HUMAN MOTION CLASSIFICATION WITH MICRO-DOPPLER RADAR AND BAYESIAN-OPTIMIZED CONVOLUTIONAL NEURAL NETWORKS

Hoang Thanh Le[1]*, Son Lam Phung*[1]*, Abdesselam Bouzerdoum*[1,2]*, and Fok Hing Chi Tivive*[1]

[1]School of Electrical, Computer and Telecommunications Engineering
University of Wollongong, NSW, Australia
[2]College of Science and Engineering, Hamad Bin Khalifa University, Doha, Qatar

ABSTRACT

In recent years, Doppler radar has emerged as an alternative sensing modality for human gait classification since it measures not only the target speed, but also the local dynamics of the moving body parts, thereby creating a unique spectral signature. This paper presents a learning-based method for classifying human motions from micro-Doppler signals. Inspired by the applications of deep learning, the proposed method extracts features from the time-frequency representation of the radar signal using a cascaded of convolutional network layers. To design a optimal network architecture, the Bayesian optimization with Gaussian process priors is employed. Experimental results on real data are presented, which show a significant improvement compared to three existing approaches.

Index Terms— micro-Doppler radar, time-frequency representation, convolutional neural network (CNN), Bayesian optimization

1. INTRODUCTION

Recent years have witnessed a high demand of using modern radar systems for surveillance, tracking, and imaging applications, in both civilian and military contexts. In contrast to sensors, such as digital cameras and infra-red sensors, radars can work in various lighting environments and from different standoff distances. Moreover, radar systems are less intrusive because they do not capture the face or other identifiable visual properties. Apart from measuring the speed and locating the object, Doppler radars can sense the frequency modulations on the radar return, which are induced by the micro-movements of the moving object, e.g., the rotation of the wheel of a vehicle or the arm swing of a walking person. These frequency modulations and the main Doppler frequency create a micro-Doppler (μ-D) signature, which can be used for motion classification.

Over the past two decades, several studies have been conducted to analyze the μ-D signatures of different moving targets, such as vehicles [1], jet engines [2], ballistic targets [3], and human gait [4]. In recent years, the research focus has been diverted to the development of classification approaches for Doppler radars. For non-rigid moving objects, several techniques have been proposed, based on image classification, where the radar signal is converted into a time-frequency representation, from which features are then extracted and classified. Kim and Ling defined six types of features from the spectrogram for classifying human activities [5]. Bjorklund *et al.* exploited the periodicity of the μ-D frequencies by converting the spectrogram into the cadence velocity diagram (CVD) [6]. Other researchers employed one dimensional (1D) principal component analysis (PCA) [7] and one dimensional (2D) PCA [8] to extract compressed features from the spectrogram. In [9], Tivive *et al.* employed a set of 2D log-Gabor filters for feature extraction before applying 2D PCA for dimensionality reduction. Bilik *et al.*, on the other hand, developed an approach based on frequency domain, which extracts three types of speech processing features: real cepstrum, linear predictive coding, and mel-frequency cepstrum coefficients (MFCC) [10]. In most of existing methods, the feature extraction and target classification are solved separately, which may result in sub-optimal classification performance.

In this paper, a deep hierarchical network architecture, known as convolutional neural network (CNN), is developed for human gait classification. The first few convolutional layers of the network are designed to extract from elementary to complex features, and the last few layers are connected to operate as a classifier. All the processing layers are trained using supervised learning. To design an optimal architecture, the network configuration parameter and the training parameters, which are considered as hyperparameters, are obtained using a Bayesian optimization technique.

2. TIME-FREQUENCY RESPRESENTATION

2.1. Time-frequency analysis

A complex target such as a human can be represented as a set of point scatterers. For a point scatterer, the Doppler radar measures the backscattered power as a function of range and

velocity. Let λ be the wavelength of the transmitted signal and $v(t)$ be the radial velocity. The instantaneous time-varying phase change associated with the point scatterer is given by

$$\phi(t) = \frac{4\pi}{\lambda} \int_0^t v(\tau) d\tau. \quad (1)$$

The received Doppler radar signal can be modeled as

$$x(t) = a(t) e^{j(\omega t + \phi(t))}, \quad (2)$$

where $a(t)$ is the reflectivity of the target, and ω is the carrier angular frequency. Equation (2) shows $x(t)$ is a non-stationary signal.

For depicting the μ-D signatures, the radar signal backscattered from the walking person is converted into a joint time-frequency (T-F) distribution using either Short-time Fourier Transform (STFT) or S-method. Let $w(\tau)$ denote the time window function. The STFT of the μ-D signal $x(t)$ is given by

$$F_x(t, \omega) = \int_{-\infty}^{\infty} x(t + \tau) w(\tau) e^{-j\omega\tau} d\tau. \quad (3)$$

To achieve better T-F resolution while minimizing the cross-term interferences, the S-method can be used for T-F analysis. The T-F representation obtained from the S-method is computed as

$$S_x(t, \omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} P(\theta) F_x(t, \omega + \frac{\theta}{2}) F_x^*(t, \omega - \frac{\theta}{2}) d\theta, \quad (4)$$

where * denotes the complex conjugate. Here, $P(\theta)$ is a finite frequency window whose width controls the cross-term suppression and auto-term resolution properties of the T-F distribution.

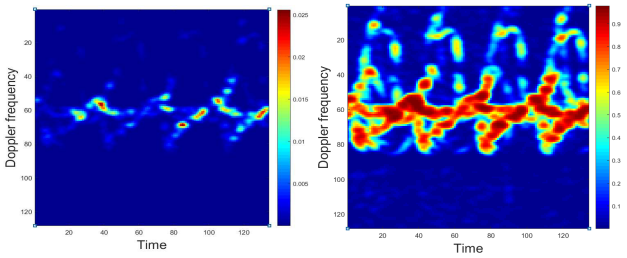
2.2. Local patch extraction

When a person is walking, the arm and leg motions generate μ-D modulations around the torso frequency. Therefore, instead of processing the entire time-frequency map, patches centered along the torso frequency are extracted and used as 2D inputs to the classifier. The patch height is determined relative to the height of the main μ-D peak, whereas the patch width is defined by a fixed length duration, which should be big enough to cover the periodic μ-D features. Furthermore, to reduce the variations of the main μ-D, which is induced by the leg swing or arm swing, the T-F patch is fixed to a size of $l_h \times l_w$, by down-sampling or up-sampling

To enhance the weak μ-D signatures, Naka-Rushton equation is applied to the extracted local T-F patch. This technique not only enhances the contrast of weak μ-D signatures but also removes background noise. Let p be a pixel of the local patch. The normalized value $\tilde{p}$ by Naka-Rushton equation is given by

$$\tilde{p} = \frac{1}{1 + (\frac{m}{p})^r}, \quad (5)$$

where m is the mean value of the patch, and $r \in \mathbb{R}^+$ is a constant controlling the slope of the input-output transfer characteristic.



(a) (b)

Fig. 1: A 2D local patch extracted from the time-frequency representation of a person walking with no-arm swinging: (a) before contrast enhancement,(b) after using Naka-Rushton equation.

3. HUMAN MOTION CLASSIFICATION

3.1. Network architecture

Deep CNNs have been shown to produce state-of-the art results in areas such as speech recognition [11], visual object detection [12], and image classification [13]. Most traditional classification approaches consist of two stages, feature extraction and classification, which are separately designed or trained. By contrast, a CNN performs both feature extraction and classification within the same architecture, where cascaded features of different complexity are extracted in the early layers and classification is performed in the later layers.

This paper proposes a CNN architecture with three main stages, where each stage has a stack of N_b blocks. Each block is composed of three layers: a convolutional layer, a batch normalization layer, and a rectified linear unit (ReLU) layer. A block can be seen as a sole feature layer to extract features automatically from the T-F patches. The three main stages are followed by a fully-connected layer and an output layer with softmax activation neurons. These neurons generate outputs in the range [0, 1]. The softmax activation neuron with the highest score is the predicted class of the input T-F patch. To reduce the number of parameters and avoid overfitting, a max-pooling layer of fixed size 2×2 with stride of 2 and no zero padding is employed after each stage, see Fig. 2.

Let Z_i be the *i*-th 2D input and K_j be the *j*-th filter kernel of size $\sigma_k \times \sigma_k$. The *j*-th output feature map Y_j is the sum of convolutions of the 2D inputs Z_i with the filter kernel K_j, then added with a trainable bias, b. Mathematically, the output

feature map can be computed as

$$Y_j = b + \sum_i K_j \otimes Z_i, \tag{6}$$

where $\otimes$ is the 2D convolution operators.

Since the proposed network architecture has three stages, the total number of convolutional layers is $3 \times N_b$, where N_b is the number of blocks in each stage. The number of kernels used in each stage is kept the same and proportional to $\frac{16}{\sqrt{N_b}}$, so as to keep the same number of trainable parameters in each stage. To keep a fixed size for every output feature map, we apply a zero padding of size $(\sigma_k - 1)/2$ with stride of 1 to the corresponding convolutional layer. In this paper, the kernel sizes applied in the first, second, and third stages are defined as 15×15, 7×7, and 3×3, respectively.

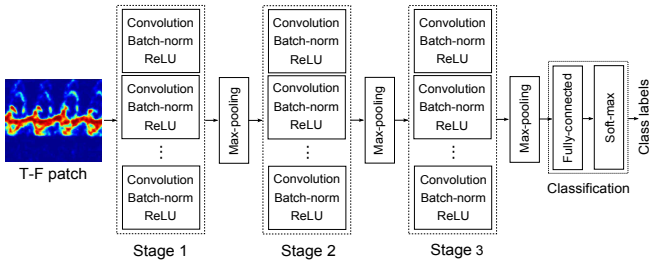


Fig. 2: The proposed deep CNN architecture for human motion classification.

3.2. Network training

The algorithm used for training the deep CNN architecture is the stochastic gradient descent with momentum [14]. Let the model parameters be denoted by $\boldsymbol{\theta}$. The training method minimizes the cross entropy function $\mathcal{L}(\boldsymbol{\theta})$ given by

$$\mathcal{L}(\boldsymbol{\theta}) = -\sum_{i=1}^{N}\sum_{j=1}^{3} t_{ij} \ln y_j(x_i, \boldsymbol{\theta}), \tag{7}$$

where t_{ij} is the indicator that the *i*-th sample belongs to the *j*-th class, $y_j(x_i, \boldsymbol{\theta})$ is the output generated by the *j*-th output neuron for the *i*-th input sample. To reduce overfitting, we add a regularization term for the weights to $\mathcal{L}(\boldsymbol{\theta})$. In this way, a penalty is applied for model complexity or extreme parameter values that can induce overfitting [15]. The regularized loss function can be written as

$$\mathcal{L}(\boldsymbol{\theta}) = -\sum_{i=1}^{N}\sum_{j=1}^{3} t_{ij} \ln y_j(x_i, \boldsymbol{\theta}) + \rho\Omega(\boldsymbol{\theta}), \tag{8}$$

where ρ is the L2 regularization coefficient. The function $\Omega(\boldsymbol{\theta})$ denoting the sum-of-squares of the parameters is given by: $\Omega(\boldsymbol{\theta}) = \frac{1}{2}\boldsymbol{\theta}^{\mathrm{T}}\boldsymbol{\theta}$.

To minimize the loss function, the model parameters are updated as follows:

$$\boldsymbol{\theta}_{m+1} = \boldsymbol{\theta}_m - \alpha\nabla_{\boldsymbol{\theta}}\mathcal{L}(\boldsymbol{\theta}_m) + \mu(\boldsymbol{\theta}_m - \boldsymbol{\theta}_{m-1}), \tag{9}$$

where α is a positive learning rate, and μ denotes the momentum added to the updates by the contribution of the previous gradient step to the current iteration. Here, $\nabla_{\boldsymbol{\theta}}$ is the gradient vector.

3.3. Bayesian optimization

Bayesian optimization (BO) is a powerful algorithm strategy to finding the extrema of unknown objective functions. It assumes this function is sampled from a Gaussian process and estimates it with a surrogate function [16]. In other words, BO typically works if the closed-form expression of the objective function is unknown, but we can obtain several observations of this function. Here, the BO technique is used to determine the optimal network architecture and training method parameters (i.e. hyperparameters) by minimizing the validation error.

Let $\mathbb{P}$ be the CNN hyperparameter space, which comprises the following key parameters: the number of blocks N_b, the learning rate α, the momentum μ, and the L2 regularization coefficient ρ. With these terminology, the objective function $\mathcal{F}$ can be modeled as

$$\mathcal{F} : \mathbb{P}(N_b, \alpha, \mu, \rho) \subset \mathbb{R}^4 \rightarrow \mathbb{R}. \tag{10}$$

Finding the optimal configuration in the CNN hyperparameter space can be defined as finding $\mathbf{p}^* \in \mathbb{P}$ such that

$$\mathbf{p}^* = \arg\min_{\mathbf{p}\in\mathbb{P}} \mathcal{F}. \tag{11}$$

Given the observations of the objective function $D_{1:n} = \{\mathbf{p}_{1:n}, \mathcal{F}(\mathbf{p}_{1:n})\}$, BO constructs a probabilistic model for $\mathcal{F}(\mathbf{p})$ and then exploit this model to determine the next location in $\mathbb{P}$ to sample. There are two main steps in this algorithm [17]: firstly, the Gaussian process is adopted to estimate the posterior distribution reflexing the updated beliefs about $\mathcal{F}$; secondly, an acquisition function $\mathcal{U}$ constructed from the posterior model is used to select the best point to evaluate the function $\mathcal{F}$.

Algorithm 1. BO for hyperparameter selection.

1: **for** i = 1 to N **do**
2: Find $\mathbf{p}^* = \arg\max_{\mathbf{p}} \mathcal{U}(\mathbf{p}|D_{1:n})$.
3: Sample the objective function: $f^* = \mathcal{F}(\mathbf{p}^*)$.
4: Augmented the data: $D_{1:n+1} = \{D_{1:n}, (\mathbf{p}^*, f^*)\}$.
5: Update the Gaussian process.
6: **end for**

The quality of a Gaussian process can be significantly affected by the covariance function, which decides the smoothness properties of samples drawn from it. This paper uses the

ARD Matern 5/2 covariance function for the Gaussian process [18]. The chosen acquisition function is the Expected Improvement as described in [19].

4. EXPERIMENTS AND ANALYSIS

4.1. Experimental setup

A continuous wave Doppler radar, ST200, with a carrier frequency of 24 GHz was used to acquire radar data. A database of μ-D signals from 18 subjects (7 females and 11 males) was generated to evaluate the CNN-based classification method [9]. Each subject performed three key human motion types: walking freely (two-arms swinging), parading while carrying an object (one-arm swinging), and carrying a heavy object (no-arm swinging). Each motion type was repeated three times. As a result, the total number of recorded Doppler signals of length 10 seconds was 162. The T-F patches were extracted along these signals with a fixed size of 128×134, which is equivalent to a length duration of 2 seconds. To evaluate the performance of the proposed method, the 2D patches were divided into 6 folds to ensure that the subjects used in the training set were not used in the test set. In this paper, the parameter r for Naka-Rushton equation was set to 1.

Table 1: The number of samples for experiments.

Folds	**1**	**2**	**3**	**4**	**5**	**6**
Training samples	16401	16434	16491	16527	16566	16461
Test samples	3240	3345	3273	3252	3171	3318

For tuning the hyperparameters during BO process, 500 samples picked randomly from the training set were used as validation set. The value range of the optimizable hyperparameters were defined as follows $N_b \in [1, 4]$, $\alpha \in [10^{-3}, 5 \times 10^{-2}]$, $\mu \in [0.8, 0.95]$, and $\rho \in [10^{-10}, 10^{-2}]$.

4.2. Experimental results

The proposed model was evaluated across different runs of 6-fold cross-validation so that all the samples are used for training and testing. The performance of our model can be assessed by taking the average classification rates of all runs. Table 2 presents the experimental results produced by BO. Each run, on average, achieves a classification rate of 96.85%. The optimal number of blocks N_b obtained from BO technique is 3, i.e., a total number of 32 layers.

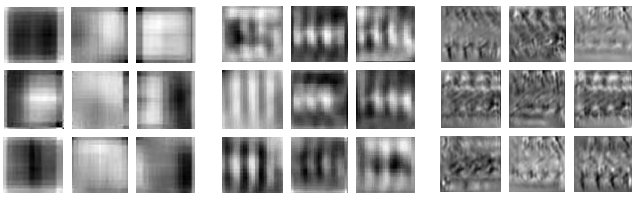
For comparison, three existing feature-based methods were also evaluated: CVD [6], log-Gabor filtering combined with $(2D)^2$-PCA [9], and MFCC [10]. In the log-Gabor-based method, the number of log-Gabor filters was 32, i.e., 4 scales and 9 orientations. In the MFCC-based method, 64 mel-scale cepstrum coefficients were extracted using 40 triangular bandpass filters. In the CVD-based method, the first three harmonic frequencies and the velocity profiles were extracted as features. Table 3 shows the comparison of classification rates obtained by different feature extraction methods over the 6 cross-validation folds. Among the tested methods, the proposed CNN-based classification method achieves the best performance. The feature visualizations shows that the network produces a glance to the μ-D signatures in the early stages (Fig. 3a, 3b), while the deeper details are explored in the last convolutional layers (Fig. 3c).

Table 2: The optimal hyperparameters and the corresponding classification rate across different runs. SE stands for the standard error.

Runs	**Optimal hyperparameters**				**CRs** $\pm$ **SE**(%)
	N_b	α	μ	ρ	
1	3	0.00100	0.94271	1.45×10^{-10}	97.65±2.3
2	3	0.00107	0.94722	1.99×10^{-10}	95.95±4.1
3	3	0.00567	0.86391	8.31×10^{-3}	96.80±3.2
4	3	0.03086	0.82040	1.02×10^{-10}	97.23±2.7
5	3	0.00101	0.92532	9.32×10^{-9}	96.60±3.4
Average					**96.85±1.3**

Table 3: Comparison of the proposed method with other feature extraction methods for human motion classification.

Methods	Proposed	log-Gabor	MFCC	CVD
CRs $\pm$ **SE**(%)	96.85±1.3	91.3±6.9	72.7±7.2	62.3±5.1



(a) (b) (c)

Fig. 3: A visualization of some learned features from low level to high level: (a) by the first stage, (b) by the second stage, (c) by the third stage.

5. CONCLUSION

This paper presents a deep learning method for classifying Doppler radar signals from human walking with different arm-motions. The Doppler radar signal is converted into a T-F representation, where local patches are extracted along the main Doppler shift. A flexible deep CNN architecture is designed to classify the local patches into three categories. To obtain the optimal network architecture, a Bayesian learning technique is employed to find four key hyperparameters. Experimental results show that the proposed method achieves promising outcomes in comparison with other techniques.

6. REFERENCES

[1] Y. Li, L. Du, and H. Liu, "Hierarchical classification of moving vehicles based on empirical mode decomposition of micro-doppler signatures," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 51, no. 5, pp. 3001–3013, 2013.

[2] S. H. Park, "Automatic target recognition using jet engine modulation and time-frequency transform," *Progress In Electromagnetics Research M*, vol. 39, pp. 151–159, 2014.

[3] A. R. Persico, C. Clemente, C. Ilioudis, D. Gaglione, J. Cao, and J. Soraghan, "Micro-Doppler based recognition of ballistic targets using 2D Gabor filters," in *Sensor Signal Processing for Defence*, 2015, pp. 1–5.

[4] V. C. Chen, F. Li, S. S. Ho, and H. Wechsler, "Analysis of micro-Doppler signatures," *Radar, Sonar and Navigation*, vol. 150, no. 4, pp. 271–276, 2003.

[5] Y. Kim and H. Ling, "Human activity classification based on micro-Doppler signatures using a support vector machine," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 47, no. 5, pp. 1328–1337, 2009.

[6] S. Bjorklund, T. Johansson, and H. Petersson, "Evaluation of a micro-Doppler classification method on mm-wave data," in *Proc. IEEE Radar Conf.*, 2012, pp. 0934–0939.

[7] B. G. Mobasseri and M. G. Amin, "A time-frequency classifier for human gait recognition," in *Proc. of SPIE Defense, Security, and Sensing*, 2009, vol. 7306, pp. 730628–9.

[8] J. Li, S. L. Phung, F. H. C. Tivive, and A. Bouzerdoum, "Automatic classification of human motions using Doppler radar," in *Proc. IEEE Int. Joint Conf. on Neural Networks*, 2012, pp. 1–6.

[9] F. H. C. Tivive, S. L. Phung, and A. Bouzerdoum, "Classification of micro-Doppler signatures of human motions using log-Gabor filters," *IET Radar, Sonar and Navigation*, vol. 9, no. 9, pp. 1188–1195, 2015.

[10] I. Bilik and P. Khomchuk, "Minimum divergence approaches for robust classification of ground moving targets," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 48, no. 1, pp. 581–603, 2012.

[11] D. Guiming, W. Xia, W. Guangyan, Z. Yan, and L. Dan, "Speech recognition based on convolutional neural networks," in *IEEE Int. Conf. on Signal and Image Processing*, 2016, pp. 708–711.

[12] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov, "Scalable object detection using deep neural networks," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2014, pp. 2147–2154.

[13] Y. LeCun, B. Yoshua, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[14] V. Patel, "Kalman-based stochastic gradient method with stop condition and insensitivity to conditioning," *SIAM Journal on Optimization*, vol. 26, no. 4, pp. 2620–2648, 2016.

[15] E. Phaisangittisagul, "An analysis of the regularization between L2 and dropout in single hidden layer neural network," in *Proc. IEEE Int. Conf. Intelligent Systems, Modelling and Simulation*, 2016, pp. 174–179.

[16] M. Aoki, "On some convergence questions in bayesian optimization problems," *IEEE Trans. on Automatic Control*, vol. 10, no. 2, pp. 180–182, 1965.

[17] T. T. Joy, S. Rana, S. Gupta, and S. Venkatesh, "Hyperparameter tuning for big data using Bayesian optimisation," in *Proc. IEEE Int. Conf. on Pattern Recognition*, 2016, pp. 2574–2579.

[18] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Advances in Neural Information Processing Systems*, 2012, pp. 2951–2959.

[19] J. Mockus, V. Tiesis, and A. Zilinskas, *Toward global optimization*, vol. 2, pp. 117–128, Elsevier, 1978.