IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 14, 2019, accepted August 26, 2019, date of publication August 30, 2019, date of current version September 17, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2938725

Short-Range Radar-Based Gesture Recognition System Using 3D CNN With Triplet Loss

SOUVIK HAZRA AND AVIK SANTRA, (Senior Member, IEEE)
Infineon Technologies AG, 85579 Neubiberg, Germany
Corresponding author: Avik Santra (avik.santra@infineon.com)

ABSTRACT Gesture recognition is the most intuitive form of human computer-interface. Gesture sensing can replace interfaces such as touch and clicks needed for interacting with a device. Gesture recognition in a practical scenario is an open-set classification, i.e. the recognition system should classify correct known gestures while rejecting arbitrary unknown gestures during inference. To address the issue of gesture recognition in an open set, we present, in this paper, a novel distance-metric based meta-learning approach to learn embedding features from a video of range-Doppler images generated by hand gestures at the radar receiver. Further, k-Nearest Neighbor (kNN) is used to classify known gestures, distance-thresholding is used to reject unknown gesture motions and clustering is used to add new custom gestures on-the-fly without explicit model re-training. We propose to use 3D Deep Convolutional Neural Network (3D-DCNN) architecture to learn the embedding model using distance-based triplet-loss similarity metric. We demonstrate our approach to correctly classify gestures using short-range 60-GHz compact short-range radar sensor achieving an overall classification accuracy of 94.5% over six fine-grained gestures under challenging practical environments, while rejecting other unknown gestures with 0.935 F1 score, and capable of adding new gestures on-the-fly without an explicit model re-training.

INDEX TERMS Gesture recognition, human-machine interface, mm-wave radar, triplet loss.

I. INTRODUCTION

Dynamic gestures are one of the most intuitive and effective approach for human-computer interaction. Gesture recognition has applications in wearable and mobile devices, gesture-controlled smart TVs, gesture-controlled smart homes, automotive infotainment systems, and augmented reality-virtual reality applications. Gesture sensing has also been used in sign language for communicating with hearing-impaired people [1] and controlling robots [2].

Traditionally hand gesture recognition systems have been based on optical sensors and cameras [3]. Although optical sensors have a high resolution that enables tracking and recognition of the motions of the finger and wrist, however they don't provide accurate depth estimates [4]. Camera-based hand gesture recognition can provide high accuracy applying sophisticated computer vision techniques such as hand segmentation, tracking, and classification [5], however such systems have limitations. They are limited by sufficient lighting conditions requirement, suffer from self-occlusion issues, and can introduce privacy issues.

Recently, radar-based approaches for dynamic hand gesture recognition has attracted much attention from industry and academia [4], [6]–[14]. Compared to vision-based gesture recognition systems, radar-based solutions are invariant to illumination conditions, hand visibility occlusions and additionally provides privacy-preserving features and capability to capture subtle hand gesture motions. Furthermore the processing and classification pipeline for radar can be relatively thin thus faciliating embedded implementation. At the FMCW radar receiver, a hand gesture produces a superposition of reflections from different parts of the hand with different range and velocity change over time, thus inducing a unique time-varying range-Doppler representation, which can help to detect and classify them reliably. In an FMCW radar, gesture recognition system typically consists of three steps - a) gesture detection and pre-processing by creating the range-Doppler image (RDI) while neglecting static targets and background environment, b) feature extraction, which can be accomplished through hand-crafted features such as Gabor transform etc. or implicitly through deep convolutional

The associate editor coordinating the review of this article and approving it for publication was Julien Le Kernec.

neural network (DCNN) and then c) classification of the detected gesture from a library of trained gestures using conventional machine learning approaches such as random forest, support vector machine, etc. or deep learning approaches.

In [6], the authors use a 2.4-GHz Doppler radar to identify and distinguish several gestures using hand-crafted features. Authors in [7] use a 24-GHz one transmit (Tx), two receive (Rx) FMCW radar to demonstrate gesture recognition by jointly calibrating the depth sensor. In [4], the authors propose a novel 3D deep convolutional neural network (3D-DCNN) for feature extraction, long-short term memory (LSTM) with connectionist temporal classification (CTC) loss function for the recognition of a gesture across time on camera and depth data. In [8], the authors introduce a 60-GHz FMCW radar sensor with two Tx, four Rx channels and demonstrate reliable gesture recognition using hand-crafted features and classification using random forest classifier. In [9], the authors propose novel signal processing pipeline to negate the effect of vibration facilitating gesture recognition using random forest classifier in a car using 60-GHz FMCW radar. In [10], authors propose a novel hand-crafted feature using sparsity-based approach for micro-Doppler extraction and subsequent recognition.

In [11], the authors proposed an end-to-end classification pipeline with 2D CNN with LSTM to implicitly extract features from range-Doppler images and recognize the gesture through the subsequent fully-connected layers and LSTM layer and demonstrated the superiority of the approach using 60-GHz FMCW radar. Authors in [12] use Doppler spectrogram from a Doppler radar and fed it into a DCNN to classify 8 gestures with 85.6% accuracy. In [13], the authors extend the 2D CNN-LSTM approach to use long recurrent all-convolutional network (LRACN) to improve the accuracy, memory footprint, computational complexity and improve inference time to facilitate customization into a real-time embedded platform. In [14], the authors propose to use 3D CNN-LSTM with CTC loss function using FMCW radar to improve the classification accuracy and support variable-length gestures by identifying the boundaries between gestures during inference, resulting in improved latency.

A. PROBLEM STATEMENT

Gesture sensing using radars have several major challenges to be addressed for deployment in a practical scenario - Firstly, the gesture recognition system should be able to handle large inter-class and intra-class differences of gestures. Inter-class differences refer to the variation where the same gesture is performed by another person, such as someone making the same gesture slow and another faster. Intra-class differences refer to the variation arising when the same person performs the same gesture at a different instance and under different sensor orientation.

Secondly, the gesture recognition system should be able to reject motions or unknown gestures. In a practical deployment, the system would be exposed to gestures or motions not trained and the system is expected to be able to infer these as random motions or gestures. Using conventional deep learning approaches, one approach of rejecting unknown gestures/motion is to apply a threshold on the output softmax probability. However, this is expected to work only for very few subsets of cases [15], which is not acceptable for a product-ready solution.

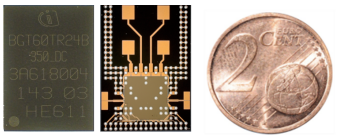
Thirdly, the system should work under all alien or unknown background environment. The gesture sensing application is also expected to work under all background environment such as in car's dashboard while perturbed with vibrations from the engine, in an unknown environment with interference from nearby walking people, background objects, etc. For gesture sensing systems to work using conventional deep learning models under all plausible background environment would require a lot of training data for each class. However, modeling or collecting training gesture data in all variable environment and background, in most cases, is practically implausible. Therefore, conventional deep learning solutions would fail under such scenarios since little or no supervised information can be made available during training.

Further, the system should be scalable and allow users to introduce gestures of their choice or convenience during life-cycle of its usage. However, conventional deep learning approaches firstly would require a lot of training examples of the new gesture to be added. Secondly, once a model is trained for a library of gestures inclusion of a new gesture for inference would require re-training the whole network or fine-tuning of last layers, which are computationally intensive and not conducive in an embedded platform without access to large compute resources.

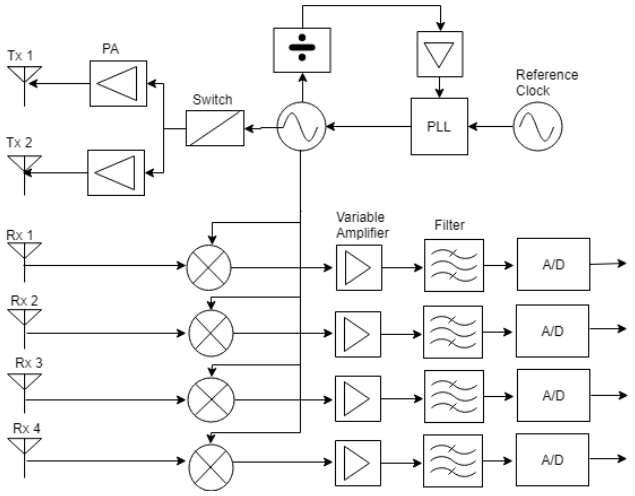
B. CONTRIBUTIONS

To address the above challenges for a scalable and product ready end-to-end gesture recognition system using FMCW radar, we propose to use meta-learning or learning-to-learn algorithms, specifically called few-shot learning [16], [17]. Meta-learning using distance-metric based models learns the relationship between data samples in the task-space and thus are capable of well adapting or generalizing to new tasks and new environments that have never been encountered during training time. To further learn appropriate feature representations and address the generalization capabilities, we exploit the triplet loss based embedding model [18]. The embedding model, thus, can simultaneously minimize the distance between similar gestures and maximize the distance between different gestures with triplet loss function metric. Such model and metrics have been successfully applied to a 2D image in literature, however, to the best of author's knowledge this is first work that extends it to 3D data, specifically video of radar RDIs.

With the well-learned gesture features embedding, k-Nearest Neighbor (kNN) algorithm is used to recognize a known gesture even under an alien environment while rejecting unknown gestures using a simple thresholding technique to minimize false alarms. The embedding network



(a) Radar chipset



(b) FMCW radar RF signal chain

FIGURE 1. (a) Infineon's BGT60TR24 60-GHz radar sensor. (b) Functional block diagram of FMCW radar signal chain.

learns a unique structure to naturally rank similarity between inputs. Once a network has been tuned, we can then capitalize on powerful discriminative features to generalize the predictive power of the network to not only new data from existing class but to entirely new classes from unknown distributions through clustering, i.e. completely new gestures, without the requirement of retraining the embedding model. Furthermore, for training, embedding models require only a few example data compared to a large amount of data typically required for conventionally deep learning approaches [19]. Using a 3D deep convolutional neural network (DCNN) architecture in conjunction with triplet loss, we can solve practical issues and also achieve strong classification performance which exceeds those of other deep learning models with near state-of-the-art deep learning performance.

The rest of the paper is organized as follows, we present the radar system design in Section II, we outline the input RDI generation in Section III, we present the proposed 3D DCNN architecture and the associated learning in Section IV, we present the system specification with details of the gesture set and implementation details in Section V, the results and discussion are presented in Section VI, and we conclude in Section VII.

II. RADAR SYSTEM DESIGN

Fig. 1a represents a highly integrated 57-64 GHz chipset of FMCW radar with four fully differential receivers and two fully differential transmitters [20] and Fig. 1b represents its functional block diagram. The RF front-end of BGT60TR24 package is integrated into an embedded wafer level ball grid array package with six integrated patch antennas realized with a metal redistribution layer. The chip includes an integrated VCO with a measured phase noise lower than -80 dBc/Hz at 100 kHz offset.

Each transmitter channel is realized using a differential cascode stage, a measured output power of 2-5 dBm over the complete frequency range. The output power of the transmitter is controllable by fine current control using a 6 bit DAC, which provides a reference voltage to the cascode stage. The chip's integrated VCO has a low impedance bias network to minimize phase noise contribution. The programmable frequency divider is driven by the fundamental signal. A programmable frequency divider with two-division ratios is included in the chipset to enable the use of both hardware and software phase-locked loop (PLL) systems.

The frequency of the FMCW waveform with bandwidth B and chirp duration T can be expressed as

$$f_T(t) = f_c + \frac{B}{T}t \tag{1}$$

where f_c is the ramp start frequency. The reflected signal from the target is mixed with a replica of the transmitted signal resulting in beat signal. The phase of the beat signal after the mixer due to k^{th} point target is

$$\phi_k(t) = 2\pi\left(f_c\tau_k + \frac{B}{T}t\tau_k - \frac{B}{2T}\tau_k^2\right) \tag{2}$$

where $\tau_k = \frac{2(R_k+v_kt)}{c}$ is the round trip propagation delay between the transmitted and received signal after reflection from the k^{th} target with range R_k and radial velocity v_k. The down-converted Intermediate Frequency (IF) signal therefore is the super-position of received signal from K point scatterers and thus expressed as

$$s_{IF}(t) = \sum_{k=1}^{K} \exp\left(2\pi\left(\frac{2f_cR_k}{c} + \left(\frac{2f_cv_k}{c} + \frac{2BR_k}{cT}\right)t\right)\right) \tag{3}$$

after ignoring the second-order terms $\frac{2Bv_k}{Tc}t^2$.

Each of the receiver channels consists of a double-balanced mixer and an intermediate frequency (IF) buffer amplifier. The IF bandwidth is expected to be between 10 kHz to 1 MHz, and thus the IF stage analog filter is set accordingly. A low noise amplifier is not present to increase the overall linearity of the receiver stage. An active RF distribution network is used to feed the single-ended local oscillator signal to all receive channels. This beat signal is low-pass filtered, which is next sampled by the 12-bit analog to digital converter (ADC).

The propagation delay is translated to beat frequency, which can be identified by spectral analysis (eg. FFT) and the Doppler manifests as a frequency over the slow time, which refers to the time index within pulses in a frame or coherent processing interval. Conventionally the angle is estimated by analyzing the phase between receive channels at a specific range and Doppler.

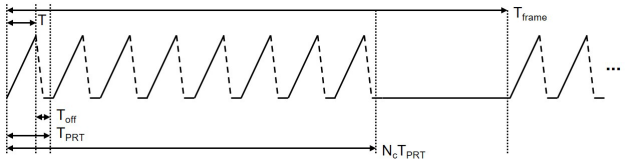


FIGURE 2. Saw-tooth ramps within a frame with timing defintions.

III. INPUT RANGE-DOPPLER IMAGES

The frequency shifts due to range and velocity arising from multiple point targets at the IF signal are decoupled by generating RDI across all virtual channels. The blob pixel intensity on the RDI represents the collective reflected energy from all scatterers at the same range and Doppler and is captured in the RDI. A gesture generates a unique time-varying RDI at the radar receiver over time sequences representing the artifacts of the hand movement to produce the gesture.

Expanding the time index t as $n_k T_{\text{frame}} + n_s T_{\text{PRT}} + n_f$, where n_f is the fast time index $0 < n_f < T$. n_s denotes the slow time index and T_{PRT} is the chirp repetition time indicating the time difference between the start of two consecutive chirps in a frame. n_k denotes the frame number, where the indexing starts when a gesture is detected and T_{frame} is the total frame as denoted in Fig. 2. The received signal at frame n_k, $s_{IF}(t; n_k)$, from consecutive chirps are arranged in the form of a 2D matrix, i.e. $s_{IF}(n_s, n_f; n_k)$. The RDI is generated for each channel by applying window function, zero-padding and then 1D fast Fourier transform (FFT) along fast time to obtain the range transformation, followed by applying window function, zero-padding and then 1D FFT along slow time index. Subsequently, the moving average filter is employed to subtract the background over the RDI from the current frame. Thus the RDIs over frames captures the dynamics of the gesture over the temporal domain.

The two 1D FFT transforms the signal $s_{IF}(n_s, n_f; n_k)$, along fast time and slow time, into range-Doppler domain

$$S(p, q, n_k) = \sum_{n_s=1}^{Z_{Nc}} \left(\sum_{n_f=1}^{Z_{NTS}} w_f(n_f) s_{IF}(n_s, n_f; n_k) e^{-j2\pi p n_f / Z_{\text{NTS}}} \right) . w_s(n_s) e^{-j2\pi q n_s / Z_{Nc}} \quad (4)$$

with NTS and Z_{NTS} being number of transmitted samples defined by DAC sampling points over chirp duration and zero-padding along fast-time respectively. Nc and Z_{Nc} being the number of chirps in a frame and zero-padding along the slow-time respectively. $w_f(n_f)$ and $w_s(n_s)$ represents the window function along fast-time and slow-time respectively, for our implementation we have used the Hamming window and Kaiser window respectively. p, q denotes the index over range and Doppler respectively. It is obvious that the peaks in the range-Doppler domain occur at

$$p_k = \left(\frac{2f_c}{c} v_k + \frac{2B}{cT} R_k \right)$$
$$q_k = \frac{2 v_k f_c}{c} \quad (5)$$

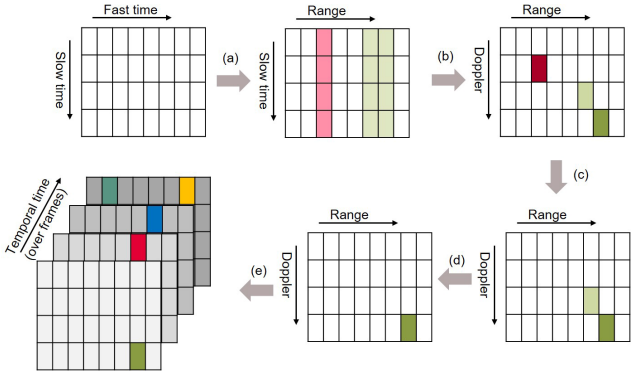


FIGURE 3. RDI generation by (a) 1D FFT along fast time, (b) 1D FFT along slow time, (c) background subtraction, (d) thresholding, (e) detection and collection of RDI over frames.

Using saw-tooth FMCW with fast ramps, $\frac{2f_c}{c} v_k << \frac{2B}{cT} R_k$, thus the range peaks appear at $\frac{2B}{cT} R_k$.

Fig. 2 depicts the sawtooth FMCW ramps within a frame along with timing consideration and the definitions thereof. The ramps within a frame are used to generate a processed RDI and across frames to generate the video of RDIs. T_{off} denotes the chirp preparation time and includes the ADC settling time, power amplifier turn on-time and PLL setup time. $N_c T_{\text{PRT}}$ denotes the coherent pulse integration time and defines the Doppler resolution. T_{frame} defines the refresh or update rate of the FMCW system and is limited by the RDI processing and classification time. The maximum velocity is given as $v_{\max} = \frac{c}{2 f_c T_{\text{PRT}}}$ and the minimum velocity is $\delta v = \frac{c}{2 f_c Z_{Nc} T_{\text{PRT}}}$.

Fig. 3 shows the signal processing steps to create the processed RDIs. The 1D FFT along fast time transforms the data to range domain, wheras the 1D FFT along slow time transform the other dimension to Doppler domain. Following the 2D FFT to transform the data into range-Doppler domain, background subtraction is achieved through moving average filter as

$$S(p, q; n_k) = S(p, q; n_k) - S_B(p, q; n_k)$$
$$S_B(p, q; n_k + 1) = \gamma S_B(p, q; n_k) + (1 - \gamma) S(p, q; n_k) \quad (6)$$

where $S(p, q; n_k)$ is the RDI at n_k^{th} frame, and $S_B(p, q; n_k)$ is the background RDI at the n_k^{th} frame. γ is the moving average coefficient and is set to 0.7 for our setup. Following the background subtraction of RDI, thresholding operation is performed to limit the values below pre-defined fixed threshold to zero so that they do not influence the subsequent neural training process, the thresholding operation also limits the range to $R_{\max} = 0.5$m to reduce interferences from farther distance targets.

For gesture detection, the effective energy of the background subtracted RDI is calculated and once it crosses a pre-defined threshold, a gesture is said to be detected and this initalizes the RDI recording for subsequent 100 frames, i.e. 2 s, which is then fed to the classification pipeline for recognition of the gesture or rejecting the gesture motion as false alarm.

IV. ARCHITECTURE AND LEARNING

The objective of the proposed 3D DCNN is to achieve an embedding $f(g)$ for a given gesture g (video of RDI) into a feature space R^d so that the squared Euclidean distance between same gestures is small and that of two different gesture is large, independent of the intrinsic properties of the radar. The model not only learn the class of g but also how different it is from other gestures g'. In terms of objective function this can be achieved through triplet loss distance-metric by training three 3D-DCNN sharing same weights feeding an anchor example, positive example (i.e. same gesture), negative example (i.e. different gestures). The triplet loss tries to form a margin between different gestures in the embedding space, thus allowing embeddings of the same gestures to exist in a close-knit cluster and distanced away from other gesture clusters.

A. 3D - DEEP CONVOLUTION NEURAL NETWORK ARCHITECTURE

Generally, 2D DCNNs are used to apply convolution on 2D feature images to extract features in the spatial dimension. However, in case of video of RDIs, temporal information that is present over a stream of input frames needs to be captured to make a sequential data classification. So, we employ 3D convolutions in our network to capture spatial and temporal information. In 3D convolution, the input is a 3D cube where the third dimension denotes the temporal dimension containing sequence of frames and is formed by stacking 2D frames sequentially.

1) 3D Convolution Layer - The convolution in this case is achieved by convolving a 3D kernel over the data cube. In i^{th} layer, the value at position (x, y, z) in j^{th} feature images is given as

$$v_{ij}^{xyz} = \tanh\left(b_{ij} + \sum_{m}\sum_{p=0}^{P_i-1}\sum_{q=0}^{Q_i-1}\sum_{r=0}^{R_i-1} w_{ijm}^{pqr} v_{(i-1)m}^{(x+p)(y+q)(z+r)}\right) \tag{7}$$

where R_i is the size of the 3D kernel along the temporal dimension, w_{ijm}^{pqr} is the $(p, q, r)^{\text{th}}$ value of the kernel connected to the m^{th} feature image in the previous layer. Multiple such filter kernels are required at each layer to extract diverse information.

2) 3D Max Pooling - The objective of employing the 3D max pooling layer is to achieve lower-dimensional feature images with retention of most relevant information. Pooling helps to make the representation invariant to small translations, increase global receptive field and helps to counter the overfitting of the model.

3) Activation Layer - The activation layer introduces non-linear transformation on the input signal by learning whether a neuron will fire or not is known as the activation function. Rectified Linear unit (ReLu) is one of the most widely used activation function. It facilitates faster backpropagation and doesn't activate all neurons thus is computationally very efficient.

4) Dropout - It is a regularization approach that reduces inter-dependent learning among a set of neurons and thus preventing overfitting of training data.

5) Dense Layer - The neurons in this layer have a complete connection to the high-level features extracted in the previous layers and their activation is computed by matrix multiplication and then a bias offset.

B. TRIPLET LOSS

The embedding model embeds a RDI sequence g into a d-dimensional embedding space. The triplet loss used for 3D data-cube is inspired by FaceNet [18] proposed in the context of 2D image face classifciation. Given the triplets (g^p, g^a, g^n) where p, a, n represents positive, anchor and negative examples respectively, we want to ensure that the anchor of a given gesture sequence is closer to all other sequences of same gesture g^p than sequences of other gestures g^n and maintain a defined margin (α) between different gesture sequences. This requires the fulfillment of the following constraint

$$\|f(g_i^a) - f(g_i^p)\|_2^2 + \alpha \;<\; \|f(g_i^a) - f(g_i^n)\|_2^2 \tag{8}$$

$\forall f(g_i^a), f(g_i^p), f(g_i^n) \in T$ and the loss function can be defined as the following

$$\sum_{i=1}^{N} [\|f(g_i^a) - f(g_i^p)\|_2^2 - \|f(g_i^a) - f(g_i^n)\|_2^2 + \alpha]_+ \tag{9}$$

where $[x]_+$ represents $\max(0, x)$. Using all the possible triplets at every epoch would result in many triplets that satisfy the constraints after a few epochs. These easy triplets would not help in training and rather slow down the convergence as they will be passed through the network. Therefore, it is desirable to select hard triplets that would contribute to the training process and lead to faster convergence. The following sub-section discusses the adapted triplet selection model.

C. TRIPLET SELECTION

Triplet selection or triplet mining is a critical aspect of training a DCNN using triplet loss, if not done correctly the loss can get stuck in local minima after reducing drastically in the first few epochs. We adapt offline triplet selection, where after every 100 epochs we save a checkpoint and select triplets that are semi-hard negatives. Semi-hard negatives are the ones where the positive anchor distance is smaller than that of negative anchor but the negative anchor distance is close to the to positive anchor distance and the negatives exist within the margin. We do not select the hard negatives as they can lead to early local minima and poor training.

D. WEIGHT INITIALIZATION

The weight initialization for 3D convolutional layers was performed by drawing samples from a normal distribution with zero-mean and standard deviation of 10^{-2}. The respective biases were initialized with samples drawn from a normal distribution also but with a mean of 0.5 instead of zero.

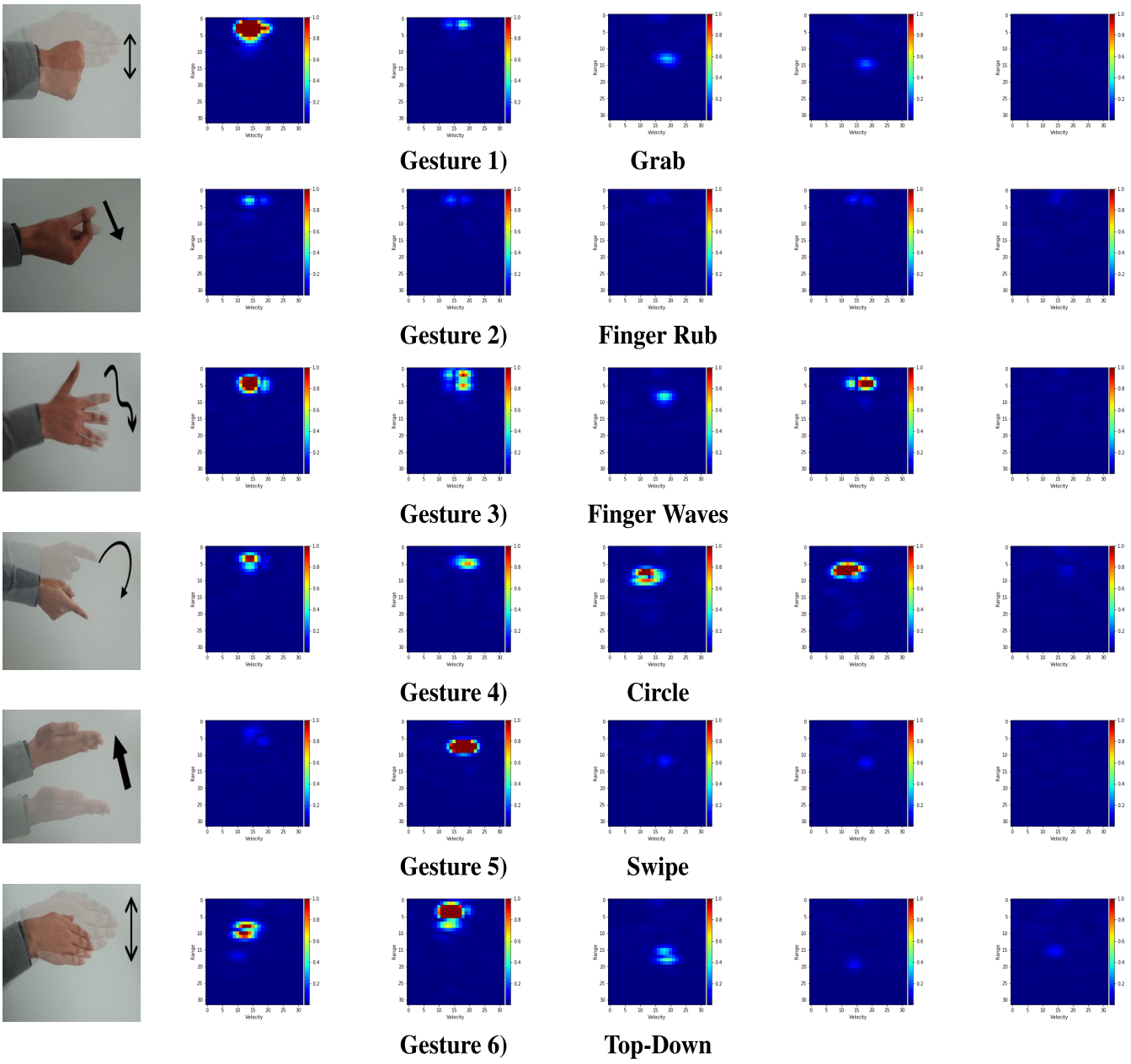


FIGURE 4. Gesture set and its corresponding video of RDI at time stamps 0.1s, 0.3s, 0.9s, 1.3s, 1.5s from detection of the gesture for Rx antenna one for 1) grab, 2) finger rub, 3) finger waves, 4) circle, 5) swipe, and 6) top-down gestures respectively. In the x-axis, the negative velocity is represented by 0 to 15 bins and 16 to 31 bins represents positive velocity, whereas the y-axis represents increasing range bins from top to bottom.

The weights for the dense layers were initialized with Xavier uniform initializer, which draws samples from a uniform distribution within [-limit, limit] where the limit is calculated by taking the square root of 6 divided by the total number of input and output units in the weight tensor.

E. LEARNING SCHEDULE

Adaptive moment estimation (Adam) [21] optimizer is used to compute adaptive learning rate for each network weight over the learning process from estimates of first and second moments of the gradients. In configuration parameters, the learning rate (alpha) is set to 0.001 and the exponential decay rate for the first (beta1) and second (beta2) moment estimates are set to 0.9 and 0.999. The epsilon that counters divide by zero problem is set to 1e-8.

F. DATA AUGMENTATION

The RDI are augmented to increase the dataset and achieve a broader generalization of the model. In our data augmentation technique, we create synthetic images with a variance to the original RDIs. At first, a mean RDI across all channels for each gesture class is created and for each time. Following that,

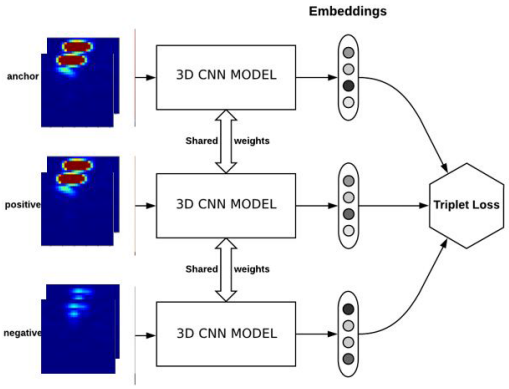


FIGURE 5. Principle of triplet loss based on proposed 3D DCNN.

we generate values for each of our synthetic records in each time step by drawing values from a normal distribution with mean equal to the corresponding original RDI at that time step and variance drawn from gesture class variation. This helps to model the time variations of a gesture performed by different individuals.

V. SYSTEM SPECIFICATIONS

A. GESTURE SET

After a rigorous literature review on the use of different hand gestures in human-computer interaction, we have defined a gesture set for training purpose. All the chosen 6 gestures involve the movement of fingers or minimum muscle movement for hand displacement and are thus dynamic and micro. Gestures like finger rub and rotation allow us to exploit the intrinsic property of the radar which unlike camera doesn't suffer from self-occlusion and can recognize very small motion. The six selected gestures, whose RDI over time are illustrated in Fig. 4, are

1) Grab (moving a hand towards the sensor, perform a grab action and move it away),
2) Finger Rub (displacement of thumb placed on the index finger),
3) Finger Waves (movement of all the fingers above the sensor like playing a piano),
4) Circle (circular movement of a finger above the sensor),
5) Swipe (moving the hand horizontally from right to left),
6) Top-Down (movement of the palm toward the sensor and away like pushing a button).

B. SYSTEM PARAMETERS

The sweep bandwidth of 7 GHz is used, hence the theoretical range resolution of our system is $\delta r = c/2B = 2.14$ cm, where c is the speed of light. The fast-time window function results in some loss of range resolution. The number of transmit DAC samples used for generating the chirp is $NTS = 64$, hence the maximum detectable range is $R_{\max} = (NTS/2) \cdot \delta r = 0.68$ m, since the BGT60TR24 sensor has only I channel. The ADC sampling frequency is set to 2 MHz, the maximum beat frequency is given by $\frac{2BR_{\max}}{cT} = 495.83$ kHz.

TABLE 1. 3D Proposed DCNN architecture used for the embedding model.

Layer (Type)	Output Shape	Parameters
conv3d_1 (Conv3D)	(None, 96,28,28,32)	16032
relu_1 (Activation)	(None, 96, 28, 28, 32)	0
max_pooling3d_1 (MaxPooling3D)	(None, 48, 14, 14, 32)	0
dropout_1 (Dropout)	(None, 48, 14, 14, 32)	0
conv3d_2 (Conv3D)	(None, 46,12,12,64)	55360
relu_2 (Activation)	(None, 46, 12, 12, 64)	0
dropout_2 (Dropout)	(None, 46, 12, 12, 64)	0
conv3d_3 (Conv3D)	(None, 44,10,10,64)	110656
relu_3 (Activation)	(None, 44, 10, 10, 64)	0
dropout_3 (Dropout)	(None, 44, 10, 10, 64)	0
max_pooling3d_2 (MaxPooling3D)	(None, 22, 5, 5, 64)	0
conv3d_4 (Conv3D)	(None, 18,1,1,64)	512064
relu_4 (Activation)	(None, 18, 1, 1, 64)	0
dropout_4 (Dropout)	(None, 18, 1, 1, 64)	0
conv3d_5 (Conv3D)	(None, 18,1,1,128)	8320
relu_5 (Activation)	(None, 18, 1, 1, 128)	0
dropout_5 (Dropout)	(None, 18, 1, 1, 128)	0
flatten_1 (Flatten)	(None, 2304)	0
dense_1 (Dense)	(None, 32)	73760
sigmoid_1 (Activation)	(None, 32)	0
Total Parameters		776,192

The chirp time is set to 32 μs and the chirp repetition time is set to 64 μs and we use 16 consecutive chirps in a frame, thus the maximum velocity is given as $v_{\max} = 39.0625$ m/s and the minimum velocity is $\delta v = 2.44$ m/s. The frame time is set to 20 ms, and 100 consecutive frames, i.e. 2 s of data is recorded once a gesture is detected. We use one Tx antenna and four Rx antennas for reception, and the 3dB azimuth and elevation Field of View (FoV) are both 70°.

C. IMPLEMENTATION DETAILS

For each gesture, 100 consecutive RDIs are stacked together and fed to the network as input shape of (100, 32, 32, 4). Triplet mining is performed and the (g^a, g^p, g^n) set is fed to the three identical 3D networks that share the same weights as depicted in Fig. 5.

Instead of linear activation, sigmoid activation is used at the last fully connected layer to prevent loss of information by yielding positive values. Different 3D architectures based on the trade-off of accuracy and model size have been explored. Our proposed solution, as depicted in Tab. 1, is aimed to run in small consumer electronics like mobile phones and wearable devices. Although, having deeper networks yielded slightly better results but increased the number of parameters and FLOPs exponentially. For our use case, the proposed

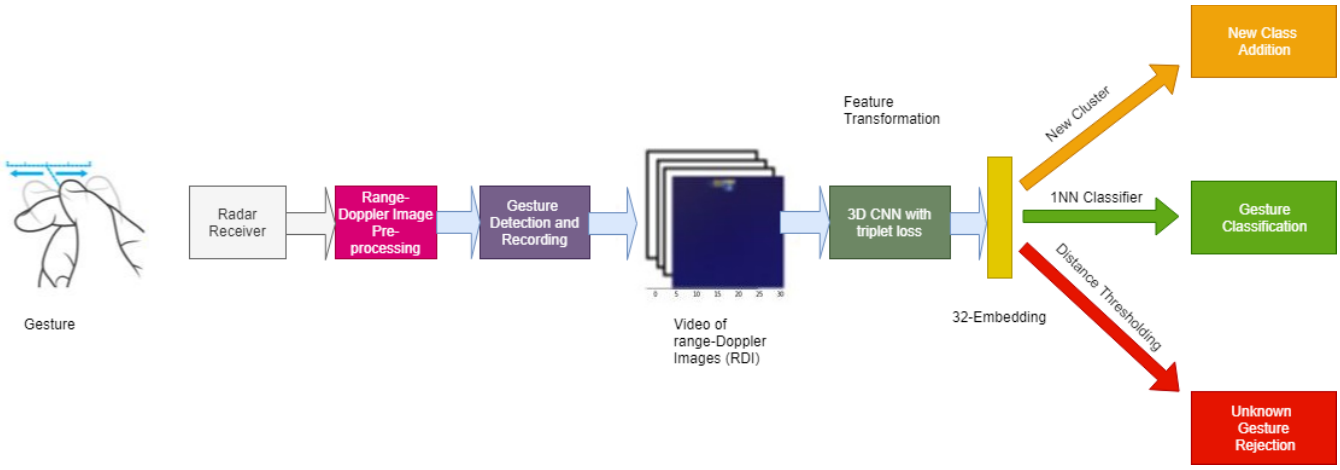


FIGURE 6. Overall gesture sensing pipeline depicting feature transformation through DCNN embedding model to enable gesture classification, rejection of unknown gestures and new gesture class addition.

model seems to be optimal with a model size of 7.96 MB and 3.1 MFLOPs (without quantization/fusion/weight-pruning optimization). Once the embedding model is trained, a 1-nearest neighbor (NN) classifier is trained to make gesture class prediction taking generated embedding of the gestures.

Typically, training deep neural networks require a large amount of data however, deep neural network using triplet loss can be trained using much less amount of data owing to the large number of possible triplet combinations. For each gesture class, 150 sequences were recorded where each sequence contains 100 frames making each gesture couple of seconds long, performed by 10 individuals with minimal prior instructions. The dataset were collected under different challenging environment, particularly gestures were recorded in a crowded background. Other environments include wherein the chip was hand-held and gesture was performed with the other hand, also wherein the chip was placed in the car's infotainment dashboard while the engine was kept on and co-passenger made random movements. The training dataset size was augmented by techniques mentioned earlier resulting in a total training size of 1800 (300 sequences for each class). A testing dataset of 420 sequences (70 sequences for each class) were recorded by 5 other individuals performing the same gestures under the same background environment as training dataset but 20 % of data was collected under different background and sensor orientation which wasn't present in the training dataset. A change in RDI energy is used to detect the start of a gesture and then a video of RDI is transferred into the 3D CNN to generate the embedding which then goes into 1-NN classifier where its either rejected as a false alarm or gesture prediction is made. The end-to-end proposed radar-based gesture recognition system is depicted in Fig. 6.

VI. RESULTS AND DISCUSSION

The overall end-to-end gesture sensing pipeline is evaluated by the following metrics -

1) t-SNE Representation [22] - We use t-SNE for visualizing the generated embeddings in a 2-dimensional space, which gives a visually intuitive understanding about how efficiently the model clusters similar gestures together. t-SNE is a well suited non-linear technique to visualize higher dimensional data by performing dimensionality reduction. Initially, the algorithm computes the similarity probability of data points in input space and targeted lower-dimensional space. Next, the algorithm tries to minimize the conditional probabilities (or similarities) in both the spaces for a perfect lower-dimensional representation. The sum of Kullback-Leiber divergence of all the data points is minimized using gradient descent technique to measure the minimization of the sum of difference of conditional probability. One must note that after this process, the data points are no longer retrievable and hence the output of t-SNE can only be used for visualization and exploration purposes.
2) Accuracy Metrics - Confusion matrices are computed to evaluate the recognition accuracy of our proposed end-to-end architecture. Further, the F1 score is computed by calculating the precision and recall to evaluate the capability of our proposed system to detect and reject false alarms.

A. GESTURE CLASSIFICATION

First, the gesture classification capability of our proposed architecture is evaluated for known classes over the test data. The accuracy of the architecture is computed with yet another test set containing 180 sequences (30 each class) recorded with different background noise to evaluate the generalization capability of our embedding model. As seen in the confusion matrix depicted in Table 2, we see that our proposed model has high accuracy over all the classes and yields an overall accuracy of 94.5 %. Furthermore, an interesting takeaway from the following result is how efficiently and accurately

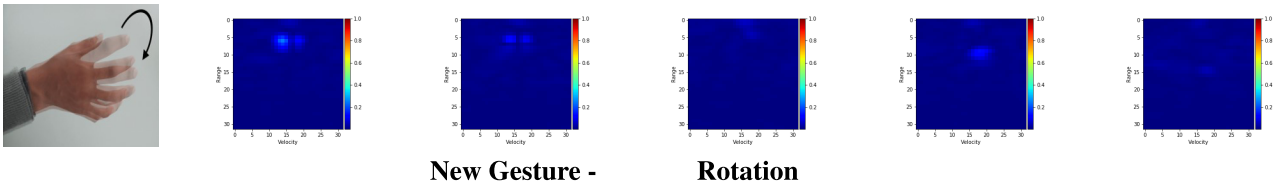


FIGURE 7. New rotation gesture and its corresponding video of RDI at time stamps 0.1s, 0.3s, 0.9s, 1.3s, 1.5s from detection of the new custom rotation gesture for receive antenna 1.

TABLE 2. Confusion matrix of the test set containing 600 examples from all classes. The overall accuracy is 94.5 %.

Actual Class \ Predicted Class	a	b	c	d	e	f
a	90	0	0	0	10	0
b	0	100	0	0	0	0
c	0	3	92	0	5	0
d	0	0	0	92	8	0
e	0	3	0	0	97	0
f	0	0	0	1	3	96

TABLE 3. Comparison of different approaches for gesture classification under alien environment.

Approach	Description	Accuracy
3D CNN	Categorical cross entropy loss function	86.3 %
2D CNN-LSTM	CTC loss function	88.1 %
3D CNN - kNN	**Triplet loss embedding**	94.5 %

it can differentiate between very similar gestures like grab (gesture d) and top-down (gesture f). The result also shows that the embeddings generated by the proposed model is invariant to background noise to a great extend.

Further, Table 3 presents the classification accuracy of the proposed approach in comparison to known 3D CNN and 2D CNN-LSTM approaches. Since the test dataset contains different background noise not present in training dataset, the proposed approach outperforms both 2D CNN-LSTM model and 3D CNN model without any significant increment in model size. This behavior is well expected and can be explained because despite having a very small training dataset, a huge number of triplet combinations are generated which allows the model to learn similarity features. This also allows high generalization capability of the model and is an intrinsic property of the proposed approach.

B. NEW CLASS ADDITION

Second, to evaluate the scalability of our proposed embedding model, the trained embedding model is used for generating few embeddings of the new class (not present in the initial training set). The trained embedding model projects the new gesture class in a unique cluster in the embedding space. The 1-NN classifier is then updated with the new class

TABLE 4. Confusion matrix of the test set containing 700 examples from all classes when a new gesture is added during inference. The overall accuracy is 94.57 %.

Actual Class \ Predicted Class	a	b	c	d	e	f	g
a	90	0	0	0	10	0	0
b	0	97	0	0	0	0	3
c	0	3	92	0	5	0	0
d	0	0	0	92	8	0	0
e	0	3	0	0	97	0	0
f	0	0	0	1	3	96	0
g	0	0	0	2	0	0	98

embeddings. This process allows adding new class with a very few number of samples (in our case we have used 15 samples) without the requirement to retrain the original deep learning model, which makes it a fast and computationally inexpensive approach towards expanding to include new gestures.

The existing test data used in the previous experiment along with 100 new rotation gestures (moving the hand like grabbing and rotating a knob as shown in Fig. 7) records were added, of which again 30 of them were recorded with unseen background not present in the training dataset. Table 4 presents the confusion matrix generated for the (6+1) classes which show high accuracy for all gesture classes like in the previous experiment, including the new gesture. We achieve an overall accuracy of 94.57% and is interesting to see how accurately the proposed model can recognize the new gesture rotation (gesture g), which was absent in the training phase of the embedding model without confusing it with a much similar gesture such as finger waves (gesture c) present in the training set.

We use t-SNE to visualize the embeddings, in Fig. 8, generated by our proposed model for a small subset of test data containing randomly chosen 10 gestures for each gesture class including the new gesture class (rotation gesture). As we can see in the figure, each gesture class forms its own cluster as expected. However, the rotation gesture class and the finger wave gesture class clusters are very close to each other with an outlier in each, since both the gestures are very similar with the involvement of all the five fingers. These observations confirm the unique embedding capability of our proposed

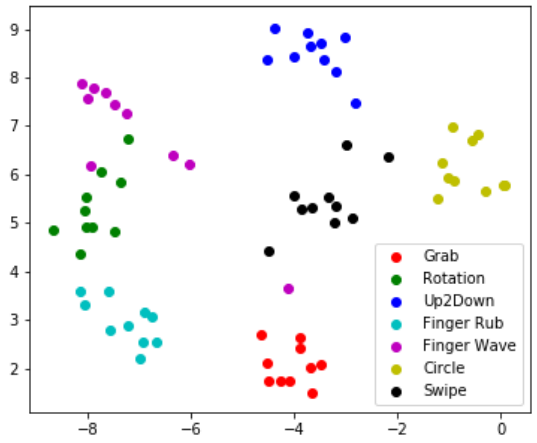


FIGURE 8. t-SNE representation of gestures embedding using proposed 3D-DCNN using triplet loss.

TABLE 5. Confusion matrix of the test set containing 200 examples from all classes. The F1 score is 0.935.

		Predicted	
		Invalid	Valid
Actual	Invalid	94	6
	Valid	7	93

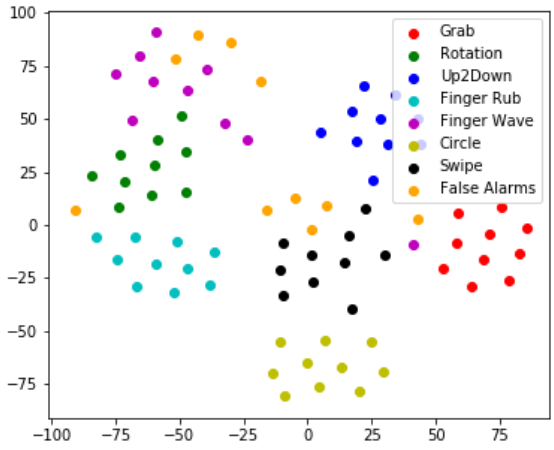


FIGURE 9. t-SNE representation of gestures embedding with invalid gesture using proposed 3D DCNN.

model and the class expandable feature of the proposed architecture.

C. UNKNOWN GESTURE REJECTION

To detect unknown or invalid gesture/motion, the normalized l_2 distance (d) between the embedding of the input gesture sequence and the nearest neighbor embedding is computed and if the value is more than a defined threshold ($d_{\max}$), it is tagged as a false alarm and thus 1-NN classification is not performed.

The value of $d_{\max}$ can be tweaked according to the requirements, where setting a very low threshold would result in very strict acceptance of performed gestures and very high threshold would result in a very relaxed acceptance of the gestures. A relatively low threshold value was selected, which is optimal in our experiment and yields a precision of 0.93, recall of 0.94 and a F1 score of 0.935 computed from Table 5.

We also visualize the embeddings, in Fig. 9, of the same subset along with 10 randomly chosen gesture data of invalid gestures. As expected, none of the invalid gestures (false alarms) are present inside any of the valid gesture clusters thus demonstrating the invalid gesture or motion rejection capability of the system.

VII. CONCLUSION

Gesture recognition is more intuitive than mouse, touch, keyboard or joystick-based human-machine interaction. In this paper, we propose a novel architecture using embedding model for hand gesture recognition, which is capable of providing high classification accuracy while rejecting unknown gestures or motion with a low memory footprint for real-time interaction with a machine using 60-GHz mm-wave short-range radar. Our proposed method generalizes well to alien environments and background noise and is demonstrated experimentally in the paper. The proposed system is also scalable for adding new custom gesture with requirements of few supervised examples and without the need of elaborate model re-training. Future work can explore the training of different deep learning architectures, such as 2D CNN-LSTM embedding model, for smaller memory footprint and lower latency during inference.

REFERENCES

[1] J. R. Pansare, S. H. Gawande, and M. Ingle, "Real-time static hand gesture recognition for American Sign Language (ASL) in complex background," *J. Signal Inf. Process.*, vol. 3, no. 3, p. 364, Aug. 2012.

[2] A. Malima, E. Özgür, and M. Çetin, "A fast algorithm for vision-based hand gesture recognition for robot control," in *Proc. IEEE 14th Signal Process. Commun. Appl.*, Apr. 2006, pp. 1–4.

[3] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 37, no. 3, pp. 311–324, May 2007.

[4] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, " Online detection and classification of dynamic hand gestures with recurrent 3D convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4207–4215.

[5] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: A survey," *Artif. Intell. Rev.*, vol. 43, no. 1, pp. 1–54, Jan. 2012.

[6] Q. Wan, Y. Li, C. Li, and R. Pal, "Gesture recognition for smart home applications using portable radar sensors," in *Proc. 36th Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2014, pp. 6414–6417.

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

[8] J. Lien, N. Gillian, M. E. Karagozler, P. Amihood, C. Schwesig, E. Olson, H. Raja, and I. Poupyrev, "Soli: Ubiquitous gesture sensing with millimeter wave radar," *ACM Trans. Graph.*, vol. 35, no. 4, Jul. 2016, Art. no. 142.

[9] K. A. Smith, C. Csech, D. Murdoch, and G. Shaker, "Gesture recognition using mm-wave sensor for human-car interface," *IEEE Sens. Lett.*, vol. 2, no. 2, Jun. 2018, Art. no. 3500904.

[10] G. Li, R. Zhang, M. Ritchie, and H. Griffiths, "Sparsity-driven micro-Doppler feature extraction for dynamic hand gesture recognition," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 2, pp. 655–665, Apr. 2018.

[11] S. Wang, J. Song, J. Lien, I. Poupyrev, and O. Hilliges, "Interacting with soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum," in *Proc. 29th Annu. Symp. User Interface Softw. Technol.*, Oct. 2016, pp. 851–860.
[12] Y. Kim and B. Toomajian, "Hand gesture recognition using micro-Doppler signatures with convolutional neural network," *IEEE Access*, vol. 4, pp. 7125–7130, 2016.
[13] S. Hazra and A. Santra, "Robust gesture recognition using millimetric-wave radar system," *IEEE Sens. Lett.*, vol. 2, no. 4, Dec. 2018, Art. no. 7001804.
[14] Z. Zhang, Z. Tian, and M. Zhou, "Latern: Dynamic continuous hand gesture recognition using FMCW radar sensor," *IEEE Sensors J.*, vol. 18, no. 8, pp. 3278–3289, Feb. 2018.
[15] A. Bendale and T. E. Boult, "Towards open set deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1563–1572.
[16] *Meta-Learning: Learning to Learn Fast*. Accessed: May 10, 2019. [Online]. Available: https://lilianweng.github.io/lil-log/2018/11/30/meta-learning.html
[17] B. M. Lake, R. Salakhutdinov, and J. B. Tenebaum, "Human-level concept learning through probabilistic program induction," *Science*, vol. 350, no. 6266, pp. 1332–1338, 2015.
[18] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 815–823.
[19] A. Santra, I. Nasr, and J. Kim, "Reinventing radar: The power of 4D sensing," *Microw. J.*, vol. 61, no. 12, pp. 26–38, Dec. 2018.
[20] I. Nasr, R. Jungmaier, A. Baheti, D. Noppeney, J. S. Bal, M. Wojnowski, E. Karagozler, H. Raja, J. Lien, I. Poupyrev, and S. Trotta, "A highly integrated 60 GHz 6-channel transceiver with antenna in package for smart sensing and short-range communications," *IEEE J. Solid-State Circuits*, vol. 51, no. 9, pp. 2066–2076, Sep. 2016.
[21] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," Dec. 2014, *arXiv:1412.6980*. [Online]. Available: https://arxiv.org/abs/1412.6980
[22] L. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

SOUVIK HAZRA received the B.E. degree from KIIT University. He is currently pursuing the M.S. degree in data science and engineering with EURECOM. He performed his bachelor's thesis on Application of Deep Learning at Airbus Group, India. He was a Research Intern with CCAF, University of Cambridge, where he was involved in deep learning and big data topics. He is also a Master Thesis Student with a focus on developing machine learning and deep learning algorithms for industrial and consumer radars at Infineon Technologies AG, Neubiberg, Germany.

AVIK SANTRA (S'09–M'10–SM'18) received the B.E. degree from the West Bengal University of Technology, in 2008, and the M.E. degree (Hons.) in signal processing from the Indian Institute of Science, Bengaluru, India, in 2010. He was a System Engineer for LTE/4G modem chipsets at Broadcom Communications and was also a Research Engineer developing fully adaptive phased array and MIMO radars at Airbus Group. He is currently an Algorithm Developer developing signal processing and machine learning algorithms for industrial and consumer radars and depth sensors at Infineon Technologies AG, Neubiberg, Germany. He has filed over 40 patents and has published 15 research articles related to various topics of radar waveform design, radar signal processing, and radar machine/deep learning topics. He was a recipient of several outstanding reviewer awards. He is a Reviewer for various IEEE and Elsevier journals.