Transfer learning for human activity classification in multiple radar setups

Jeremy Fix⋆, Israel Hinostroza†, Chengfang Ren†, Giovanni Manfredi†, Thierry Letertre†
§*CentraleSupélec, 57000 Metz, France*
† *Université Paris-Saclay, CentraleSupélec, SONDRA, 91190 Gif-sur-Yvette, France*
⋆*Université de Lorraine, CentraleSupélec, CNRS, Loria, 57000 Metz, France*
jeremy.fix@centralesupelec.fr, israel.hinostroza@centralesupelec.fr

***Abstract*—Deep Learning techniques require vast amount of data for a proper training. In human activity classification using radar signals, the data acquisition can be very expensive and takes a lot of time, but radar databases are starting to be available to the public. In this work we show that we can use these available radar databases to pretrain a neural network that will finish its training on the final radar data even though the radar configuration is different (geometry configuration and carrier frequency).**

***Index Terms*—deep learning, transfer learning, micro-doppler, cadence velocity, human activity**

I. Introduction

Machine learning and especially deep learning are known to require a vast amount of annotated data. In computer vision, it is common to benefit from large corpora of data, even out of the domain of interest, by pretraining the predictor on this dataset and then finetuning on the data of interest. In this paper, we are interested in activity classification from continuous wave (CW) radar signals. When someone acts in front of a CW radar system, its motion alters the frequency of the radar wave due to the Doppler effect which can be detected and hopefully used to classify the "observed" activity. However, as in any other machine learning problem, the question of the availability of large amounts of data is raised. In this paper, we study to which extend, as in computer vision or other domains, training a neural network can benefit from transfer learning with a, possibly out of domain, large dataset of activity recordings from radar signals. The paper is structured as follows: in section II we present the different recording setups and available data, their preprocessing is detailed in section III. The neural network architecture and training procedure are presented in section IV before the transfer learning approach in section V and the results in section VI. A discussion on the results and further works concludes the paper.

II. Datasets

In this paper, we consider two sources of data consisting of radar signal recordings. The first dataset is a large collection provided in the context of the radar challenge proposed during the IET 2020 International Radar Conference[1]. The second dataset was collected by our SONDRA research group. In both datasets, 6 activities were considered, but two of them are different, depending on the source. The main characteristics of those datasets are presented in table I. All the activities induce a different velocity pattern which ultimately leads to different micro-doppler signatures.

A. Radar challenge dataset

The FMCW radar dataset used in this work has been downloaded from [1]. The data was collected using an off-the-shelf linear FMCW radar (by Ancortek) operating at C-band (5.8 GHz) with a bandwidth of 400 MHz and chirp signal (linear frequency modulation) duration of 1 ms, delivering an output power of approximately 18 dBm. The transmitting and receiving antennas have a gain of about 17 dBi [1]. The radar data is composed of de-chirped complex signals (128 $I - Q$ samples per sweep or chirp). Six activities must be classified using radar measurements: falling, picking up, drinking, walking, sitting down, and standing up. All the classes were almost balanced, only the falling class had 30% samples less than the other classes. The recordings were performed in 10 different rooms.

B. SONDRA dataset

A CW radar dataset have been collected in the main corridor of our laboratory. It is composed of complex signals of sampling frequency of 2 KHz after demodulation and decimation. For the transmitter, we used a E4438BC Agilent signal generator, output power of 10 dBm and antenna gains of 10 dBi, 6 dBi and 0 dBi for 4 GHz, 2.35 GHz and 0.9 GHz, respectively. For the receiver, we used a software defined radio USRP 2495R from National Instruments, internal gain of 20 dB and antenna gain of 10 dBi for all the frequencies. We have considered two scenarios: active (transmitter and receiver synchronization) line-of-sight, quasi-monostatic, hence similar to the Radar challenge scenario (although the waveform and some activities are different); and passive through-the-wall radar (one antenna behind a wall, distance between antennas of 3.4 m, transmitter and receiver not synchronized), which is a different and more complex configuration compared to the Radar challenge dataset.

We purposefully considered settings that are similar to the radar challenge with the 4 GHz carrier frequency line-of-sight quasi-monostatic setup but also tried to push the limits by studying the performance of transfer on a completely different

[1] https://humanactivitiyclassificationwithradar.grand-challenge.org/

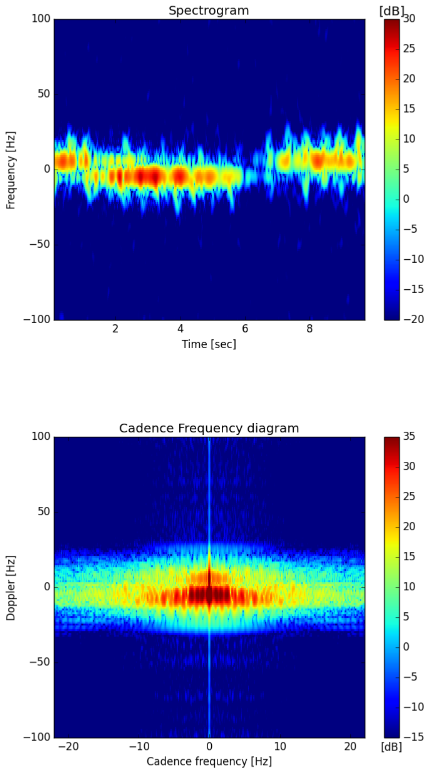

Fig. 1: Spectrogram and cadence velocity representations for the walking activity at 0.9 GHz CW.

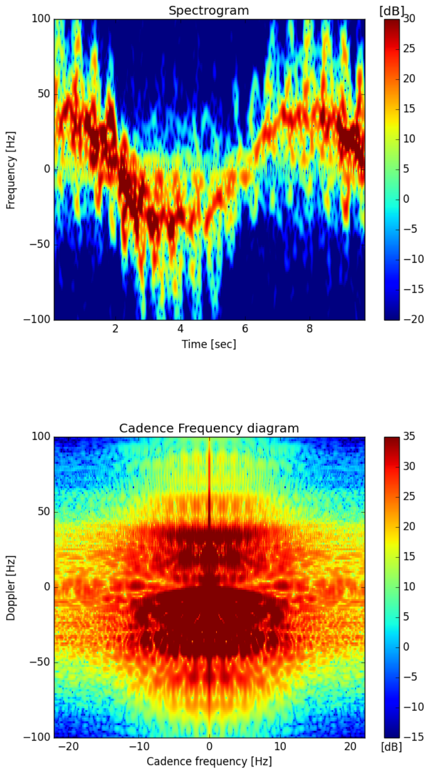

Fig. 2: Spectrogram and cadence velocity representations for the walking activity at 4 GHz.

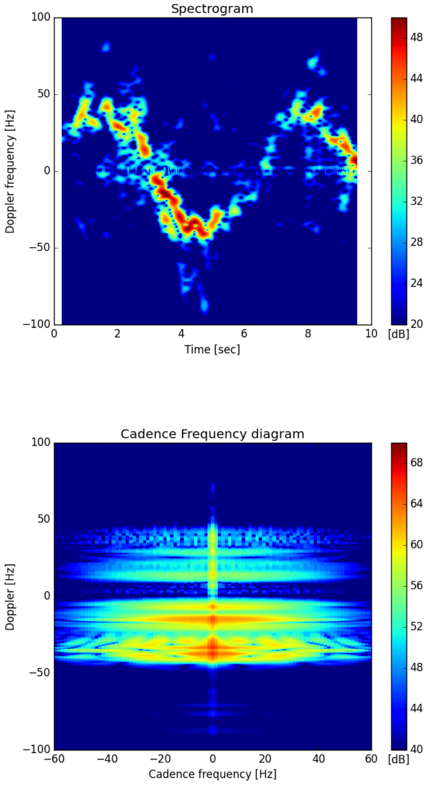

Fig. 3: Spectrogram and cadence velocity representations for the walking activity at 5.8 GHz FMCW.

TABLE I: Main characteristics of the two radar datasets used in this paper.

	Radar Challenge	SONDRA dataset		
Carrier Freq.	5.8 GHz	4 GHz	2.35 GHz	0.9 GHz
Radar config.	quasi monostatic		bistatic (baseline 3.4m)	
Scenario config.	line of sight		through-the-wall	
Waveform	FMCW	CW		
Nb. of Samples	1752	300	407	455
Activities	walk			
	sit down			
	stand up			
	pick up			
	drink	walk with object		
	fall	put down object		

setup with 2.35 GHz and 0.9 GHz carrier frequencies in a through-the-wall passive radar.

III. Preprocessing

Since the target SONDRA dataset exploits a CW signal, no range information is available (whereas in our previous work [2], three representations were extracted: spectrogram, doppler-range and range-time), hence only two signal representations were considered: spectrogram and cadence-velocity [3] and the FMCW radar signals are transformed into a synthesized CW to collapse the range information.

A. Synthesized CW signal from FMCW data

The raw-data of the Radar challenge dataset are reshaped as a $128 \times n_{chirps}$ matrix, where n_{chirps} is the number of chirps. The chirps last 1 ms, and the number of chirps depends on the recordings lasting either 5, 10, or 20 seconds. In the following, we denote as fast time the temporal axis along the 128 samples and as slow time the temporal axis along the n_{chirps} lines.

The range-time representation is computed from a Fourier transform along the fast time.

To get spectrograms independent of range, hence closer to what we obtain with a CW radar, we need to get rid of the range information by ”synthesizing” CW signals. This is obtained by aggregating all the range bins, as expressed in the next equation:

$$C_{r\ prof}(t) = (c_0(t), c_1(t), \ldots, c_{N-1}(t))^T$$
$$S_{CW}(t) = \sum_{k=0}^{N-1} c_i(t) \qquad (1)$$

where $C_{r\ prof}(t) \in \mathbb{C}^N$ is the complex-valued range profile vector (N elements, in our case 128, samples per sweep) for a specific t (slow time, an integer multiple of chirp duration, 1 ms in our case), $S_{CW}(t) \in \mathbb{C}$ the complex-valued synthesized

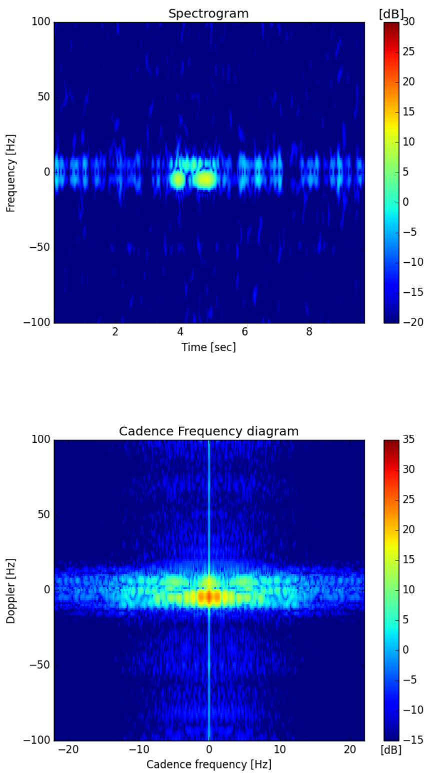

Fig. 4: Spectrogram and cadence velocity representations for the sit activity at 0.9 GHz CW.

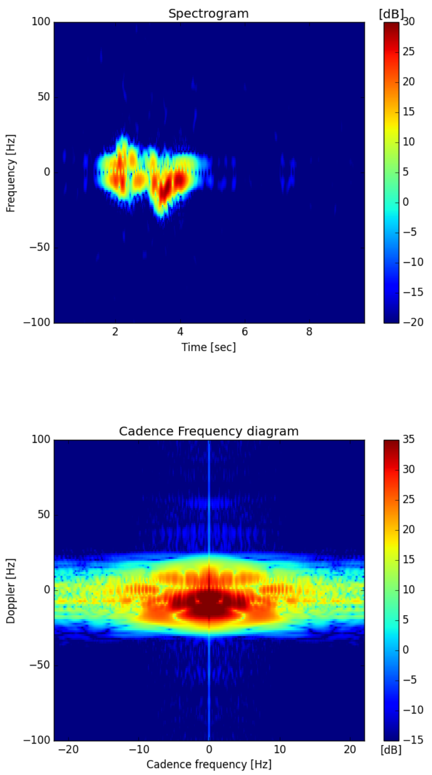

Fig. 5: Spectrogram and cadence velocity representations for the sit activity at 4 GHz MHz.

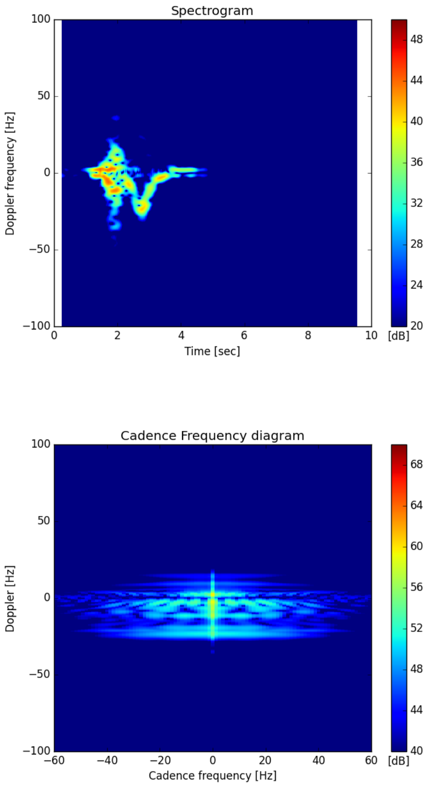

Fig. 6: Spectrogram and cadence velocity representations for the sit activity at 5.8 GHz FMCW.

CW signal for a specific t. Notice that after this operation we obtain a single complex value for each sweep (1 ms of duration), hence, the final effective sampling is 1 KHz. After this, we can obtain spectrogram and cadence-velocity diagrams as in the case of a regular CW signal as detailed in the next sections.

B. Spectrogram and Cadence-velocity

For the spectrogram obtained from the radar challenge dataset we used a window of 0.512 s and time stride of 12 ms. For the SONDRA dataset we used a window of 0.3 s and time stride of 30 ms.

For the cadence-velocity diagram, which is a Fourier transform from the spectrogram for each fixed Doppler frequency (hence along the time axis, see [3]), we truncated the cadence frequency axis (horizontal axis) up to ±60 Hz for the radar challenge dataset, and up to ±22 Hz for the SONDRA dataset.

As seen in the figures 1-6, the Doppler frequency gets closer to 0 Hz as the carrier frequency decreases, hence the representations (spectrogram and cadence-velocity) were truncated in the Doppler frequency axis (vertical axis) up to ±100 Hz for the higher carrier frequencies 5.8 GHz (radar challenge) and 4 GHz (SONDRA dataset). For the other frequencies (SONDRA dataset), the truncation was done up to ±60 Hz and ±23 Hz, for the carrier frequencies of 2.35 GHz and 0.9 GHz, respectively. Note that the dB scale does not any physical meaning because it depends on several internal factors of the hardware of the system.

IV. Convolutional neural network classifier

We use the same neural network as in [2] in their submission for the radar challenge except that, here, we do not use the same input representation. As detailed in the previous section, two representations are extracted from both the FMCW and CW radar signals: a spectrogram and a cadence velocity representation which are resized to 128×128 matrices. Each diagram feeds a branch of the neural network before being concatenated and feeding a classification head. The neural network architecture is summarized in table II. The inputs are normalized by division by the maximum value and centered/reduced, the statistics being computed on the training set and used for normalizing the training and validation data. The normalization coefficients are computed for every experiment from their own training set.

Given inputs of size 128×128, the output feature maps of the convolutional stages, before the global average poolings, are 8×8 in spatial dimension and the depth of the stack of convolutional and pooling layers make their receptive field cover the whole input.

TABLE II: Neural network architecture. Conv(n) denotes a 2D convolutional layer with n kernels of size $(3, 3)$. Every convolution has a stride 1 and zero padding of size 1. MaxPool is a 2D max pooling layer with kernel size $(2, 2)$, stride 2.

Cadence velocity (128 × 128)	Spectrogram (128 × 128)
Conv(64), Relu Conv(64), Relu MaxPool, Dropout(0.5)	Conv(64), Relu Conv(64), Relu MaxPool, Dropout(0.5)
3× [BatchNorm Conv(64), Relu Conv(64), Relu MaxPool, Dropout(0.5)]	3× [BatchNorm Conv(64), Relu Conv(64), Relu MaxPool, Dropout(0.5)]
BatchNorm Conv(64), Relu Conv(64), Relu GlobalAvg, Dropout(0.5)	BatchNorm Conv(64), Relu Conv(64), Relu GlobalAvg, Dropout(0.5)
Fully connected (128), Relu Dropout(0.5)	
Fully connected (6), Softmax	

The neural network is optimized with Adam, using an initial learning rate $\epsilon = 0.001$, halving the learning rate every 10 epochs. The batch size is 16. For regularization, in addition to the dropout layers, we use a L2 regularization with $\lambda = 0.003$. A validation split is built from 20% of the data randomly chosen from the training set. The best model is selected by early stopping on the validation accuracy.

The parameters are initialized with the default strategy of pytorch [4] for convolutional and linear layers, namely the bias and the weights are all sampled from a uniform distribution in $[-\frac{1}{\sqrt{fan\_in}}, \frac{1}{\sqrt{fan\_in}}]$

V. Transfer learning

In order to investigate the interest of using a large FMCW dataset for training a classifier on CW data, we consider a very simple parameter based approach of transfer learning [5]. As it is common in other machine learning domains (e.g. image classification with ImageNet [6]) we pretrain the neural network on the FMCW dataset as in [2] and then reuse the best learned parameters as a starting point for the gradient descent over the CW dataset. For training on the CW dataset, we could reinitialize at random the parameters of the classification head but as some activities overlap, the linear softmax layer is not reinitialized at random. Instead, we rather re-arrange the weights of the output layer during the transfer so that the classes of the target dataset reuse knowledge extracted from the initial pretraining. All the parameters of the architecture are trainable (none of them are frozen). To study the interest of transfer, we also train the networks from a random initialization.

VI. Results

The learning curves for the training are shown on figure 7 for the CW data at 0.9 GHz, on figure 8 for the CW data at 2.35 GHz and on figure 9 for the CW data at 4 GHz. The two metrics of the cross entropy loss and the accuracy are depicted on both figures. The first observation for all the CW carrier frequencies is that the models learn faster with pretrained parameters than from randomly initialized parameters: the same accuracy is reached with transfer in half the number of steps required to reach the same accuracy without transfer. This gives sense to the pretraining: some knowledge, extracted during the training on the FMCW dataset, is certainly reused when the neural network is finetuned on the CW datasets.

Maybe more surprisingly, in addition to be faster, the transfer learns a better model. If we look at the accuracies, the transfer ends up with a gain from 5% at 0.9 GHz to 10% at 2.35 GHz and 20% at 4 GHz on the validation sets. There is a also a gain on the cross entropy loss meaning the models are more certain in their predictions of the correct classes. Although the cross-entropy is averaged over the samples, it is reasonable to assume that the model trained with transfer is more confident about its predictions. The influence of one sample in the cross-entropy loss is $\log(p_i)$ with p_i the probabilities assigned by the model to the correct class. The benefit of the transfer learning is higher for the CW carrier frequency most similar with the FMCW carrier frequency. There might be several factors explaining this. First, if the FMCW and CW carrier frequencies are close, the representations themselves are more similar and the neural network can rely on similar pattern detectors to decide the class. The setup for which the largest improvement in the performance is observed is for a 4 GHz carrier frequency in a quasi-monostatic line-of-sight configuration, almost the same as for the radar challenge. Second, the setups with the carrier frequency of 2.35 GHz and 0.9 GHz are through-the-wall and bi-static, therefore a really different setup from the original 4 GHz line-of-sight setup where the velocity profiles are expected to be different. Still, even in these extreme situations, we observe some benefits of the pretraining.

Finally, it is not shown on the figures but with transfer learning, we observed the learning curves were less varying and were more consistent across multiple runs.

VII. Discussion

The results of this paper show that training a neural network can benefit from large datasets of radar recordings to train feature detectors that can generalize to other setups. The setup variation we explored are both for the alignment between the radar apparatus and the moving target and the carrier frequency. For the carrier frequency, our experiments show that it is possible to benefit from a pretraining on a higher carrier frequency and to transfer on lower ones. It is yet unclear if the benefits would be similar or not if we would have operated in the other direction.

In one experiment, not shown in the paper, we tried to change the content of the FMCW representations to better match what we would expect if the carrier would have been the same as the one for the CW data. The hypothesis was that more similar carrier frequency lead to more similar features to extract. We therefore degraded the FMCW signals sampled at

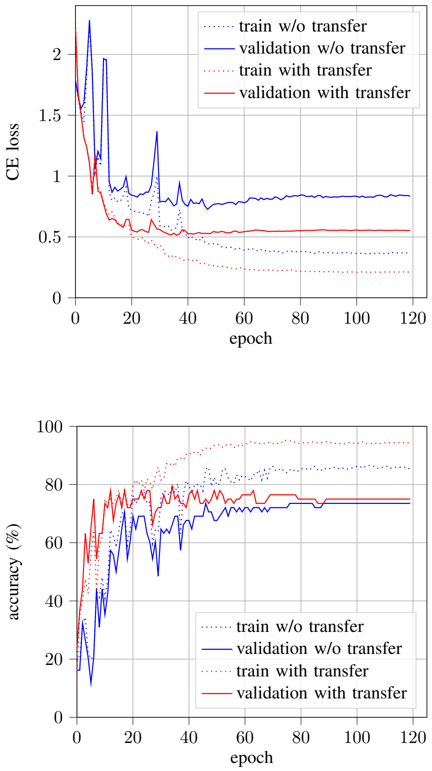

Fig. 7: Cross-entropy loss (top) and accuracy (bottom) of the best runs with and without transfer learning for 900 MHz.

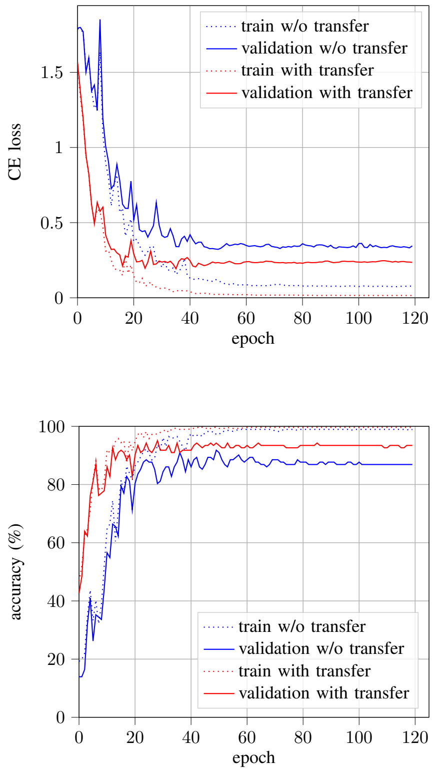

Fig. 8: Cross-entropy loss (top) and accuracy (bottom) of the best runs with and without transfer learning for 2.35GHz.

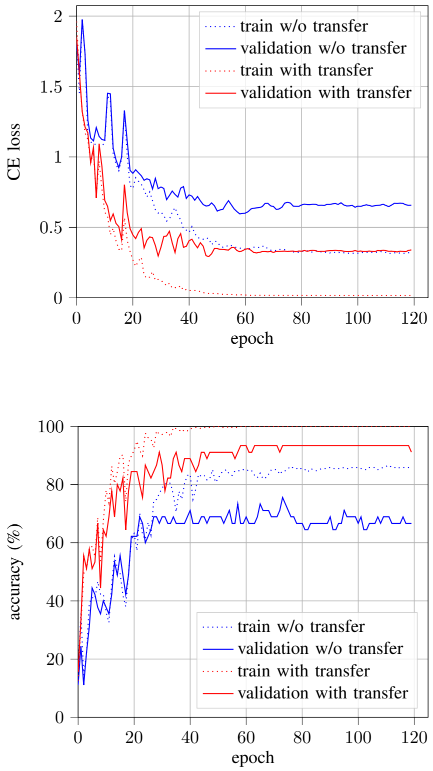

Fig. 9: Cross-entropy loss (top) and accuracy (bottom) of the best runs with and without transfer learning for 4GHz.

5.8 GHz down to 0.9 GHz before transferring the network to the 0.9 GHz CW signals but the results were not as good as without the degradation. The neural network was more likely to overfit the CW dataset. Certainly, a careful exploration of the hyperparameters of the training is required to conclude. It would be worth investigating other forms of regularization than the ones we considered in this paper such as the recently introduced Mix-Up [7] or label smoothing [8].

In any case, the pretraining approach is, even if promising, still a very naive approach of transfer learning and there are plenty of other approaches proposed in the literature on meta- and few-shot learning [5] that would be worth investigating in future works.

Acknowledgment

We thank the Eurométropole de Metz, the region Grand Est, CentraleSupélec, the fondation CentraleSupélec, the Conseil Département de la Moselle for financing the Data Centre d'Enseignement that provided computational resources used for part of the experiments.

References

[1] F. Fioranelli, S. A. Shah, H. Li, A. Shrestha, S. Yang, and J. L. Kernec. (2020) Radar signatures of human activities. [data collection]. [Online]. Available: http://researchdata.gla.ac.uk/848/

[2] P. Cadart, M. Merlin, G. Manfredi, J. Fix, C. Ren, I. Hinostroza, and T. Letertre, "Classification in C-band of doppler signatures of human activities in indoor environments," in *IET International Radar Conference (IET IRC 2020)*, vol. 2020, 2020, pp. 412–416.

[3] W. Zhang and G. Li, "Detection of multiple micro-drones via cadence velocity diagram analysis," *Electronics Letters*, vol. 54, no. 7, pp. 441–443, 2018. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/el.2017.4317

[4] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035.

[5] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, 2021.

[6] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision – ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham: Springer International Publishing, 2014, pp. 818–833.

[7] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *International Conference on Learning Representations*, 2018. [Online]. Available: https://openreview.net/forum?id=r1Ddp1-Rb

[8] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.