf{X}_t + \mathbf{W}_{hc} * \mathbf{H}_{t-1}) \\
\mathbf{O}_t &= \sigma(\mathbf{W}_{xo} * \mathbf{X}_t + \mathbf{W}_{ho} * \mathbf{H}_{t-1} + \mathbf{W}_{co} \odot \mathbf{C}_{t-1}) \\
\mathbf{H}_t &= \mathbf{O}_t \odot \phi(\mathbf{C}_t)
\end{aligned}$$

where $\phi(\cdot)$ is the nonlinear activation function and $*$ is the convolution operator. Also, the activations $\mathbf{H}_t$ of C-GRU can be written as

$$\begin{aligned}
\mathbf{Z}_t &= \sigma(\mathbf{W}_z * \mathbf{X}_t + \mathbf{U}_z * \mathbf{H}_{t-1} + \mathbf{W}_{cz} \odot \tilde{\mathbf{H}}_{t-1}) \\
\mathbf{R}_t &= \sigma(\mathbf{W}_r * \mathbf{X}_t + \mathbf{U}_r * \mathbf{H}_{t-1} + \mathbf{W}_{cr} \odot \tilde{\mathbf{H}}_{t-1}) \\
\tilde{\mathbf{H}}_t &= \phi(\mathbf{W} * \mathbf{X}_t + \mathbf{R}_t \odot (\mathbf{U} * \mathbf{H}_{t-1})) \\
\mathbf{H}_t &= \mathbf{Z}_t \odot \mathbf{H}_{t-1} + (\mathbf{1} - \mathbf{Z}_t) \odot \tilde{\mathbf{H}}_t
\end{aligned}$$

As a nonlinear activation function (denoted as ϕ), *tanh* is widely used since it can suppress the activation. However, this option might slow down the training process and also cause vanishing gradient problem. The activation function *relu* might boost the training process but still causes a value exploding problem. In order to handle this problem, we employed the regularization on the weights, which also prevents the overfitting of the model. For the purpose of comparison, we provide the experiment results for CRNN units with *tanh* and *relu* and show that the *relu*-based models outperforms *tanh*-based models.

Center Frequency	Bandwidth	Sampling period	Range Resolution	Velocity resolution
77GHz	500MHz	50ms	0.3m	0.11m/s

Table 1: The characteristics of the high range resolution radar used in the experiment

4. EXPERIMENTS

In our experiments, we evaluate the classification performance of the network using the data obtained by the fast chirp FMCW radar of Hyundai Mobis. The detailed specifications of the radar are listed in the Table 1.

4.1. Datasets and Experimental Settings

In our experiments, the range-velocity images are obtained from 2-D DFT of radar signal matrices. The datasets are categorized into four types: pedestrians, cyclists, vehicles, and none for non-existence of any target. As illustrated in Fig. 2, one of three targets moves through the detection range of radar at a fixed speed. In the experimental environments, pedestrians, cyclists, and vehicles moved at nearly $(5km/h, 12km/h)$, $(12km/h, 20km/h)$, and $(20km/h, 30km/h)$, respectively. Note that the targets' speeds and the distances to radar are not available and hence unused for the target classification. When the target is moving, radar signal reflected from the moving target is discretely sampled in every 50ms. The radar signals are transformed into a range-velocity images of size 100×256 by taking 2-D DFT patterns. The dataset consisting of 1,505 snapshots is divided into 88% and 12% for training and test, respectively.

In order to train the recurrent model, at most l consecutive RV images are concatenated as a sequence. For realistic scenario, we set l to 3 and 5. Then, it takes 0.1 and 0.2 sec to obtain the signals, which are enough to capture the move-

target	C-LSTM, relu, $l = 5$				C-LSTM, relu, $l = 3$				C-LSTM, tanh, $l = 5$				C-LSTM, tanh, $l = 3$				LSTM, $l = 5$			
class	Ped	Cyc	Veh	No	Ped	Cyc	Veh	No	Ped	Cyc	Veh	No	Ped	Cyc	Veh	No	Ped	Cyc	Veh	No
Ped	90.8	6.8	0	2.4	95.8	1.9	0	2.3	94.8	1.6	0	3.6	98.1	1.9	0	0	64.3	14.9	0	20.9
Cyc	34.5	65.5	0	0	29.2	70.8	0	0	42.9	57.1	0	0	84.7	15.3	0	0	66.1	28.5	0	5.4
Veh	0	0	100	0	0	0	98.6	1.4	0	0	100	0	0	0	100	0	0	0	98.2	1.8
target	C-GRU, relu, $l = 5$				C-GRU, relu, $l = 3$				C-GRU, tanh, $l = 5$				C-GRU, tanh, $l = 3$				CNN, $l = 1$			
class	Ped	Cyc	Veh	No	Ped	Cyc	Veh	No	Ped	Cyc	Veh	No	Ped	Cyc	Veh	No	Ped	Cyc	Veh	No
Ped	95.6	0	0	4.4	98.6	0.5	0	0.9	78.6	9.5	0	7.1	95.4	4.2	0	0.5	91.7	8.3	0	0
Cyc	9.5	90.5	0	0	34.7	65.3	0	0	45.3	47.6	0	7.1	73.6	26.4	0	0	65	35	0	0
Veh	1.2	0	98	0.8	0	0	100	0	0	0	100	0	0	0	92.9	7.1	0	0	100	0

Table 2: The test results for proposed networks, LSTM, and CNN. Each number represents the proportion of the predicted class in percentage. The classes in the first column denotes the ground truth. The activation function ϕ is replaced to either *relu* or *tanh*. The sequence length l is either 3 or 5 for recurrent units and 1 for CNN model.

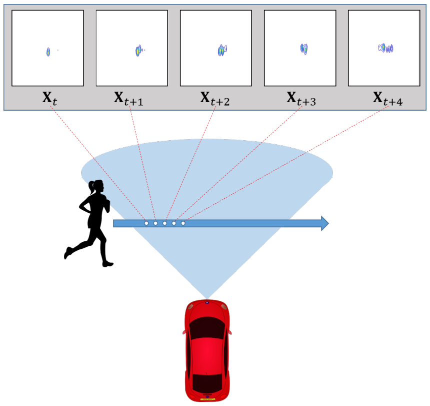

Fig. 2: A target moving straight at a fixed speed in a lateral direction to the stationary radar. The radar signals are obtained every 50 ms.

ments of targets. At the beginning and the end of radar signals of moving target, none-class images are added. For instance, if 5 consecutive images of a target are given (e.g., $\mathcal{X} = \{\mathbf{X}_1, \cdots, \mathbf{X}_5\}$) and $l = 3$, we can obtain 7 sequences (e.g., $\mathcal{S}_1 = \{\mathbf{N}, \mathbf{N}, \mathbf{X}_1\}, \mathcal{S}_2 = \{\mathbf{N}, \mathbf{X}_1, \mathbf{X}_2\}, \cdots, \mathcal{S}_7 = \{\mathbf{X}_5, \mathbf{N}, \mathbf{N}\}$ where $\mathbf{N}$ is the none-class image). As we assume the single target scenario, every sequence includes the RV images of a single target. In other words, images of pedestrian and none-class together can be included in a sequence (none-class is not a target) but images of the combinations of moving targets such as pedestrian and cyclist cannot. Therefore, every sequence can be mapped into one of four classes.

4.2. Deep Neural Network Architecture

In this part, we describe the proposed neural network architecture. In the front-end, each convolutional recurrent layer outputs 5 feature maps. The 5 feature maps from l CRNN units are concatenated, and hence, the number of channels of input to the convolutional layer becomes $5l$. For convolutional layer, we use 50 filters of size 2×2. Finally, the output is fully connected to 4 nodes and each node is mapped to each of class (pedestrian, cyclist, vehicle, and none.) Softmax function is applied to the output vector of dense layer. We use the cross-entropy as a loss function with one-hot label vector. We use AdamOptimizer and its learning rate and momentum are set to 10^{-3} and 0.9, respectively. In order to avoid the overfitting of the network, we add ℓ_2-norm regularization on weight matrices with multiplier set to 0.01.

In the convolutional recurrent units, we design the activation $\mathbf{H}_t$ to be smaller size than $\mathbf{X}_t$ to reduce the computational cost. We used the 4×4 filters for the input $\mathbf{X}_t$ with 5 channels and stride-2. For the previous activation $\mathbf{H}_{t-1}$, we used 3×3 filters with 5 channels and stride-1. The filters use in peephole connections are of the same size as the activations.

4.3. Comparison Results

We perform the simulation using the proposed neural network, conventional LSTM without convolution, and CNN without recurrent units. For the conventional LSTM network, the RV images are reshaped into vectors but the same sequences are used as in the convolutional recurrent units. We set the input and output size to each LSTM unit to be 25600 and 10, respectively. For CNN model, every RV image is labelled individually. In CNN model, we use 2 convolutional layers (4×4 and 3×3 filters with 50 and 200 channels, respectively) and 1 fully connected layer. The simulation results are shown in Table 2. For each class, we compute the proportion of predicted class given the target class. We present the various results by changing CRNN units, activation functions, and sequence lengths. As an activation function ϕ, either *relu* or *tanh* is used. Other activation functions can also be applied to CRNN units. In this work, however, we focus on the essential effect of changing the activation function from *tanh* to *relu*. The obtained results are the average of experiments for three times. We observe that the performances of C-LSTM and C-GRU models are more or less similar. We also observe that the *relu*-based model outperforms the *tanh*-based model. Particularly, *tanh*-based model learns the dynamics of moving targets slowly and often fails to learn the dynamics.

	C-LSTM	C-GRU	LSTM	CNN
# of parameters	135K	103K	218K	260K
# of operations	11.3M	10.5M	1M	598M

Table 3: The number of parameters (without biases) and operations (multiplications only) used in the models

Among all targets, discrimination between pedestrians and cyclist is particularly difficult using conventional approaches but we can see that CRNN-based models classifies the two targets more clearly. For the purpose of the comparison of the complexity, we provide the number of parameters and operations of the all models in Table 3. Since we use the same structure for C-LSTM and C-GRU models, the difference of complexity only results from the units themselves. The networks using CRNN achieve improved classification performances with reasonable amount of memory requirements to LSTM and CNN models as well.

5. CONCLUSION

In this paper, we have presented a convolutional recurrent units-based deep neural network for moving target classification. We applied the convolutional LSTM and GRU units as feature extractors. The proposed networks have shown the ability to learn the dynamics of moving targets. Also, we have shown from the simulation results that the proposed networks outperform the conventional networks which lacks either apply convolutional layers or recurrent layers. For the commercialization of autonomous driving systems, we believe that the proposed work will be a useful for further works.

6. REFERENCES

[1] I. Matsunami, R. Nakamura, and A. Kajiwara, "Rcs measurements for vehicles and pedestrian at 26 and 79ghz," in *Signal Processing and Communication Systems (ICSPCS), 2012 6th International Conference on*. IEEE, 2012, pp. 1–4.

[2] I. Bilik, J. Tabrikian, and A. Cohen, "Gmm-based target classification for ground surveillance doppler radar," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 267–278, 2006.

[3] E. Schubert, F. Meinl, M. Kunert, and W. Menzel, "High resolution automotive radar measurements of vulnerable road users–pedestrians & cyclists," in *Microwaves for Intelligent Mobility (ICMIM), 2015 IEEE MTT-S International Conference on*. IEEE, 2015, pp. 1–4.

[4] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Action classification in soccer videos with long short-term memory recurrent neural networks," no. 9. Springer, 2010, pp. 154–159.

[5] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., no. 10. Curran Associates, Inc., 2014, pp. 568–576. [Online]. Available: http://papers.nips.cc/paper/5353-two-stream-convolutional-networks-for-action-recognition-in-videos.pdf

[6] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

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

[8] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[9] 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.

[10] N. Ballas, L. Yao, C. Pal, and A. Courville, "Delving deeper into convolutional networks for learning video representations," *arXiv preprint arXiv:1511.06432*, 2015.

[11] A. G. Stove, "Linear fmcw radar techniques," in *IEE Proceedings F (Radar and Signal Processing)*, vol. 139, no. 5. IET, 1992, pp. 343–350.

[12] V. Winkler, "Range doppler detection for automotive fmcw radars," in *Microwave Conference, 2007. European*. IEEE, 2007, pp. 1445–1448.

[13] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with lstm recurrent networks," *Journal of machine learning research*, vol. 3, no. Aug, pp. 115–143, 2002.

[14] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.