

Novel Radar-based Gesture Recognition System using Optimized CNN-LSTM Deep Neural Network for Low-power Microcomputer Platform

Mateusz Chmurski^{1,2}^a and Mariusz Zubert²^b

¹*Infineon Technologies AG, Am Campeon 1-15, 85579 Neubiberg, Germany*

²*Department of Microelectronics and Computer Science, Lodz University of Technology, Wólczańska 221/223, 90-001 Łódź, Poland*

Keywords: Deep Learning, Gesture Recognition, Optimization, Pruning, CNN, LSTM.


Abstract: The goal of the embedded hand gesture recognition based on a radar sensor is to improve a human-machine interface, while taking into consideration privacy issues of camera sensors. In addition, the system has to be deployable on a low-power microcomputer for the applicability in broadly defined IoT and smart home solutions. Currently available gesture sensing solutions are ineffective in terms of low-power consumption what prevents them from the deployment on the low-power microcomputers. Recent advances exhibit a potential of deep learning models for a gesture classification whereas they are still limited to high-performance hardware. Embedded microcomputers are constrained in terms of memory, CPU clock speed and GPU performance. These limitations imply a topology design problem that is addressed in this work. Moreover, this research project proposes an alternative signal processing approach – using the continuous wavelet transform, which enables us to see the distribution of frequencies formed by every gesture. The newly proposed neural network topology performs equally well compared to the state of the art neural networks, however it needs only 54.6% of memory and it needs 20% of time to perform inference relative to the state of the art models. This dedicated neural network architecture allows for the deployment on resource constrained microcomputers, thus enabling a human-machine interface implementation on the embedded devices. Our system achieved an overall accuracy of 95.05% on earlier unseen data.


1 INTRODUCTION

Gesture sensing is one of the most intuitive and common approaches in the field of human-computer interaction (HCI). It can be applied in many fields of our life. From smart homes (Wan et al., 2014; Alemuda and Lin, 2017), smart cars, game consoles to diverse mobile devices such as wearables e.g. smart watches or mobile phones (Alemuda and Lin, 2017). Previous work on that topic has shown many inadequacies of such kind of systems (Ahmed and Cho, 2020). Conventional gesture sensing systems mainly utilize optical sensors, for instance, depth sensors (Ma and Peng, 2018; Tran et al., 2020; Lai and Yanushkevich, 2018), RGB cameras (Wang and Payandeh, 2017; Li et al., 2018) or a combination of both (Van den Bergh and Van Gool, 2011; Chai et al.,

2016; Yunan Li et al., 2016). However, they offer an unsatisfactory robustness capability, which heavily depends on environmental conditions such as fog, background clutter, illumination conditions or operating environment (Ahmed and Cho, 2020; Shanthakumar et al., 2020). As opposed to video-based gesture sensing solutions, radar sensors are not affected by an operating environment and various illumination conditions (Hazra and Santra, 2018) as they can perform well in highly lit and dark environments. Privacy concern is another drawback of an optical-based gesture sensing (Schiff et al., 2009), particularly in an age of continuously increasing need of privacy and personal data protection.

Another significant problem of gesture recognition task is a generalizability of the system

^a <https://orcid.org/0000-0002-5442-4744>

^b <https://orcid.org/0000-0001-7924-7724>

over multiple users and multiple operating environments, what makes a gesture recognition a very sophisticated task (Yasen and Jusoh, 2019). Additionally, in case of such systems, a power consumption and a low-memory-footprint are a subject to high overheads, resulting in an algorithmic complexity, an unacceptable inference time, a decreased system portability, thereby a lack of deployment possibility on a low-power microcomputer.

Radar sensors provide also a touchless and a high-resolution environment for capturing gestures allowing users for an easy interaction with the system and a classification of very fine-grained gestures, what manifests itself in an attraction of considerable interest in an academic and an industrial circles (Patole et al., 2017).

In most of cases, the task of gesture classification with radar is solved by an analysis of the range-doppler maps representing a dependency between a velocity and a distance of an object reflected from the sensor (Hazra and Santra, 2018; Hazra and Santra, 2019). Alternatively, authors in (Zhang et al., 2017) generate time-frequency spectrograms carrying out in the following order a fast Fourier transform (FFT) and a continuous wavelet transform (CWT). In both cases, data generated by signal processing algorithms are preprocessed and passed to deep learning networks.

Feature extraction in problems related to a hand gesture recognition plays one of the most significant roles in an obtaining a high recognition rate. In order to extract features, we applied a Time-Distributed Convolutional Neural Network that applies the same convolutional neural network to every time step of the gesture sample for an automatic feature extraction. Extracted feature vector is fed to a long-short term memory (LSTM) layer for an analysis of changing frequencies over time. As a result, a feature vector generated by an LSTM is passed to a fully connected layer for a gesture classification.

Inspired by topologies for sequential signal classification (Hazra and Santra, 2018; Hazra and Santra, 2019), we propose an optimized architecture with fewer parameters than the original (Hazra and Santra, 2018) to classify radar signal representing different hand gestures. This paper investigates also a use of alternative signal processing approach based on a direct application of CWT to a raw radar signal, thereby generating the training data for a deep learning algorithm. The main contributions of this paper are as follows:

1. We present an optimized implementation of the deep learning algorithm for recognizing

hand gestures using a Frequency Modulated Continuous Wave (FMCW) radar.

2. We prove the possibility of the deployment on two generations of Raspberry Pi with nearly real time inference time. To the best of our knowledge, for hand gesture recognition with radar, a deep learning algorithm deployed on an embedded computer have not been implemented.
3. We introduce an alternative signal processing for a radar signal based on a CWT, allowing for a generation of scalograms. To the best of our knowledge, this procedure has not been used in the field of radar signal processing.

The rest of this paper is structured as follows: chapter 2 deals with a theoretical background concerning an operation principle of mmWave radar sensor, including details of signal processing; chapter 3 presents a gesture vocabulary proposed in this work; chapter 4 gives an overview concerning the proposed neural network topology; finally, an evaluation and a final discussion are presented in chapters 5 and 6, respectively.

2 RADAR

2.1 Operation Principle of mmWave Radar Sensor

This paper adopts a zigzag wave of FMCW radar sensor. FMCW radars are devices transmitting an electromagnetic power through transmit antennas (continuous wave with a linearly increasing frequency). Such a linearly increasing frequency is called a chirp. The transmitted electromagnetic signal is reflected by the hand (target) and the radar receives the reflected signal after a certain time delay. Both, a transmitted and a received signal are mixed (multiplied) and passed to a low-pass filter in order to generate a raw signal apt for a further signal processing. Through, mixing of signals, intermediate frequency components are extracted, amplified and converted into digital signals by using an Analog to Digital converter (ADC), as it is illustrated in Figure 1.

In this project, the Infineon mmWave radar sensor was employed for solving the gesture recognition task - it is depicted in Figure 2.

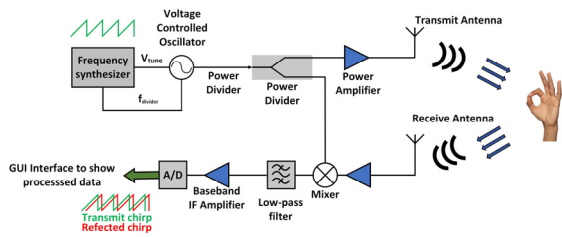


Figure 1: FMCW radar block diagram.

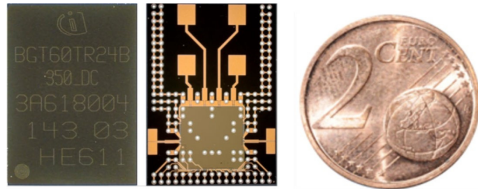


Figure 2: Infineon DEMO BGT60TR24 60GHz mm-Wave radar sensor (Infineon, 2019).

Signal transmitted by an FMCW radar can be expressed in the following form (Lin et al., 2016):

$$S_{TX}(t) = A_{TX} \cos(2\pi f_c t + 2\pi \int_0^t f_T(\tau) d\tau) \quad (1)$$

where $f_T(\tau) = \frac{B}{T} \tau$ is the transmit frequency, f_c is the carrier frequency, B is the bandwidth, T is the signal period, A_{TX} is the amplitude of S_{TX} .

Assuming that the time delay between the transmitted and the received signal is marked as $\Delta\tau$. The Doppler frequency caused by the motion of the target (hand) is denoted by Δf_d , in that way, receive frequency of the moving target can be expressed in the following way (Lin et al., 2016):

$$f_R(t) = \frac{B}{T} (t - \Delta\tau) + \Delta f_d \quad (2)$$

where $\Delta\tau = \frac{2(R+vt)}{c}$, R is the range of the target (hand) from the radar sensor, v is the speed of the target, c is the speed of light, Δf_d is the doppler shift, which is defines as follows formula (Lin et al., 2016):

$$\Delta f_d = -\frac{2f_c v}{c} \quad (3)$$

Signal reflected from the target and thereby received by the radar antenna is expressed with the following formula (Lin et al., 2016):

$$S_{RX}(t) = A_{RX} \cos(2\pi f_c (t - \Delta\tau) + 2\pi \int_0^t f_R(\tau) d\tau) \quad (4)$$

An intermediate frequency (IF) signal generated as a result of mixing the received signal and the transmitted signal and forwarding it to the low-pass

filter is expressed with the following formula (Lin et al., 2016):

$$S_{IF}(t) = \frac{1}{2} \cos(2\pi \left(f_c \frac{2R_0}{c} \right) + 2\pi \left(\frac{2R_0 B}{cT} + \frac{2f_c v}{c} \right) t) \quad (5)$$

where: R_0 is the range at $t = 0$.

2.2 Radar Signal Processing

Data frame received from the radar signal is formed from chirps, while chirps consist of samples. A structure of the radar data frame is depicted in the Figure 3.

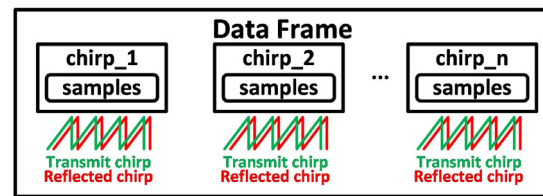


Figure 3: Structure of the radar data frame.

An FMCW radar exhibits a different behaviour depending on the data frame configuration, what allows for a manipulation of various parameters, e.g., range resolution, velocity resolution etc. The raw signal coming from the sensor needs to be preprocessed in order to be able to interpret it and extract from it relevant features for machine learning algorithm. A signal processing technique widely used in the field of radar signal processing is an FFT.

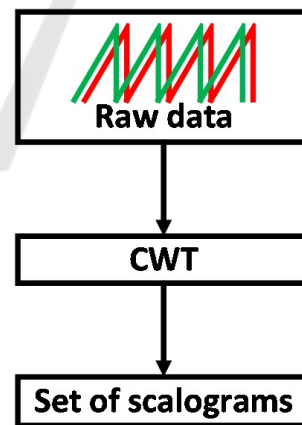


Figure 4: Preprocessing pipeline.

It decomposes a raw signal into a set of sinusoidal waves with different frequencies. In spite of widespread use of FFT in the domain of digital signal processing, it has also disadvantages such as lack of time-frequency resolution compared to CWT (Mallat, 2008) and high degree of computational complexity.

Much research on the gesture recognition with radar has been carried out, authors of (Wang et al., 2016; Zhang et al., 2018; Cai et al., 2019) propose a usage of 2D FFT, applying a first order FFT to resolve the signal in range and a second order FFT to resolve the signal in velocity, thereby generating range-doppler maps representing a distance and a velocity of the target relative to the sensor.

Other works within the scope of gesture recognition with radar (Zhang et al., 2017) propose an application of first order FFT, after that they perform CWT, generating time-frequency spectrograms. This paper introduces an another approach compared to (Wang et al., 2016; Zhang et al., 2018; Cai et al., 2019; Zhang et al., 2017), which avoids a usage of first order FFT as well as second order FFT. Instead of it, we apply directly a CWT to a raw signal, thereby generating CWT maps as an input to the deep learning algorithm. This process is depicted in the Figure 4.

2.2.1 Continuous Wavelet Transform

It is time-frequency analysis method allowing for a calculation of correlation (Mallat, 2008) between an analyzed signal and a wavelet function $\Psi(t)$. The similarity between signal under consideration is calculated separately for each time interval, what results in two-dimensional representation.

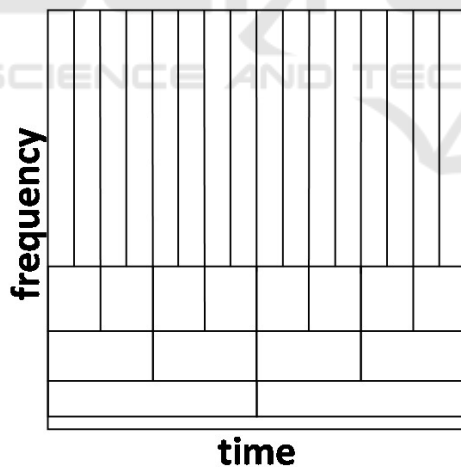


Figure 5: Multiresolution time-frequency plane.

CWT decomposes the signal under consideration into set of wavelets (Mallat, 2008). Wavelets are functions (oscillations) being highly localized in time with zero average (Mallat, 2008).

$$\int_{-\infty}^{\infty} \Psi(t) dt = 0 \quad (6)$$

Given that, wavelets are highly localized in time, they can be correlated (convolved) with the signal under consideration at different locations in time. From the mathematical perspective, CWT is described with the following equation (Mallat, 2008):

$$\int_{-\infty}^{\infty} f(t) \frac{1}{\sqrt{s}} \Psi\left(\frac{t-u}{s}\right) dt = f * \Psi(u) \quad (7)$$

where: s is a scale factor, t is a time, u is a translation, f is a function representing the signal to be analyzed, $\Psi(t)$ is a mother wavelet.

3 PROPOSED GESTURES

The gesture vocabulary was chosen after numerous tests and thorough literature research. The dataset was defined in such a way in order to be able to perceive substantial dissimilarities between individual gesture samples. A separate gesture sample is comprised of 20 data frames from two receiving antennas. Each individual gesture lasts one second, with a frame interval chosen to be $50 \mu s$. In addition, the gesture set is composed of 5819 samples, every gesture was performed by 4 individuals without giving precise instructions how the gesture should be performed.

1. Circle
2. Up-down
3. Down-up
4. Left-right (swipe)

4 PROPOSED NN ARCHITECTURE

4.1 Deep Learning Model

Topology proposed by authors was implemented in Keras (Chollet et al., 2015) with TensorFlow (Abadi et al., 2016) backend. It is built from two components:

- Time Distributed convolutional component – visual features extraction component;
- Recurrent component – time modelling component (Hu et al., 2020);

4.2 Theoretical Background

4.2.1 Convolutional Layer

This layer employs a mathematical operation called convolution. It automatically extracts visual features,

convolving training convolutional filters K with an input feature space V . Therefore, convolution is described with the following equation (Goodfellow et al., 2016):

$$Z_{i,j,k} = \sum_{l,m,n} [V_{l,(j-1) \times s + m, (k-1) \times s + n} K_{i,l,m,n}] \quad (8)$$

where: i, j, k are i_{th} channel, j_{th} row and k_{th} column of the output feature space Z and s is a stride parameter. Whereas l is l_{th} channel in an input feature space V and m and n are correspondingly m_{th} row and n_{th} of kernel K .

4.2.2 Timedistributed Layer

The layer wrapper applying the same layer to every timestep of an input, thereby enabling a feature extraction from the data having a temporal characteristics (Chollet et al., 2015).

4.2.3 Recurrent Unit

The temporal feature modelling is performed using an LSTM unit that is mainly comprised of three foundations: a forget gate, an input gate and an output gate, controlling an information flow. The input provided to an LSTM is fed to different gates, controlling which operation is performed on the cell memory: write (input gate), read (output gate) or reset (forget gate). The vectorial representation of an LSTM layer is as follows:

$$i(t) = \sigma_i(W_{ai}a(t) + W_{hi}h(t-1) + W_{ci}c(t-1) + b_i)$$

$$f(t) = \sigma_f(W_{af}a(t) + W_{hf}h(t-1) + W_{cf}c(t-1) + b_f)$$

$$c(t) = f(t)c(t-1) + i(t)\sigma_c(W_{ac}a(t) + W_{hc}h(t-1) + b_c)$$

$$o(t) = \sigma_o(W_{ao}a(t) + W_{ho}h(t-1) + W_{co}h(t) + b_o)$$

$$h(t) = o(t)\sigma_h(c(t))$$

(9)

where i, f, o, c and h denote respectively the input gate, forget gate, output gate, cell activation vectors and hidden states. The terms σ represent activation functions. However, the vector $X = \{x(1), x(2), \dots, x(T)\}$ denotes an input to the memory cell layer at time t , while $W_{ci}, W_{hi}, W_{ai}, W_{cf}, W_{hf}, W_{af}, W_{hc}, W_{ac}, W_{co}, W_{ho}, W_{ao}$ are weight

matrices and subscripts mean from-to relationships. The terms b_i, b_f, b_c and b_o are the bias vectors.

4.3 Implementation Details

Due to the nature of the data, the proposed deep learning algorithm is comprised of two components: one for visual features extraction from the sequence of CWT maps, followed by the component modelling the temporal features – an LSTM. Basically, the first component is built from the five convolutional layers with successively increasing number of filters, 16, 32, 64, 96 and 128. The first four convolutional layers are followed by a BatchNormalization, ReLu activation, MaxPooling2D and Dropout2D, however the last convolutional layer is followed by a BatchNormalization (Ioffe and Szegedy, 2015) with a ReLu activation. Each of convolutional layers utilizes 3x5 kernel size with stride 1 and same padding. Weights of convolutional layers are initialized using glorot initializer, whereas biases are initialized with zeros. First three convolutional layers are followed by MaxPooling2D with pool size 2x2 and stride 2 in both dimensions and Dropout2D with dropout rate 0.5, however, the fourth convolutional layer is followed by MaxPooling2D with 1x2 pool size and Dropout2D with 0.3 dropout rate.

In order to reduce an overfitting effect, we applied two regularization techniques: an l2 and a dropout (Srivastava et al., 2014). This set of operations is wrapped in a sequential module and fed to the TimeDistributed layer, applying the same set of operations to every timestep. Visual feature extraction module is depicted in the Figure 6. The output of visual feature extractor is passed to the MaxPooling3D in order to reduce the dimensionality, it is flattened, dropped out with 0.3 dropout rate, and fed to the recurrent layer. Linear and recurrent kernels of the LSTM layer are initialized using a glorot initializer and an orthogonal initializer, while bias is initialized with zeros. In order to avoid an overfitting effect and increase an overall system performance, we have applied a dropout to linear transformations (Gal and Ghahramani, 2016; Pascanu et al., 2013) and to recurrent connections (Gal and Ghahramani, 2016; Pascanu et al., 2013) between LSTM units. L2 regularization was applied to kernel, recurrent connections and bias.

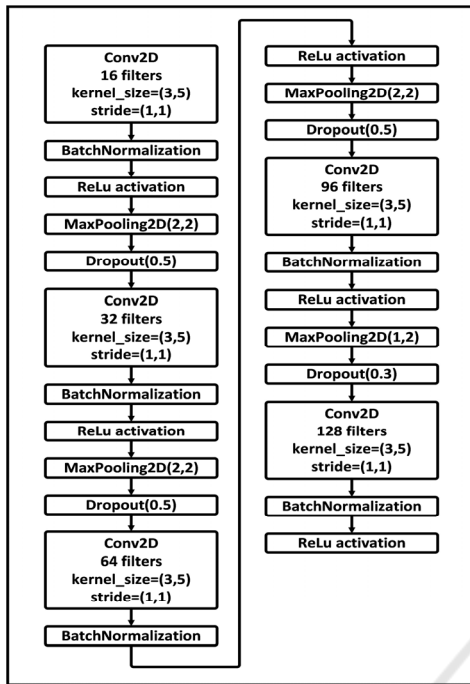


Figure 6: Visual feature extraction module.

The LSTM utilizes a tanh activation and sigmoid as a recurrent activation. The feature vector generated by the LSTM is fed to fully-connected layer with a softmax activation in order to perform the final classification.

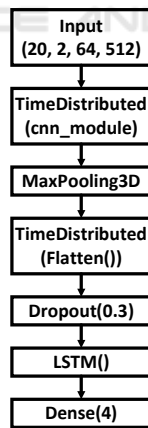


Figure 7: Final architecture.

After many tests and experimentations with hyperparameters, the final model was trained with categorical cross-entropy loss function with adaptive moment optimization algorithm (ADAM). The final architecture is presented in the Figure 7.

5 EVALUATION

System defines four gestures. In general, one has gathered 5819 – correspondingly circle class – 1500 samples, up-down class - 1415 samples, down-up class – 1404, left-right class – 1500 samples. Each gesture sequence is composed of 20 data frames which are subsequently transformed into sequence of 20 CWT maps lasting 1 s. The dataset has been split in proportion 70%/30%, train and validation split, respectively.

Table 1: Comparative Characteristics of Proposed Model.

Model	Accuracy	Inference time	Size
ConvNet+LSTM (Hazra and Santra, 2018)	94,37%	1,0s	13,6MB
Proposed NN architecture	95,05%	0,2s (x86 processor)	7,43MB
Proposed NN architecture	95,05%	2,0s (Raspberry Pi3)	7,43MB
Proposed NN architecture	95,05%	1,5s (Raspberry Pi4)	7,43MB

To prove the performance of our model, we have created separate test dataset consisting of 2001 samples and we have tested this model against the test dataset. Table 1 presents a comparative characteristics. We can observe, our model achieves 95.05% accuracy compared with (Hazra and Santra, 2018), consuming less hard disk space and offering shorter inference time. Proposed system was tested on the laptop with the following specification – 8GB RAM, i7-6700HQ @ 2.60GHz – being able to perform an inference within 0.2s. Additionally, system was also tested on both versions of RaspberryPi 3 and 4, thereby achieving an inference time of 2.0s and 1.5s, what creates good perspectives for the future development.

Table 2: Parameters Summary.

Symbol	Parameter name	Value
P_{Tx}	Transmit power	31
N_{chirps}	Number of chirps	1
N_{Sper_chirp}	Number of samples per chirp	512
$N_{RXantennas}$	Number of RX antennas	2
TX_{Mode}	TX mode	Use only TX2
V_{Gain}	VGA gain	+10dB

Data for training, validation and testing were collected with parameters presented in Table 2. During boot-up, system gathers 100 raw-data frames and it calculates mean from them.

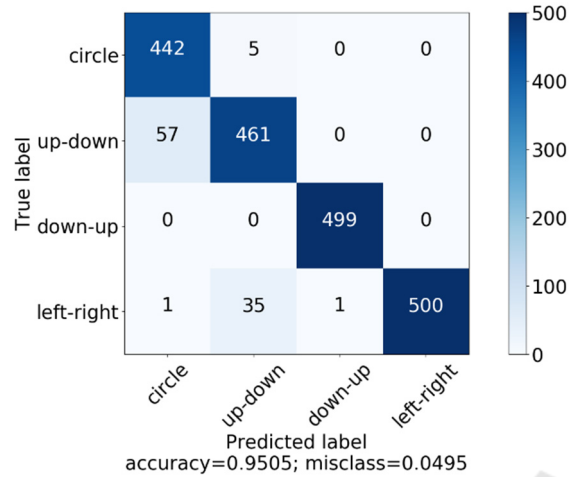


Figure 8: Confusion matrix.

Mean frame is used for gesture detection. Namely, it calculates root mean square between current frame and mean frame. In case of exceeding the threshold, system starts data gathering for period of 1s. Figure 8 depicts the confusion matrix. We can observe that circle is classified with the lowest accuracy rate. In 57 cases, it was confused with up-down gesture and in one case with left-right (swipe) gesture.

This is explicable because left-right gesture as well as up-down gesture have relatively similar characteristic to circle gesture, what is illustrated in the following figures: Figure 9, Figure 11 and Figure 12.

As far as up-down and down-up gestures are concerned, first of them is sometimes confused with circle in 5 cases and with left-right in 35 cases. However, down-up is misclassified only in one case. Swipe gesture is classified with 100% accuracy. Moreover, our model was tested on Raspberry Pi3 and Raspberry Pi4, achieving nearly real-time inference time: 2.0s and 1.5s.

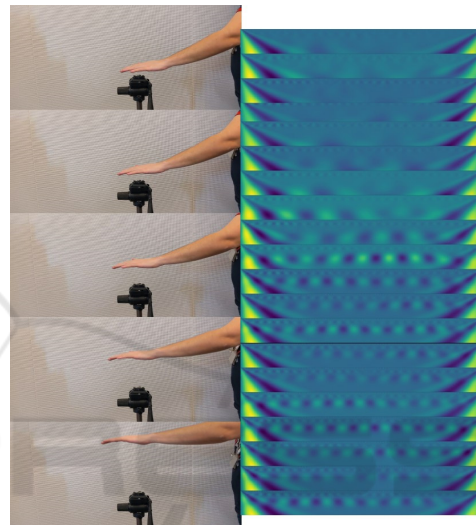


Figure 10: Down-up.

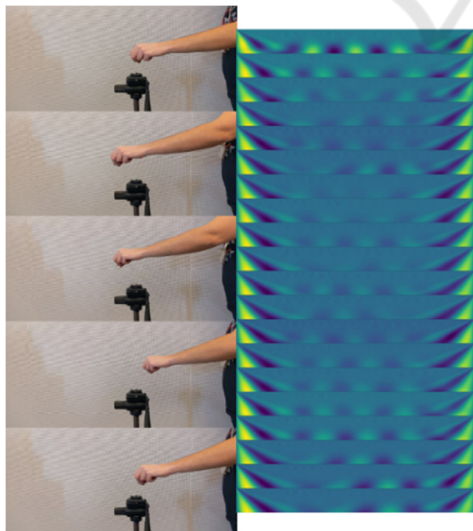


Figure 9: Circle.

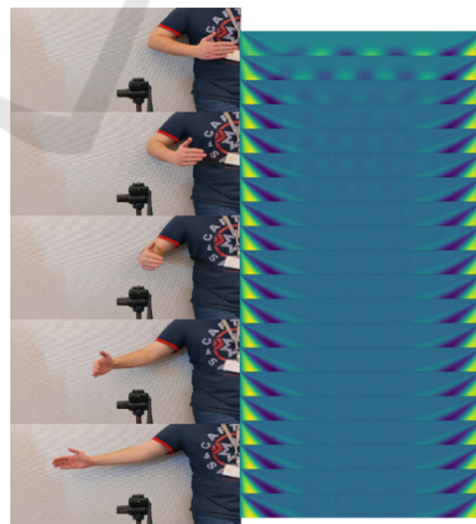


Figure 11: Left-right.

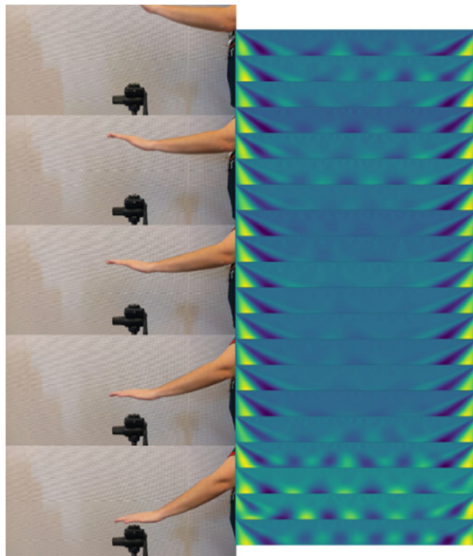


Figure 12: Up-down.

6 FINAL DISCUSSION

Gestures are a standard means of communication used by people to exchange an information between each other. Thus, it would also be natural for people to use them to communicate with computers. Because of this, the applicability of gestures in a human-computer interaction seems to be relevant topic from the scientific point of view. This paper proposes a hand gesture recognition system using a dedicated CNN-LSTM architecture. Our solution employs a use of FMCW radar in conjunction with the low-power microcomputer(s) Raspberry Pi3, Raspberry Pi4 and deep learning techniques. The proposed model achieves good performance on earlier unseen data. In comparison to (Hazra and Santra, 2018), our model achieves a real-time interaction performance on x86 class CPU and nearly real-time interaction performance on ARMv8 class CPU(s). It uses less number of parameters, what implies smaller size of model, possibility of deployment on the low-power micro-computer. In the future, we are planning to introduce sensor-fusion capability and support for user defined gestures.

REFERENCES

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I. J., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mane, D., Monga, R., Moore, S., Murray, D. G., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P. A., Vanhoucke, V., Vasudevan, V., Viegas, F. B., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2016). Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *CoRR*, abs/1603.04467.
- Ahmed, S. and Cho, S. H. (2020). Hand gesture recognition using an ir-uwb radar with an inception module-based classifier. *Sensors*, 20(2):564.
- Alemuda, F. and Lin, F. J. (2017). Gesture-based control in a smart home environment. In *2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pages 784–791.
- Cai, X., Ma, J., Liu, W., Han, H., and Ma, L. (2019). Efficient convolutional neural network for fmcw radar based hand gesture recognition. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers, UbiComp/ISWC '19 Adjunct*, page 17–20, New York, NY, USA. Association for Computing Machinery.
- Chai, X., Liu, Z., Yin, F., Liu, Z., and Chen, X. (2016). Two streams recurrent neural networks for large-scale continuous gesture recognition. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 31–36.
- Chollet, F. et al. (2015). Keras.
- Gal, Y. and Ghahramani, Z. (2016). A theoretically grounded application of dropout in recurrent neural networks.
- Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. The MIT Press.
- Hazra, S. and Santra, A. (2018). Robust gesture recognition using millimetric-wave radar system. *IEEE Sensors Letters*, 2(4):1–4.
- Hazra, S. and Santra, A. (2019). Short-range radar-based gesture recognition system using 3d cnn with triplet loss. *IEEE Access*, 7:125623–125633.
- Hu, C., Hu, Y., and Seo, S. (2020). A deep structural model for analyzing correlated multivariate time series.
- Infineon (2019). Internal technical documentation. Technical report, Infineon Technologies AG.
- Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift.
- Lai, K. and Yanushkevich, S. N. (2018). Cnn+rnn depth and skeleton based dynamic hand gesture recognition. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 3451–3456.
- Li, D., Chen, Y., Gao, M., Jiang, S., and Huang, C. (2018). Multimodal gesture recognition using densely connected convolution and blstm. In *2018 24th*

- International Conference on Pattern Recognition (ICPR)*, pages 3365–3370.
- Lin, J.-J., Li, Y.-P., Hsu, W.-C., and Lee, T.-S. (2016). Design of an fmcw radar baseband signal processing system for automotive application. *SpringerPlus*, 5(1):42.
- Ma, X. and Peng, J. (2018). Kinect sensor-based longdistance hand gesture recognition and fingertip detection with depth information. *Journal of Sensors*, 2018.
- Mallat, S. (2008). *A Wavelet Tour of Signal Processing, Third Edition: The Sparse Way*. Academic Press, Inc., USA, 3rd edition.
- Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks.
- Patole, S. M., Torlak, M., Wang, D., and Ali, M. (2017). Automotive radars: A review of signal processing techniques. *IEEE Signal Processing Magazine*, 34(2):22–35.
- Schiff, J., Meingast, M., Mulligan, D. K., Sastry, S., and Goldberg, K. (2009). Respectful cameras: Detecting visual markers in real-time to address privacy concerns. In *Protecting Privacy in Video Surveillance*, pages 65–89. Springer.
- Shanthakumar, V. A., Peng, C., Hansberger, J., Cao, L., Meacham, S., and Blakely, V. (2020). Design and evaluation of a hand gesture recognition approach for real-time interactions. *Multimedia Tools and Applications*, pages 1–24.
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958.
- Tran, D.-S., Ho, N.-H., Yang, H.-J., Baek, E.-T., Kim, S.H., and Lee, G. (2020). Real-time hand gesture spotting and recognition using rgb-d camera and 3d convolutional neural network. *Applied Sciences*, 10(2):722.
- Van den Bergh, M. and Van Gool, L. (2011). Combining rgb and tof cameras for real-time 3d hand gesture interaction. In *2011 IEEE Workshop on Applications of Computer Vision (WACV)*, pages 66–72.
- Wan, Q., Li, Y., Li, C., and Pal, R. (2014). Gesture recognition for smart home applications using portable radar sensors. In *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 6414–6417.
- Wang, J. and Payandeh, S. (2017). Hand motion and posture recognition in a network of calibrated cameras. *Advances in Multimedia*, 2017.
- Wang, S., Song, J., Lien, J., Poupyrev, I., and Hilliges, O. (2016). Interacting with soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*, UIST '16, page 851–860, New York, NY, USA. Association for Computing Machinery.
- Yasen, M. and Jusoh, S. (2019). A systematic review on hand gesture recognition techniques, challenges and applications. *PeerJ Computer Science*, 5:e218.
- Yunan Li, Qiguang Miao, Kuan Tian, Yingying Fan, Xin Xu, Rui Li, and Song, J. (2016). Large-scale gesture recognition with a fusion of rgb-d data based on the c3d model. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 25–30.
- Zhang, J., Tao, J., and Shi, Z. (2017). Doppler-radar based hand gesture recognition system using convolutional neural networks. In *International Conference in Communications, Signal Processing, and Systems*, pages 1096–1113. Springer.
- Zhang, Z., Tian, Z., and Zhou, M. (2018). Latern: Dynamic continuous hand gesture recognition using fmcw radar sensor. *IEEE Sensors Journal*, 18(8):3278–3289.