Vision Transformer with Convolutional Encoder-Decoder for Hand Gesture Recognition using 24 GHz Doppler Radar

Kavinda Kehelella[1], Gayangana Leelarathne[1*], Dhanuka Marasinghe[1*], Nisal Kariyawasam[1], Viduneth Ariyarathna[2**], Arjuna Madanayake[3**], Ranga Rodrigo[1***], and Chamira U. S. Edussooriya[1,3**]

[1] Department of Electronic and Telecommunication Engineering, University of Moratuwa, Moratuwa 10400, Sri Lanka
[2] Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02120, USA
[3] Department of Electrical and Computer Engineering, Florida International University, Miami, FL 33174, USA
* Student Member, IEEE
** Member, IEEE
*** Senior Member, IEEE

Abstract—Transformers combined with convolutional encoders have been recently used for hand gesture recognition (HGR) using micro-Doppler signatures. We propose a vision-transformer-based architecture for HGR with multi-antenna continuous-wave Doppler radar receivers. The proposed architecture consists of three modules: a convolutional encoder-decoder, an attention module with three transformer layers, and a multi-layer perceptron. The novel convolutional decoder helps to feed patches with larger sizes to the attention module for improved feature extraction. Experimental results obtained with a dataset corresponding to a two-antenna continuous-wave Doppler radar receiver operating at 24 GHz (published by Skaria *et al.*) confirm that the proposed architecture achieves an accuracy of 98.3% which substantially surpasses the state-of-the-art on the used dataset.

Index Terms—Vision transformers, attention mechanisms, deep learning, micro-Doppler signatures, hand gesture recognition.

I. INTRODUCTION

Hand gesture recognition (HGR) plays a vital role in human computer interactions, augmented/mixed reality, and human-machine teaming, where specific gestures made by the human hands may be used to control electronics systems. Examples include interfaces to smart phones, vending machines, drones/robots, and gaming devices [1]–[3]. On-body device-based approaches, where a person wears or carries a device (e.g., inertial sensors, or radio frequency (RF) identification tags) have been employed for HGR. Device-free approaches such as computer vision, acoustic sensing, and RF sensing have also been employed [2], [3]. Compared to device-based sensing, device-free sensing is user friendly and widely adopted [2], [3]. Vision-based and acoustic-based device-free approaches are vulnerable to environmental conditions such as light intensity, rain, smoke, and external noise, while suffering from privacy issues. On the contrary, RF sensing using radar is not as vulnerable to environmental conditions and do not significantly violate privacy. Compared to WiFi-based sensing that operates at 2.4 GHz or 5.8 GHz, radar-based sensing that operates in millimeter waves (e.g., at 60 GHz) can detect very small movements of a hand/finger [3], [4].

Sensing with continuous-wave (CW) [5], [6] and frequency-modulated CW radars [7]–[10] predominantly utilizes micro-Doppler signatures for HGR or human activity recognition. Dynamics of a moving object induce Doppler modulation on the reflected signal when an RF signal strikes the object in motion [11]. CW radars capture micro-Doppler signatures without range information whereas frequency-modulated CW radars capture both micro-Doppler signatures and range information. Recent works [5], [10], [12] demonstrated that radars with multi-antenna receivers achieve higher accuracy than single-antenna radar receivers for HGR applications.

Radio-frequency machine learning has been utilized to recognize hand gestures with micro-Doppler signatures, where pseudo images generated from received RF signals (e.g., spectrograms and time-Doppler maps) were used as the input [1]. In [5], [7], convolutional neural networks (CNNs) and in [13], auto-encoders with long short-term memory (LSTM) were employed for HGR. With the popularity of attention-based models with transformers, first employed in natural language processing [14] and subsequently adopted to computer vision tasks [15], recent works on HGR exploited CNNs together with transformers. In [16], a deep residual three-dimensional CNN with a transformer network was used for HGR with a dataset from [4]. A CNN with one-dimensional convolution/correlation and attention-based network was used in [17] to classify human activities. In [18], an attention+CNN approach via an LSTM was used to recognize human gestures performed from a distance. A CNN feature extractor with an attention-based network was used in [19] for person and activity recognition. In [20], a CNN and a vision-transformer were used for HGR for in-vehicle environments. In these transformer networks, the output feature map of the CNN was directly fed as the input to the transformer. Due to the lower spatial size of the output feature map compared to the input pseudo image to the CNN, the direct feeding leads to patches with lower spatial sizes in the transformers; however, such patches with lower spatial size may hinder the full-potential of transformers [15].

In this paper, we propose a vision-transformer-based architecture for HGR using multi-antenna CW radar. The architecture consists of three modules: 1) a convolutional encoder-decoder, 2) an attention module with three transformer layers, and 3) a multi-layer perceptron (MLP). Compared to previous works on transformers, our architecture employs *a convolutional decoder* to up-sample the output feature map of the convolutional encoder before feeding to the vision transformer. This enables us to use a relatively large patch sizes in the vision transformer as well as to train with relatively small datasets. We employ the dataset from [5], where a two-antenna CW Doppler radar receiver was employed, for validating our algorithms with experiments. The proposed architecture achieves an accuracy of 98.3% which substantially surpasses the accuracy achieved in [5].

II. PROPOSED TRANSFORMER ARCHITECTURE

The proposed architecture, shown in Fig. 1, consists of a convolutional encoder-decoder, an attention module, and an MLP. We consider a CW radar with one transmit antenna and two receive

Fig. 1: Proposed architecture with a five layer convolutional encoder, a two layer convolutional decoder, an attention module with three vision-transformer layers, and an MLP for classification.

Table 1: Specifications of the convolutional layers in the encoder.

Layer	Kernel size	Number of filters	Output size
Input	-	-	$180 \times 60 \times 3$
Conv 1	7×7	4	$180 \times 60 \times 4$
Conv 2	5×5	8	$90 \times 30 \times 8$
Conv 3	3×3	16	$45 \times 15 \times 16$
Conv 4	3×3	32	$23 \times 8 \times 32$
Conv 5	3×3	64	$12 \times 4 \times 64$

antennas together with a coherent mixer with in-phase and quadrature. The dataset contains 14 gestures. See [5] for a system overview and more details on the dataset. The input to the model is a three-channel RGB image generated from spectrograms from two receiver antennas and the angle of arrival of signals obtained from the phase difference between two receiver antennas [5].

The convolutional encoder consists of five CNN layers, each followed by a max pooling layer. The CNN layers learn features required for subsequent processing with the attention module. Furthermore, the max pooling layers reduce noise in the input and downsample feature maps [5]. The decoder is used to increase the spatial size of the output feature map from the encoder because the size of the output feature map is too small to generate patches for the attention module. The extracted features from the convolutional encoder-decoder are divided into patches, added with positional embeddings [14], [15] and given as the input to the attention module. The attention mechanism allows the modeling of dependencies regardless of their distance in the input or output sequences [14]. We employ three transformer layers in the attention module. These transformer layers are developed using the vision-transformer models in [15], where multi-head attention is used in each layer. The MLP is employed as the classifier. Next, we describe each module in detail.

A. Convolutional Encoder-Decoder Architecture

Convolutional encoder-decoder is primarily used to extract features from input images. The number of filters (kernels) and the size of kernels of the five convolutional/correlation layers are presented in Table 1. The size of the input is $180 \times 60 \times 3$, and the input is normalized to the range [0, 1] before feeding to the encoder for faster convergence of the model. Rather than feeding the feature maps obtained from the encoder having a size $6 \times 2 \times 64$ directly to the attention module, we use a convolutional decoder with transposed convolution to up-sample the feature vectors. The convolutional decoder is used to enhance the spatial size of feature maps. There are two transposed convolutional layers with 64 filters with 1×1 kernels and stride 2×2. The resultant feature map from the convolutional decoder of size $24 \times 8 \times 64$ is fed to the attention module. .

B. Attention Module

In the attention module, shown in Fig. 2, the input feature map $\mathbf{x_u} \in \mathbb{R}^{8 \times 24 \times 64}$ is divided into a sequence of square patches $\mathbf{x_p} \in \mathbb{R}^{N \times (p^2 \times 64)}$,

where N is the number of patches and p is the height and width of a patch. Note that N can be calculated as $N = 8 \times 24/p^2$. We use $p = 4$, and as a result, the $\mathbf{x_u}$ is divided into 12 patches. Then the tensor $\mathbf{x_p}$ is flattened to produce a tensor $\mathbf{x_f} \in \mathbb{R}^{12 \times 1024}$, which is subsequently projected linearly by a fully-connected layer to produce $\mathbf{x_e} \in \mathbb{R}^{12 \times 16}$ patch embeddings. Next, patch embeddings are added with positional embeddings $\mathbf{x_{pos}} \in \mathbb{R}^{12 \times 16}$ to integrate the positional information [14], [15] to generate input embeddings $\mathbf{x_i} \in \mathbb{R}^{12 \times 16}$. The sequence of input embeddings then serves as the input to the vision transformer, where $\mathbf{x_i}$ is normalized and fed into the multi-head attention module. Here query $\mathbf{Q} \in \mathbb{R}^{12 \times 16}$, key $\mathbf{K} \in \mathbb{R}^{12 \times 16}$ and value $\mathbf{V} \in \mathbb{R}^{12 \times 16}$ matrices are generated by normalizing $\mathbf{x_i}$, i.e., $\mathbf{Q} = \mathbf{K} = \mathbf{V} = Norm(\mathbf{x_i})$.

The multi-head attention module contains independent self-attention modules that operate in parallel. Multi-head attention has the advantage of allowing the model to jointly attend to information from several representation subspaces [14]. We select the number of heads h as four with the projection dimension $d_k = d_q = d_v$ as 16. The multi-head attention can be performed on $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ matrices as

$$MultiHead(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = Concat\left(\mathbf{H}_1, \mathbf{H}_2, ..., \mathbf{H}_j, ..., \mathbf{H}_h\right)\mathbf{W}^O, \quad (1)$$

$$\mathbf{H}_j = Attention(\mathbf{Q}\mathbf{W}_j^Q, \mathbf{K}\mathbf{W}_j^K, \mathbf{V}\mathbf{W}_j^V), \quad (2)$$

where $\mathbf{W}_j^Q \in \mathbb{R}^{d_k \times \frac{d_k}{h}}$, $\mathbf{W}_j^K \in \mathbb{R}^{d_k, \frac{d_k}{h}}$, $\mathbf{W}_j^V \in \mathbb{R}^{d_k \times \frac{d_k}{h}}$ and $\mathbf{W}^O \in \mathbb{R}^{d_k \times d_k}$ are learnable projection matrices, and $Attention(\hat{\mathbf{Q}}, \hat{\mathbf{K}}, \hat{\mathbf{V}})$ is defined as

$$Attention(\hat{\mathbf{Q}}, \hat{\mathbf{K}}, \hat{\mathbf{V}}) = softmax\left(\frac{\hat{\mathbf{Q}}\hat{\mathbf{K}}^T}{\sqrt{d_k}}\right)\hat{\mathbf{V}}. \quad (3)$$

Here, $\hat{\mathbf{Q}}$, $\hat{\mathbf{K}}$, $\hat{\mathbf{V}}$ are projected matrices from $\mathbf{Q}$, $\mathbf{K}$, and $\mathbf{V}$ matrices, respectively [14], [15]. Next, to provide a residual connection to the transformer, the input embeddings and output from the multi-head attention layer are added together. This tensor is then normalized and fed into the MLP block, which consists of two layers with 32 and 16 neurons. The output of the transformer layer is fed into the next transformer layer. In our model, we employ three transformer layers (l) as shown in Fig. 2. Finally, the output of the third transformer layer is flattened and fed into the MLP classifier.

C. Multi-Layer Perceptron

We use an MLP as the classifier with categorical cross-entropy loss function, which is typically employed for multi-class classification problems. The MLP consists of two fully-connected layers followed by a softmax layer with 14 units for the 14 gestures. Fully-connected layers have 1024 neurons and 512 neurons with 0.5 dropout to reduce overfitting. Softmax layer calculates corresponding class probabilities for every class. Then the class labels (y) can be predicted by performing $\arg\max(\cdot)$ function on the softmax output that returns the index corresponding to the largest probability from the output class probabilities.

Fig. 2: Attention module with patch and positional embeddings and three transformer layers.

III. EXPERIMENTAL RESULTS

A. Dataset and Training

We use the dataset in [5] for experimental validation. This dataset was collected using Infineon radar development board BGT24MTR12 operating at 24 GHz with a single transmit antenna and an array of two receiving antennas with each receiver producing in-phase and quadrature components. The dataset consists of 14 hand gestures: (1) single blink, (2) double blink, (3) single push-pull, (4) double push-pull, (5) single round, (6) double round, (7) single swipe, (8) double swipe, (9) single thumbs up, (10) double thumbs up, (11) single waving, (12) double waving, (13) single slide, and (14) double slide, each having 250 samples. Each sample is a three-channel image of size 180×60, where the first two channels are the spectrograms generated from the signals received by two antennas, and the third channel contains the angle of arrival matrix which was generated from the phase difference between the received signals.

We use 80% of the samples for training and 20% of the samples for testing. The validation accuracy is obtained from the five-fold cross validation on the training dataset. Moreover, the test accuracy is obtained by independently training the model five times. We present the averages and the standard deviations of test accuracies in the next subsection. The weighted adaptive moment (AdamW) optimizer is used for training with a learning rate of 0.001 and a weight decay of 0.0001. We train the model for 100 epochs with a batch size of 64. The transformers allow for substantially higher parallelization, and hence we used an NVIDIA Tesla T4 GPU to train the model.

B. Experimental Results

1) Optimum Parameters of the Attention Module: The parameters and hyper-parameters of the proposed architecture are tuned to achieve the best validation accuracy. The height and width of a patch p and the number of dimensions d_k in the linear projection play a critical role in complexity and performance of the proposed architecture. We analyze the performance of the attention module with respect to these two parameters under three criteria as presented in Table 2. Here, we change p and the number of dimensions in the linear projection with fixed parameters for the convolutional encoder-decoder and three transformer layers. We observe that our model with $p = 4$ and $d_k = 16$ achieves the best accuracy of 98.3%. Furthermore, our model has the least number of parameters, with a 25% reduction compared to the next best model. It is evident that attention model with $p = 4$ and and $d_k = 16$ is the *best among the potential candidates*.

Table 2: Variation of the test accuracy (avg ± std), the F1-score and the number of total and trainable parameters with the patch size (p) and the projection dimension (d_k).

p	d_k	Total parameters	Trainable parameters	Test accuracy (%)	F1-score (%)
2	16	1,375,190	1,374,942	95.2 ±0.55	95.1
2	32	2,213,574	2,213,326	95.8 ±0.65	95.8
2	64	3,982,502	3,982,254	94.9 ±0.70	94.7
4	**16**	**797,078**	**796,830**	**98.3 ±0.50**	**98.4**
4	32	1,057,350	1,057,102	97.7 ±0.58	97.5
4	64	1,670,054	1,669,806	95.6 ±0.39	95.5

Table 3: Variation of the test accuracy (avg ± std), the F1-score and the number of total and trainable parameters with the number of transformer layers (l).

l	Total Parameters	Trainable Parameters	Test accuracy (%)	F1-score (%)
1	786,198	785,950	96.0 ±0.75	96.1
2	791,638	791,390	94.5 ±0.79	94.3
3	**797,078**	**796,830**	**98.3 ±0.50**	**98.4**
4	802,518	802,270	97.5 ±0.53	97.4
5	807,958	807,710	97.3 ±0.75	97.5
6	813,398	813,150	96.9 ±0.67	96.9

The number of transformer layers l in the attention module is the other important parameter. A model with more transformer layers has the ability to extract more informative features, however, at the same time, the model complexity increases. Moreover, more transformer layers tend to increase the variance of the model because of the relatively small dataset that was used. We analyzed the performance of our model by varying l from 1 to 6 while fixing other parameters, and the results are presented in Table 3. We can see that *the best accuracy of* 98.3% *is achieved with* $l = 3$, which is selected for our architecture, even though the number of total and trainable parameters are slightly higher compared to $l = 1$ and $l = 2$.

2) Ablation Study: Our architecture is composed with both convolutional encoder-decoder and an attention module. Since the state-of-art deep neural network model for this dataset is a CNN, we select the CNN in our model as the baseline. We employ an ablation study to verify the improved performance with the convolutional decoder and the attention module. We also trained the standalone modules (decoder and attention module) on the dataset and the results are presented in Table 4.

When training with standalone attention module, three-channel images are split into patches with $p = 36$ and provided as the input to the encoder. Furthermore, for the combination of the convolutional encoder and the attention module, $p = 1$ and $d_k = 16$ are selected.

We observed that the standalone convolutional encoder outper-

Table 4: Ablation study with different combinations of modules.

Architecture	Test accuracy (%)	F1-score (%)
Convolutional Encoder	96.7 ±0.41	96.7
Convolutional Encoder-Decoder	96.7 ±0.59	96.6
Attention Module	91.9 ±0.73	91.8
Convolutional Encoder + Attention Module	96.0 ±0.67	95.9
Convolutional Encoder-Decoder + Attention Module	**98.3 ±0.50**	**98.4**

Table 5: Classification performance achieved with different models.

Architecture	Test accuracy	Precision	Recall	F1-score (%)
CNN [5]	95.1 ±0.54	95.0	95.1	95.1
ResNet50 [21]	92.1 ±0.62	92.0	92.1	92.0
VGGNet16 [22]	95.0 ±0.61	94.8	94.7	94.7
Our model	**98.3 ±0.50**	**98.4**	**98.4**	**98.4**

formed the standalone attention module by 4.8 percentage points. This is because convolution has inductive bias such as translation invariance which lacks in transformers [15]. Therefore, when trained on small datasets, transformers do not generalize the model. Furthermore, the combination of the convolutional encoder and the attention module (with $p = 1$ and $d_k = 16$) outperforms the standalone attention module by 4.1 percentage points. More importantly, our architecture outperforms the four other approaches. In particular, our architecture achieves 2.3 percentage points higher accuracy than that of the combination of the convolutional encoder and the attention module *verifying the importance of the convolutional decoder.*

3) Comparison with Other Architectures: We compare accuracies achieved with our model, CNN architecture in [5], and two popular image classification deep neural network models: ResNet50 [21] and VGGNet16 [22], pretrained on ImageNet dataset. We modify the last softmax layer of both ResNet50 and VGGNet16 to have 14 classes. In transfer learning, we train only the parameters of the last layer of both networks while freezing other layers with pretrained parameters. The achieved test accuracies are presented in Table 5. Our model outperforms other three models, in particular, the original work [5] by a margin of 2.8 percentage points. Similar to other transformer-based architectures, a limitation of the proposed architecture is that the number of total parameters (797,078) is considerably higher than that of the CNN architecture (7676) [5] leading to higher computational and memory complexities. The interference times of the CNN [5], ResNet50 [21], VGGNet16 [22] and the proposed architectures are 35 ms, 130 ms, 170 ms, and 96 ms, respectively. The inference time of the proposed architecture is higher than that of the CNN architecture [5] and lower than those of the ResNet50 [21] and VGGNet16 [22] architectures. Even though the proposed architecture has more layers and parameters compared to the CNN [5], the inference time is only ≈3 times higher due to the highly parallel implementation of the attention module. Future work may consider the reduction of the complexity of the proposed architecture using pruning techniques [23].

IV. CONCLUSION

We propose a vision-transformer-based architecture for HGR with multi-antenna CW Doppler radar receivers. The convolutional decoder up-samples the output feature map of the convolutional encoder enabling us to feed patches with larger sizes to the attention module. This results in improved feature extraction in the attention module. Experimental results confirmed that the proposed architecture achieves an accuracy of 98.3% which is substantially higher than the state-of-the-art on the used dataset, and the ablation study confirms that the convolutional decoder improves the accuracy on HGR.

REFERENCES

[1] S. Ahmed, K. D. Kallu, S. Ahmed, and S. H. Cho, "Hand gestures recognition using radar sensors for human-computer-interaction: A review," *Remote Sensing*, vol. 13, no. 3, pp. 1–24, Feb. 2021.

[2] I. Nirmal, A. Khamis, M. Hassan, W. Hu, and X. Zhu, "Deep learning for radio-based human sensing: Recent advances and future directions," *IEEE Communications Surveys & Tutorials*, 2021.

[3] B. Fu, N. Damer, F. Kirchbuchner, and A. Kuijper, "Sensing technology for human activity recognition: A comprehensive survey," *IEEE Access*, vol. 8, pp. 83 791–83 820, 2020.

[4] J. Lien, N. Gillian, M. E. Karagozler, P. Amihood, C. Schwesig, E. Olson, H. Raja, and I. Poupyrev, "Soli: Ubiquitous gesture sensing with millimeter waver radar," *ACM Transactions on Graphics*, vol. 35, no. 4, pp. 1–19, Jul. 2016.

[5] S. Skaria, A. Al-Hourani, M. Lech, and R. J. Evans, "Hand-gesture recognition using two-antenna Doppler radar with deep convolutional neural networks," *IEEE Sensors Journal*, vol. 19, no. 8, pp. 3041–3048, 2019.

[6] R. Zhao, X. Ma, X. Liu, and F. Li, "Continuous human motion recognition using micro-Doppler signatures in the scenario with micro motion interference," *IEEE Sensors Journal*, vol. 21, no. 4, pp. 5022–5034, 2021.

[7] P. Molchanov, S. Gupta, K. Kim, and K. Pulli, "Short-range FMCW monopulse radar for hand-gesture sensing," in *IEEE Radar Conference*, 2015, pp. 1491–1496.

[8] N. Kern, M. Steiner, R. Lorenzin, and C. Waldschmidt, "Robust Doppler-based gesture recognition with incoherent automotive radar sensor networks," *IEEE Sensors Letters*, vol. 4, no. 11, pp. 1–4, Nov. 2020.

[9] Z. Ni and B. Huang, "Open-set human identification based on Gait radar micro-Doppler signatures," *IEEE Sensors Journal*, vol. 21, no. 6, pp. 8226–8233, 2021.

[10] A. Ninos, J. Hasch, and T. Zwick, "Real-time macro gesture recognition using efficient empirical feature extraction with millimeter-wave technology," *IEEE Sensors Journal*, vol. 21, no. 13, pp. 15 161–15 170, Jul. 2021.

[11] V. Chen, F. Li, S.-S. Ho, and H. Wechsler, "Micro-Doppler effect in radar: Phenomenon, model, and simulation study," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 42, no. 1, pp. 2–21, 2006.

[12] A. A. Pramudita, Lukas, and Edwar, "Contactless hand gesture sensor based on array of CW radar for human to machine interface," *IEEE Sensors Journal*, vol. 21, no. 13, pp. 15 196–15 208, Jul. 2021.

[13] B. Zhang, L. Zhang, M. Wu, and Y. Wang, "Dynamic gesture recognition based on RF sensor and AE-LSTM neural network," in *International Symposium on Circuits and Systems*, 2021, pp. 1–5.

[14] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 1–11.

[15] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, T. Unterthiner, and X. Zhai, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2021, pp. 1–21.

[16] G. Jaswal, S. Srirangarajan, and S. D. Roy, "Range-Doppler hand gesture recognition using deep residual-3DCNN with transformer network," in *International Conference on Pattern Recognition*, 2021, pp. 759–772.

[17] G. Lai, X. Lou, and W. Ye, "Radar-based human activity recognition with 1-D dense attention network," *IEEE Geoscience and Remote Sensing Letters*, vol. 19, pp. 1–5, Jan. 2022.

[18] S. Hazra and A. Santra, "Radar gesture recognition system in presence of interference using self-attention neural network," in *IEEE International Conference on Machine Learning and Applications*, 2019, pp. 1409–1414.

[19] Y. He, X. Li, and X. Jing, "A mutiscale residual attention network for multitask learning of human activity using radar micro-Doppler signatures," *Remote Sensing*, vol. 11, no. 21, pp. 1–18, Nov. 2019.

[20] L. Zheng, J. Bai, X. Zhu, L. Huang, C. Shan, Q. Wu, and L. Zhang, "Dynamic hand gesture recognition in in-vehicle environment based on FMCW radar and transformer," *Sensors*, vol. 21, no. 19, pp. 1–20, Sep. 2021.

[21] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[23] Vadera, Sunil and Ameen, Salem, "Methods for Pruning Deep Neural Networks," *IEEE Access*, vol. 10, pp. 63 280–63 300, Jun. 2022.