Università degli Studi di Ferrara

DIPARTIMENTO DI INGEGNERIA

CORSO DI LAUREA TRIENNALE IN INGEGNERIA ELETTRONICA E INFORMATICA

RETI DI TELECOMUNICAZIONI E INTERNET

An AI-based methodology for crowd counting via wireless sensing

Relatore:
Chiar. mo Prof. Andrea Conti

Laureanda:
Boukayoud Hiba

Correlatore:
Dott. Alessandro Vaccari

Anno Accademico 2024-2025

Sommario

Le reti wireless di nuova generazione utilizzeranno il sensing come mezzo per abilitare una serie di nuovi servizi. Tra questi, la stima del numero di persone simultaneamente presenti in un ambiente è essenziale per consentire applicazioni in tema di sicurezza, assistenza e gestione energetica. Per monitorare il numero di persone, considerando anche quelle non connesse alla rete, gli approcci convenzionali si basano sull'utilizzo di videocamere e microfoni. Tuttavia, tali tecnologie forniscono elevata accuratezza solo in condizioni ambientali controllate, ad esempio in condizioni di linea di vista. Inoltre, esse non garantiscono la privacy dal momento che necessitano della registrazione di immagini e suoni. Tali limitazioni hanno fatto si che l'attenzione si spostasse su approcci basati su segnali a radiofrequenza. Le tecnologie radar consentono di collezionare forme d'onda riflesse dalle persone presenti in un'area d'interesse. In particolare, radar multiantenna operanti ad onde millimetriche rappresentano una tecnologia promettente per il conteggio di obiettivi in tempo reale grazie alla loro notevole risoluzione sia nella stima di distanza che di angolo. L'utilizzo delle reti neurali consente il processamento dei segnali radar al fine di stimare il numero di persone senza richiedere ulteriore processamento o conoscenza preliminare dell'ambiente d'interesse.

Questa tesi mira a stimare il numero di persone simultaneamente presenti in un ambiente a partire dai campioni delle forme d'onda riflesse, acquisite tramite un radar a onde millimetriche operante a 77 GHz. Si ottiene una rappresentazione dell'ambiente nel dominio delle distanze e delle velocità utilizzata come ingresso a una rete neurale che, risolvendo un problema di classificazione, fornisce in tempo reale una stima sul numero di persone. Questa tesi è organizzata come segue. Il capitolo 1 presenta l'importanza di avere a disposizione strumenti che permettono di fare sensing basato su segnali a radio-frequenza. Il capitolo 2 introduce i principi di funzionamento di radar multiantenna operante ad onde millimetriche; il capitolo 3 presenta le reti neurali e le loro applicazioni su dati radar; infine, il capitolo 4 descrive la configurazione sperimentale e i risultati ottenuti.

Abstract

Next-generation wireless networks will leverage sensing to support a range of new services. Among these, estimating the number of people simultaneously present in an environment is essential to enable applications in smart safety, energy management, and assistive services. To count unconnected (i.e., device-free) humans, existing systems rely on sensors based on cameras and microphones. These technologies guarantee high accuracy only under controlled environmental conditions, such as line-of-sight conditions. Moreover, they pose several privacy concerns due to the need of recording images and sounds. A promising approach to perform accurate people counting is the employment of reflected radio frequency (RF) signals. In particular, the presence of a device-free target can be revealed by processing the waveforms reflected by the human body. The employment of RF signals guarantees to operate in critical environmental conditions, while preserving the privacy. In particular, frequency modulated continuous wave (FMCW) radars offer a promising solution for real-time target counting thanks to the high resolution available in both range and angle.

This thesis aims to develop a neural network based framework for device-free target detection and counting in indoor environment using a millimeter-wave (mmWave) multiple-input multiple-output (MIMO) radar. The proposed framework is validated via experimentation employing an off-the-shelf radar at 77 GHz in an indoor environment.

Contents

Acronyms

AoA angle of arrival

CMOS complementary metal-oxide-semiconductor

CNN convolutional neural network

DFT discrete Fourier transform

DL Deep learning

DNN deep neural network

FFT fast Fourier transform

FMCW frequency modulated continuous wave

IF intermediate-frequency

MIMO multiple-input multiple-output

ML machine learning

MLPs multi-layer perceptrons

mmWave millimeter-wave

NLOS non-line-of-sight

NN neural network

RA range-angle

RD range-Doppler

ReLU rectified linear unit

RF radio frequency

RNN recurrent neural network

Chapter 1

Introduction

Real-time situational awareness is a key enabler for novel services in the next generation (xG) wireless networks [1–7]. In particular, target counting allows novel applications including smart safety procedures and emergency response [8–13], informs access control and flow management [14–19], and enables personalized assistance [20–23].

Sensing approaches are commonly distinguished in device-based and device-free. Device-based methods assume collaborative targets that carry a tag or a personal device providing precise analytics [24, 25]. However, it is often impractical in real settings because it requires continuous user compliance. In contrast, device-free methods allow situational awareness, non-collaborative targets using sensors installed in the environment.

Conventionally, target counting is performed using cameras, microphones, or thermal sensors [26]. While effective in controlled settings, their performance is strongly conditioned by the environment and by privacy requirements [27, 28]. Cameras need an unobstructed field of view and adequate illumination; acoustic and thermal sensors are sensitive to background activity and thermal drift. Moreover, cameras and microphones raise concerns about the recording of sensitive data. These limitations have led to increased study of device-free approaches based on radio signals, where information is extracted from the echoes that naturally arise when an emitted waveform reaches people and objects. Such approaches allow operation through partial occlusions and in poor visibility, and they do not require people to carry any device.

Among radio technologies, millimeter-wave (mmWave) multiple-input multiple-output (MIMO) radar is especially suitable for indoor sensing [29–32]. At millimeter wavelengths, the large available bandwidth translates into high range resolution [33–36], allowing the radar to distinguish nearby targets as separate since

their echoes arrive at different round-trip times. Despite these advantages, radar measurements are rife with clutter and multipath, resulting in unpredictable, time-changing interference and scene-dependent distortions [37, 38]. Because the measurements depend on each room and change over time, there is no single analytical rule that can reliably convert raw radar signals into an accurate target count across different indoor spaces. This points to data-driven models that learn the mapping from examples. Traditional machine learning (ML) models depend on hand-crafted features that often fail to generalize across domains [39]. Therefore, using Deep learning (DL) is preferable, as it removes manual feature design and adapts directly to the statistics of radar maps, learning hierarchical representations directly from data and allowing feature extraction and high-accuracy counting [40–46]. Processing the reflected signal produces range-Doppler (RD) and range-angle (RA) maps, which appear as image-like patterns [47, 48]. RD maps can be used as inputs to deep neural networks (DNNs) in order to detect target presence and perform counting [49, 50]. A conventional choice is to employ convolutional neural networks (CNNs) [51, 52]. RD frames are two-dimensional arrays with strong local correlations; convolutional layers adopt filters (i.e. localized receptive fields) and weight sharing across positions, leading to compact models with high statistical efficiency.

This thesis aims to develop and validate a device-free people-counting methodology based on mmWave frequency modulated continuous wave (FMCW) MIMO radar for realistic indoor environments [53]. Raw echoes are processed into RD and RA representations; counting is then performed as single-frame classification on RD maps using a compact CNN. The model operates one frame at a time to support low-latency use and its outputs are presented in an interactive web interface that continuously displays the current RD map, the corresponding RA view and the predicted number of moving targets.

The remainder of the thesis is organized as follows. Chapter 2 presents the operating principles of FMCW radar and the formation of RD and RA maps. Chapter 3 reviews neural architectures with emphasis on convolutional networks and their suitability for radar data. Chapter 4 details the experimental setup, the dataset preparation, the model used for counting, and the interactive visualization, reporting the achieved results in terms of accuracy. Lastly, the final chapter offers a brief overview of the results.

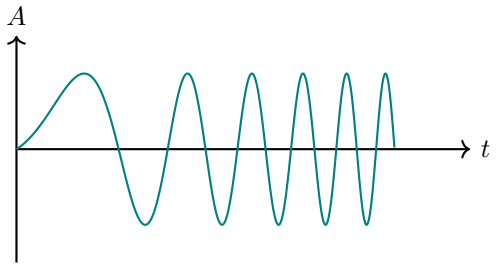
Chapter 2

Sensing via FMCW MIMO radar at mmWaves

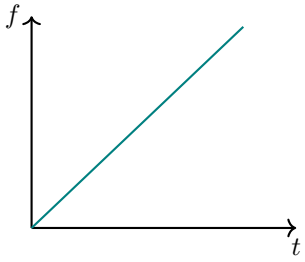
Sensing with radio waves offers a privacy–preserving alternative to cameras and microphones in indoor spaces, and it remains effective when visibility is poor or line–of–sight is obstructed. Within this family, mmWave FMCW radar is particularly well suited for people counting. The large bandwidth available at mmWave supports fine range discrimination, while the short wavelength enables compact antenna arrays and accurate angle estimation with compact devices. These properties make it possible to observe motion and spatial layout without requiring individuals to carry any device. FMCW radars continuously transmit a frequency-modulated signal, known as a chirp, that is reflected by objects. The reflected echoes are processed to estimate range, radial velocity, and angle of arrival.

The main advantages of this technology lie in its high-accuracy detection of minute motion as well as in the compactness of the components used for mmWave signal processing, including antennas.

These millimeter-scale components can be integrated on complementary metal-oxide-semiconductor (CMOS) chips which, together with the analog–digital blocks incorporating radio-frequency transmitters (TX) and rceivers (RX), overcome the challenges previously associated with increased power consumption, higher costs and the complexity inherent in handling high-frequency signals.



(a) Chirp signal, with amplitude as a function of time.



(b) Chirp signal, with frequency as a function of time.

Figure 2.1: FMCW chirp representation

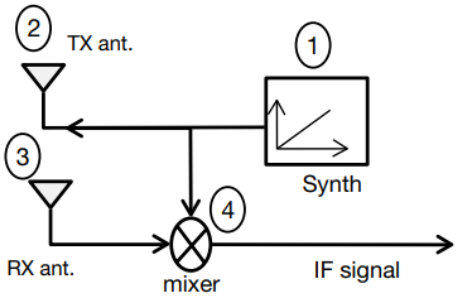


Figure 2.2: FMCW radar block diagram [30].

2.1 FMCW signal processing for analytics extraction

A single FMCW chirp is defined as

$$s_{\mathrm{tx}}(t) = \cos\left(2\pi f_0\, t + \pi S t^2\right), \tag{2.1}$$

with start frequency f_0, bandwidth B, duration T_{c}, and slope $S = B/T_{\mathrm{c}}$. The duration is the time interval of a single frequency sweep, while the slope is the rate of change of instantaneous frequency with time.
Figure 2.1 shows a chirp both in time and frequency: panel (a) depicts the waveform amplitude, whereas panel (b) shows the linear sweep of the instantaneous frequency.

Figure 2.2 shows the essential radar architecture which includes a synthesizer, two antennas (RX and TX), and a mixer.

A point target at range d returns a delayed, scaled replica

$$s_{\mathrm{rx}}(t) \approx \alpha\, \cos\Big(2\pi f_0(t_n - \tau) + \pi S(t_n - \tau)^2\Big) \tag{2.2}$$

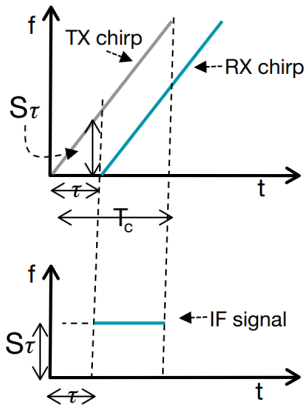


Figure 2.3: IF signal is obtained by combining TX and RX chirps [30].

with α as the n-th signal amplitude, $t_n = t - nT_c$, and the time delay τ given by

$$\tau = \frac{2d}{c} \tag{2.3}$$

Multiplying the transmitted and received chirps and low-pass filtering (mixer+LPF) results in the intermediate-frequency (IF) signal

$$s_{\mathrm{IF}}(t) = \mathrm{LPF}\{\, s_{\mathrm{tx}}(t)\, s_{\mathrm{rx}}(t)\,\} \approx \frac{\alpha}{2}\cos\big(2\pi f_{\mathrm{b}}\, t_n + \phi\big), \tag{2.4}$$

where

$$f_{\mathrm{b}} = S\,\tau = \frac{2S}{c}\,d = \frac{2B}{cT_c}\,d \tag{2.5}$$

is the beat frequency and $\phi = 2\pi f_D T_c$ is the phase offset introduced by Doppler between successive chirps, with f_D the Doppler frequency. The IF signal is a single tone whose constant frequency equals the instantaneous-frequency separation of the two chirps. Geometrically, in the $f-t$ (frequency-time) plane this is the vertical distance between the TX and delayed RX ramps, which is constant and equal to $S\tau$, as Figure 2.3 shows.

The data received by an FMCW MIMO receiver are naturally arranged along three axes: fast-time (IF samples within a chirp), slow-time (the sequence of chirps), and spatial channel (antennas). This forms a third-order tensor known as the radar data cube (Figure 2.4). The three axes contain essential information used for localization: fast-time encodes range (round-trip delay), slow-time encodes Doppler (radial velocity) and the spatial channel encodes angle of arrival (AoA).

In the presence of multiple targets, the received waveform is a superposition of

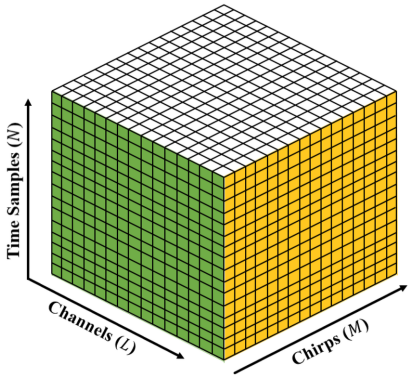


Figure 2.4: Radar data cube [53].

RX chirps, each delayed by an amount of time proportional to the corresponding range. These chirps translate to multiple constant-frequency IF tones, $f_{\mathrm{b},i} = S\tau_i$, where τ_i corresponds to the delay associated with the i-th target. In order to separate the different IF tones a Fourier transform processing must be applied, resulting in a frequency spectrum with a separate peak for each different tone. The peaks denote the presence of an object at a specific distance.

2.1.1 Range resolution

When two objects move closer they become hard to distinguish as two separate entities by the radar. To overcome this problem, as the Fourier transform theory states, the length of the IF signal must be increased. This increases the range resolution, defined as the ability to distinguish between different objects. In order to detect the range of the object each reflected chirp is processed through fast Fourier transform (FFT) (i.e., range-FFT).
From (2.4) the range follows directly:

$$d = \frac{c}{2S}\,\mathrm{f_b}. \tag{2.6}$$

2.1.2 Velocity resolution

The range-FFT peaks align across chirps in location, while their phases are distinct. The phase difference between chirps corresponds to a motion in the object.

$$\Delta\phi \;=\; \frac{4\pi\, v\, T_\mathrm{c}}{\lambda}. \tag{2.7}$$

The velocity can be derived as:

$$v \;=\; \frac{\lambda\,\Delta\phi}{4\pi\, T_\mathrm{c}}. \tag{2.8}$$

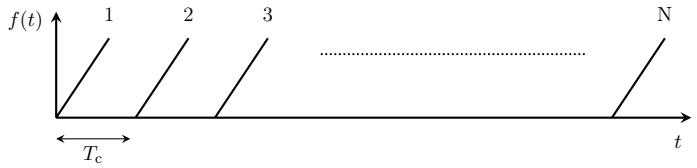


Figure 2.5: A frame includes N chirps each spaced by T_c seconds.

The measurement is unambiguous only if $|\Delta\phi| < \pi$. From the equation above:

$$v < \frac{\lambda}{4T_c}, \qquad v_{max} = \frac{\lambda}{4T_c}. \tag{2.9}$$

This defines the maximum relative speed that can be estimated from two chirps separated by T_c.

The phase-comparison approach for velocity estimation becomes ineffective when multiple targets with different velocities are moving at the same range from the radar. The result is two reflected chirps with identical IFs collapsing into a single peak in the range-FFT. The velocity estimation method used in this case implies the transmission of a set of N equally spaced chirps , a chip frame (Figure 2.5), from the radar.

A range-FFT processing is applied on the reflected set of chirps producing a set of N phasors (identically located peaks each with a different phase). In order to resolve the two objects, a Doppler-FFT needs to be performed on the N phasors. Considering f_1 and f_2 as the Doppler frequencies associated with the phase increments for each chirp. The corresponding speeds follow from:

$$v_1 = \frac{\lambda f_1}{2T_c}, \qquad v_2 = \frac{\lambda f_2}{2T_c}. \tag{2.10}$$

Applying the discrete Fourier transforms resolvability condition $\Delta f > 1/T_f$ to the relation (2.7), the velocity resolution is derived as:

$$v_{res} = \frac{\lambda}{2T_f}. \tag{2.11}$$

which improves with the increasing of the frame period T_f, defined as $T_f = 2T_c$. By computing a discrete Fourier transform (DFT) along fast-time (range processing) and then a DFT along slow-time (Doppler processing) on each range bin, the RD map is obtained; it represents signal energy as a function of range and radial velocity.

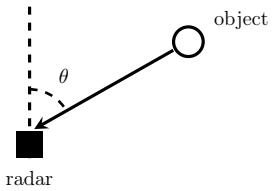


Figure 2.6: The angle of arrival θ is the one between the normal to the antenna plane and the incident signal direction.

2.1.3 Angle resolution

Small variations in target distance produce phase changes at the peak of the range-FFT or Doppler-FFT. This allows to conduct angular estimation: with at least two RX antennas, the differential path length from the object to each of the antennas produces distinct peak phases, allowing AoA estimation. The AoA is defined as the angle of a reflected signal with the horizontal plane, as Figure 2.6 shows.
The phase change can be derived as

$$\Delta\phi \;=\; \frac{2\pi\,\Delta d}{\lambda} \tag{2.12}$$

where $\Delta d = lsin(\theta)$. Therefore, given the measured $\Delta\phi$, the AoA θ is estimated via

$$\theta \;=\; \sin^{-1}\!\left(\frac{\lambda\,\Delta\Phi}{2\pi\,l}\right) \tag{2.13}$$

Coupling range processing with a spatial transform across the virtual array results in the RA map, which represents energy as a function of range and AoA.

Chapter 3

Deep learning architectures: from perceptrons to CNN

Device–free people counting from radar measurements calls for methods that can cope with variability in rooms, trajectories, and multipath. Hand–crafted signal features are often sensitive to operating conditions: what works in one configuration may degrade in another. This motivates a data–driven strategy in which models are fit directly from examples rather than derived from fixed rules. Within this perspective, ML provides a principled way to learn the mapping from radar observations to people counts directly from examples. However, traditional ML pipelines still rely on expert-designed features that often fail to generalize across domains.

Among ML methods, DL has become a highly flexible and general-purpose paradigm for learning from data. Deep models learn hierarchical representations rather than relying on hand-crafted features: early layers capture simple patterns that later on will be combined into increasingly abstract concepts used for prediction and decision making. Formally, a deep network is a parametric function built by stacking linear transformations with nonlinear activations. In this context, a linear transformation is a mapping that forms weighted combinations of the input components, while an activation function is a fixed rule applied to each component of the result. If all activations were linear, a stack of layers would still behave like a single linear mapping and could not model curved or interacting relationships. Using nonlinear activations breaks this limitation and allows the network to bend decision surfaces, represent interactions among variables and approximate complex input–output relationships.

Artificial neural networks (NNs) were originally inspired by biological neural processing: abstract units mimic neurons that aggregate weighted synaptic inputs

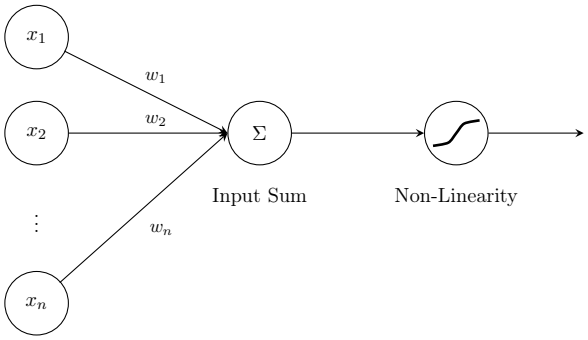


Figure 3.1: The perceptron.

and apply an activation, approximating the idea of signal transmission across connections; today these models are implemented as mathematical constructs.

3.1 Multi-layer perceptron architectures

A NN is a parametric function built from layers; the way those layers are arranged, its architecture, shapes how information flows and which relationships the model can learn.
Each layer is built from simple units, called neurons, that combine inputs, add a bias and apply a non-linear rule; stacking many such layers lets the network move from raw signals to progressively richer representations, reducing the need for hand-crafted features.

3.1.1 Perceptron and deep neural models

The simplest neural model is the perceptron, a single artificial neuron that takes a feature vector as input, computes a linear combination with a bias term

$$z = \mathbf{x}^\top \mathbf{w} + b \tag{3.1}$$

where $\top$ denotes the transpose, and applies a nonlinear activation g to produce the output

$$y = g(z). \tag{3.2}$$

The weights w_n determine how strongly each input influences the output changing its tilt, the bias b shifts the response along the activation axis changing its position, and the nonlinearity z enables the model to represent non-linear relationships. Multiple perceptrons arranged side by side, sharing the same inputs but with different weights, form a dense layer with multiple outputs. When the activation function g is chosen as a step function, the unit operates as a threshold

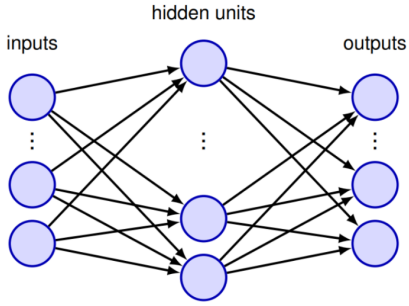
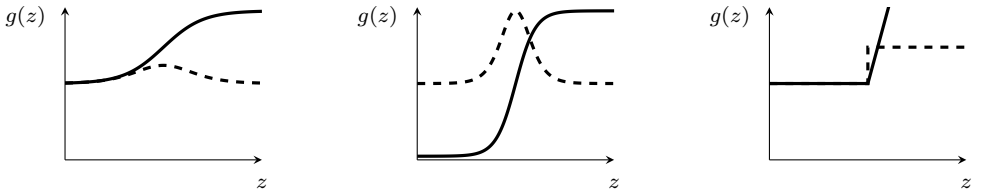


Figure 3.2: Multi-layer fully connected NN [45].



(a) Sigmoid Function (b) Hyperbolic tangent (c) Rectified Linear Unit

Figure 3.3: Comparison of three activation functions (solid) and their derivatives (dashed).

element, turning on only when the pre-activation (3.1) surpasses the threshold and staying off otherwise. Early implementations had multiple processing layers, however only the final stage had trainable weights, classifying the model as a single-layer one. Figure 3.1 illustrates the perceptron's activation function: the step function. The shift from the perceptron to multi-layer networks was driven by the need to model non-linear decision boundaries and learn intermediate features: introducing hidden layers with non-linear activations lets the model compose simple features into more complex ones. The intermediate nodes are labeled as hidden because their values are not observed within the training data; since the information flow is strictly from inputs to outputs the multi-layer perceptrons (MLPs) are called feed-forward NNs. Figure 3.2 shows the schematic of a multi-layer perceptron with fully connected layers. The step function used for the single-layer perceptron is replaced by continuous and differentiable activations and the model is paired with a differentiable loss function that quantifies prediction error.

The loss quantifies how wrong a prediction is; for classification it is typically (softmax) cross-entropy, which measures the distance between the predicted probabilities and the true label distribution, while for regression a common choice is mean-squared error.

The empirical loss measures the total loss over the entire dataset, and is defined

by

$$J(\mathbf{W}) \;=\; \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}\Big(f(\mathbf{x}^{(i)};\,\mathbf{W}),\,y^{(i)}\Big). \tag{3.3}$$

where $\mathcal{L}(\hat{y}, y)$ is the per-example loss function, with $\hat{y} = f(\mathbf{x}^{(i)}; \mathbf{W})$, that measures the discrepancy between the prediction and the true label. The accuracy of the model is therefore defined as the fraction of correctly classified examples, i.e., the percentage of predictions that match the true labels. Under these conditions, the network admits gradient-based optimization: training is the process of fitting a NN's parameters to labeled data by minimizing the empirical loss.

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} J(\mathbf{W}). \tag{3.4}$$

The training method used for these networks is based on backpropagation: it computes the loss gradients (derivatives of the loss function) for all parameters, which are randomly initialized, and then updates them iteratively in the negative-gradient direction. In practice, weights are propagated from the output layer back through the hidden layers after a forward evaluation.

During training, backpropagation relies on the local slope of the activation function to determine weight updates. The slope corresponds to the derivative of the activation function.

Few common activation functions and their corresponding derivatives are illustrated in Figure 3.3.

The error propagated backward is scaled by the local slope, indicating for earlier layers whether to increase or decrease each weight and by how much. When the slope is close to zero, learning slows; when it is sufficiently large and stable, updates are more effective.

A critical parameter is the learning rate: if it is too small, learning is slow and can get stuck; if too large, training becomes unstable or diverges. Modern practice therefore employs adaptive methods or learning-rate schedules that adjust the step size during training to improve convergence.

While MLPs process fixed-size inputs under a purely feed-forward scheme, many real signals are sequential and order-dependent. A recurrent neural network (RNN) addresses sequential structure.

In contrast to MLPs, RNNs pass information forward in time while feeding part of the state back into the model, creating a dependency between the current state and the previous one. The training process is based on a process called back-propagation through time, which uses a stochastic gradient descent unrolling the computation over the sequence. Each output unit produces a probability distribu-

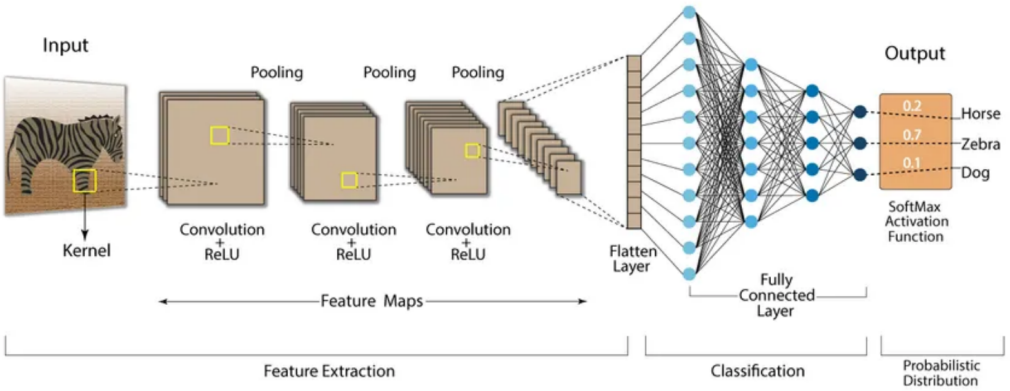


Figure 3.4: CNN architecture [51].

tion via a softmax function, which turns a vector of real numbers into a probability distribution over classes, and the loss is measured using cross-entropy; the overall loss function is the sum of the losses calculated for each unit. The main limitations of these architectures are based on the vanishing or exploding gradients over long dependencies.

3.1.2 Convolutional architectures

A CNN targets spatiotemporal structure. Nearby pixels that compose an image, for example, are highly correlated and some patterns repeat across the 2-D grid. CNNs approach grants sparse local connectivity, allowing each unit to have access only at its receptive field rather than the full input, and parameter sharing, applying the same set of filter weights, the *kernel*, across all locations; this allows the reduction of the number of learnable parameters. Each image is a set of numbers, with each number representing a pixel. In the convolutional layer, a kernel is slided over the image to produce feature maps; afterwards, a non-linearity is introduced and an optional normalization to stabilize training. A pooling layer can be applied in order to shrink the feature maps by combining two consecutive convolutional layers in order to reduce computation, overfitting and add tolerance to changes in scale and position. Figure 3.5 shows a side-by-side comparison: in the overfitted case the curve chases noise in the training samples, achieving very low training error while failing to generalize, whereas a good fit captures the underlying trend. Higher layers combine lower-level features, such as edges, into increasingly abstract patterns, while the fully connected layers maps features to task outputs.

Figure 3.4 shows a CNN architecture highlighting feature extraction and the subsequent classification block. CNNs are trained with backpropagation: gradients

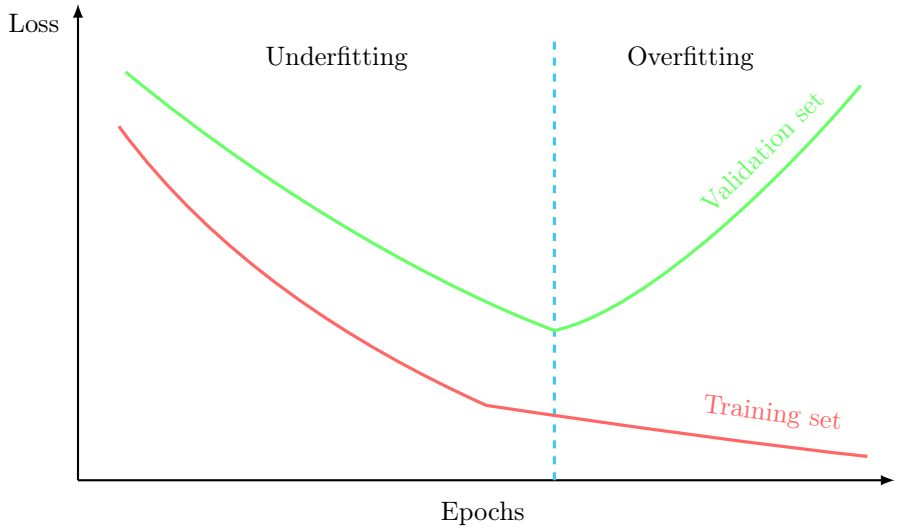


Figure 3.5: The overfitting problem in NN training.

propagate through the shared convolution kernels and the non-linear layers so that all parameters are updated jointly. Within convolutional blocks, ReLU-family activations are the common choice for fast and stable learning, while the output activation remains task-dependent. rectified linear unit (ReLU) leaves positive inputs unchanged and sets negative inputs to zero as defined by (3.5), adding nonlinearity and promoting sparse activations.

$$f(x) = \max(0, x). \tag{3.5}$$

Beyond this baseline, several general practices improve performance and generalization. Batch normalization normalizes intermediate activations to stabilize and accelerate training; dropout randomly zeroes activations to curb overfitting; data augmentation expands the effective dataset to improve robustness. To improve convergence various training strategies can be used. Among them, early stopping is particularly effective: it monitors validation performance and halts training when improvement stalls for a preset patience window, typically restoring the best-performing weights. This prevents overfitting, shortens training time, and yields a model that generalizes better.

3.2 CNN design for sensing via FMCW MIMO radar

Analyzing RDs and RAs outputs requires a neural architecture able to detect and learn patterns in grid-structured arrays with strong spatial correlations: as previously noted, CNNs are particularly appropriate for this purpose.

Treating RD and RA maps as two-dimensional images allows convolutional filters to detect target-specific structures and features, suppressing both static clutter and multipath interference. Deeper layers aggregate low-level structure into more abstract radar representations; the prediction layers then convert these features into the estimated number of targets.

Training proceedes with mini-batches, random initialization and gradient-based optimization; several optimization algorithms can be employed. Among common choices, Adam is widely used because it dynamically adapts the learning rate for each parameter, accelerating convergence and making it particularly effective when dealing with complex and noisy data, such as FMCW radar signals.

Chapter 4

Target Counting based on RD map processing via CNN

4.1 Experimentation

A mmWave FMCW MIMO radar was employed for all experiments. The unit operated as an FMCW system with four transmit and sixteen receive antennas and linear chirps sweeping from 77 GHz to 78 GHz (Figure 4.1). The radar streamed baseband samples to a host PC, where standard processing produced the range–Doppler (RD) and range–angle (RA) representations used by the counting model. Data were collected in an indoor 6 x 4 m^2 corridor at the Department of Engineering, University of Ferrara, shown in Figure 4.2. Six datasets were acquired, each containing 2000 measurements (frames). The i-th dataset ($i = 0, \ldots, 5$) contains scenes with exactly i moving people, from empty corridor to five subjects walking. These datasets were used to build the training/validation splits for the CNN. After training, the model was deployed in the same corridor. The radar streamed RD frames to the CNN model, which returned the estimated number of targets for each incoming frame; the results were visualized in real time together with the current RD/RA maps on a website.

Counting is treated as a multi-class classification problem on single range–Doppler frames. Each RD map is treated as a single–channel grayscale image and normalized per frame to $[0, 1]$ via min–max scaling. The CNN produces a probability for each count in $\{0, \ldots, K\}$, where K is the maximum number of targets considered ($K = 5$ in our experiments); the predicted count is the class with the highest probability.

The CNN used is composed as follows:

Convolutional 2D input layer with 64 feature maps, a 3×3 kernel, stride 1×1,

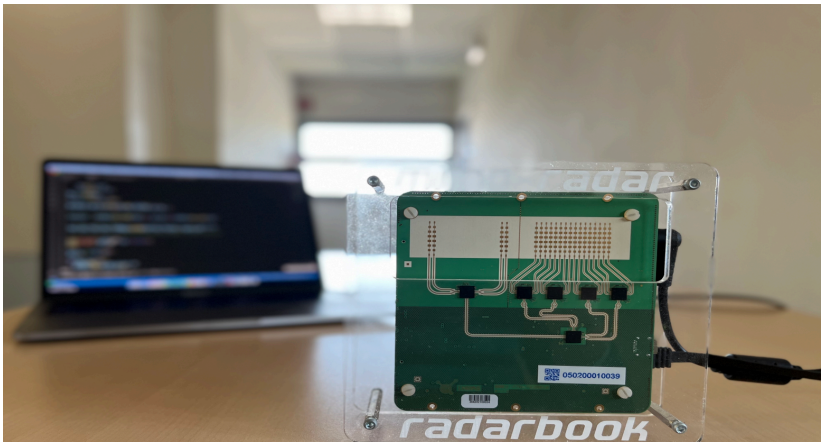


Figure 4.1: The 77 GHz MIMO radar employed in the experimentation

Figure 4.2: Measurement campaign environment.

valid padding, and ReLU activation.

Max pooling layer with a 2×2 kernel and stride 2×2.

Convolutional 2D layer with 32 feature maps, a 3×3 kernel, stride 1×1, valid padding, and ReLU activation.

Max pooling layer with a 2×2 kernel and stride 2×2.

Flatten layer that converts the feature maps into a one-dimensional vector.

Fully connected layer with 16 units and ReLU activation.

Output layer with $K+1$ units and softmax activation, producing the class probabilities for the admissible counts.

The model is implemented in TensorFlow, an open-source DL framework. Training uses the Adam optimizer with a learning rate of 10^{-2} with sparse categorical cross-entropy as the loss function. Validation accuracy is tracked throughout training, and the checkpoint with the best validation performance is retained. For evaluation, the predicted count is taken as the class with the highest soft-

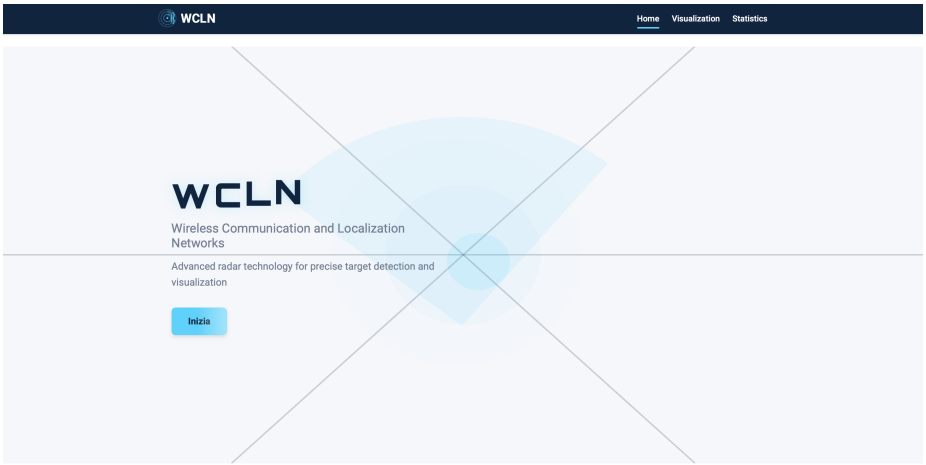


Figure 4.3: GUI Homepage

max probability. For deployment in the live pipeline, the trained Keras model is converted to TensorFlow Lite (TFLite). The conversion applies the default post-training optimizations to reduce model size and improve CPU inference speed. The exported TFLite model is then loaded by the inference service, which processes each incoming RD frame and returns the predicted count in real time.

4.2 Graphical user interface

A lightweight GUI was developed to operate the system and visualize results in real time. The application follows a client–server design: an Angular frontend renders the views and handles user interaction, while an Express.js backend handles data transmission between the radar and the CNN inference and streams updates to the browser.

On the server side, Express exposes a small set of REST endpoints for configuration and historical queries, mantaining a real-time channel (WebSocket) to push fresh data. The graphical interface is built around a streaming pipeline that separates acquisition, inference, and visualization, so that each stage can operate independently while exchanging data with low latency. The radar continuously acquires data and, for each frame, computes the RD map and a parallel RA view. The two representations are serialized into compact binary messages together with a timestamp.

Frames are distributed through a messaging layer. Two mechanisms are used in combination: (i) a publish/subscribe channel for broadcast delivery to multiple consumers and (ii) point-to-point sockets for direct requests, where a client can retrieve the most recent RD, RA, or the current target count. A dedicated

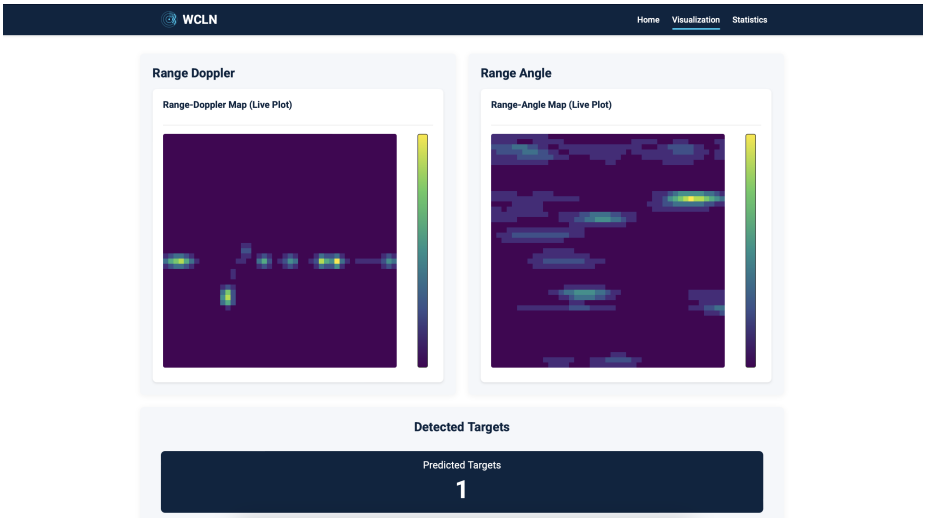
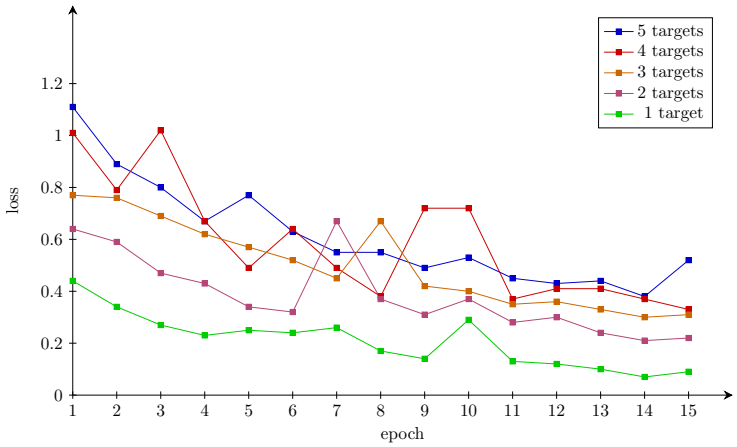


Figure 4.4: GUI visualization page

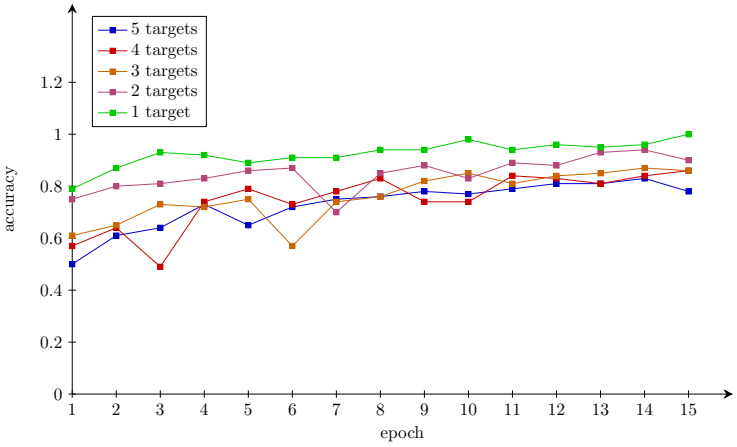
Figure 4.5: GUI page showing counting statistics real-time.

service subscribes to the stream, deserializes each frame, performs the required normalization, and executes the counting model on the RD map. The service maintains a short rolling history of predictions to support trend plots and exposes the latest outputs through socket endpoints. The result is a steady flow of tuples $\{\mathrm{RD}, \mathrm{RA}, \mathrm{count}, t\}$ ready for visualization. A web server bridges the inference service to the client application using a persistent real-time channel (WebSocket) and updates are pushed as soon as a new frame is available.

The frontend is a single-page application organized into three views. The Home page (Figure 4.3) presents the project context and provides access to the live session. The Visualization page shows, at each instant t, the range–Doppler map, the range–angle map, and the predicted number of targets, updating continuously as



(a) Training loss



(b) Training accuracy

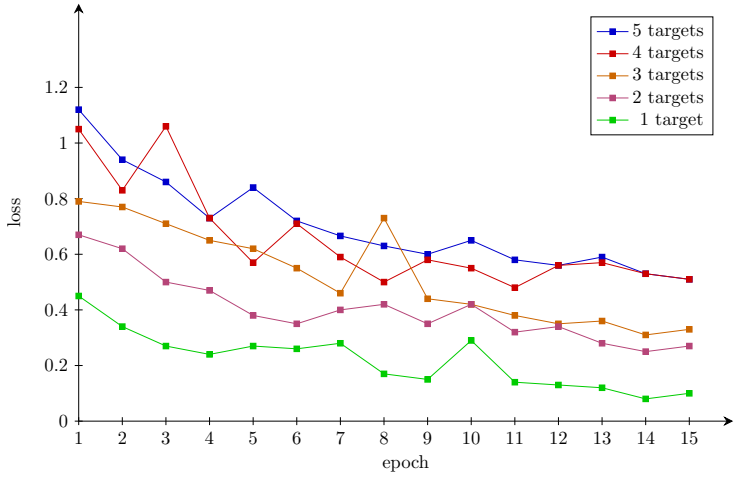
Figure 4.6: Training loss and accuracy

new frames arrive, as shown in Figure 4.4. The Statistics page aggregates the live stream to display target count over time and simple indicators such as total detections, peak value and average rate (Figure 4.5). Angular services subscribe to the server stream and propagate updates to the components, ensuring low-latency rendering without page reloads.

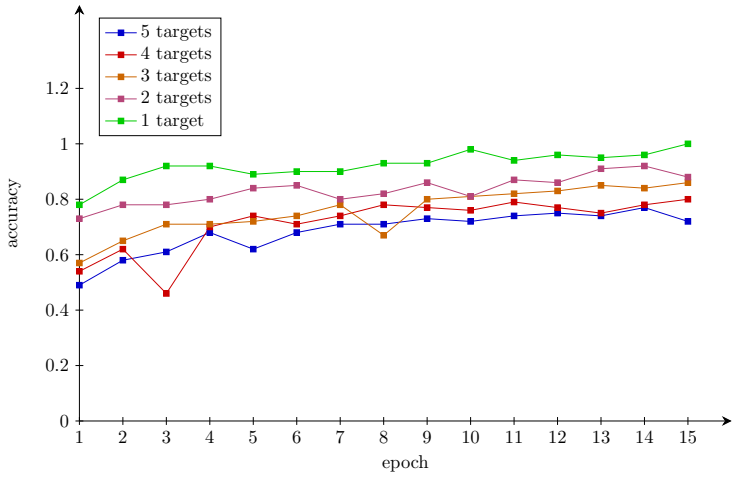
4.3 Results

Each dataset used for the CNN training contains a total of 2000 measurements (i.e. frames). Each frame is a 49×55 range–Doppler matrix (49 range bins, 55 Doppler bins) acquired before clutter mitigation.

The dataset was split evenly into training and validation subsets, with 50% as-



(a) Validation loss



(b) Validation accuracy

Figure 4.7: Validation loss and accuracy

signed to training and 50% to validation.

The learning dynamics followed a clear pattern. Figure 4.6 reports the training accuracy and loss over 15 epochs, while Figure 4.7 shows the corresponding validation curves. Most of the gain occurred at the beginning of training: accuracy rose rapidly while the loss dropped sharply, after which both curves progressed more gradually and approached a steady trend. The class ranking by occupancy is the same for both training and validation: scenes with fewer targets are learned more easily, whereas higher occupancies remain harder.

On the held-out validation set, performance was strongest at low occupancies. Accuracy reached 100% for a single target and remained above 85% for two targets.

For three and four targets accuracy ranged from 80% to 85%, with the five-target class peaking at 77%.
The absence of a late rise in validation loss or drop in validation accuracy argues against overfitting.

Chapter 5

Conclusion

Accurate sensing is fundamental in next-generation wireless networks, with device-free people counting being an essential application. Reliable indoor approaches required methods able to estimate how many people are present without using cameras or asking users to carry devices, thus avoiding issues of occlusions, poor visibility, and privacy. A device-free counting approach has been developed relying on reflected radio frequency (RF) signals collected by an off-the-shelf mmWave FMCW MIMO radar. In particular, an approach was developed in which a CNN estimates the number of people by classifying range–Doppler (RD) maps produced by the radar.

The experimental study was conducted with an off-the-shelf mmWave FMCW MIMO radar in indoor environments, including non-line-of-sight (NLOS) configurations with furniture and typical sources of clutter. Accuracy and loss metrics were used to quantify performance, and attention was paid to robustness under multipath and environmental interference.

On the held-out validation set, accuracy exceeded 75 % for counts up to five targets and peaked at 95 % for the two-target case. Training performance mirrored these values with consistently higher accuracy in each class.

Bibliography

[1] A. Conti, S. Mazuelas, S. Bartoletti, W. C. Lindsey, and M. Z. Win, "Soft information for localization-of-things," *Proc. IEEE*, vol. 107, no. 11, pp. 2240–2264, Sep. 2019.

[2] S. Bartoletti, S. Mazuelas, A. Conti, and M. Z. Win, "Efficient localization via soft information with generic sensing measurements," *IEEE Trans. Wireless Commun.*, vol. 24, no. 7, pp. 5400–5414, Jun. 2025.

[3] T. Padmashree and S. S. Nayak, "5G technology for e-health," in *Proc. Int. Conf. IoT in Social, Mobile, Analytics and Cloud*, Palladam, India, Nov. 2020, pp. 211–216.

[4] A. Acemoglu *et al.*, "5G robotic telesurgery: remote transoral laser microsurgeries on a cadaver," *IEEE Trans. Med. Robot. Bionics*, vol. 2, no. 4, pp. 511–518, Oct. 2020.

[5] J. Suomalainen, J. Julku, M. Vehkaperä, and H. Posti, "Securing public safety communications on commercial and tactical 5G networks: A survey and future research directions," *IEEE Open J. Commun. Soc.*, vol. 2, pp. 1590–1615, Jul. 2021.

[6] Z. Qi, A. Lahuerta-Lavieja, J. Li, and K. K. Nagalapur, "Deployable networks for public safety in 5G and beyond: A coverage and interference study," in *Proc. IEEE 5G World Forum*, Montreal, QC, Canada, Oct. 2021, pp. 346–351.

[7] K. Yu, L. Tan, L. Lin, X. Cheng, Z. Yi, and T. Sato, "Deep-learning-empowered breast cancer auxiliary diagnosis for 5Gb remote e-health," *IEEE Wireless Commun.*, vol. 28, no. 3, pp. 54–61, Jul. 2021.

[8] Z. Wang, Z. Liu, Y. Shen, A. Conti, and M. Z. Win, "Location awareness in beyond 5G networks via reconfigurable intelligent surfaces," *IEEE J. Sel. Areas Commun.*, vol. 40, no. 7, pp. 2011–2025, Mar. 2022.

[9] K. Shafique, B. A. Khawaja, F. Sabir, S. Qazi, and M. Mustaqim, "Internet of things (IoT) for next-generation smart systems: A review of current challenges, future trends and prospects for emerging 5G-IoT scenarios," *IEEE Access*, vol. 8, pp. 23 022–23 040, Jan. 2020.

[10] S. H. Alsamhi, F. A. Almalki, F. Afghah, A. Hawbani, A. V. Shvetsov, B. Lee, and H. Song, "Drones' edge intelligence over smart environments in b5G: Blockchain and federated learning synergy," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 1, pp. 295–312, Dec. 2022.

[11] M. Z. Win, Z. Wang, Z. Liu, Y. Shen, and A. Conti, "Location awareness via intelligent surfaces: A path toward holographic NLN," *IEEE Veh. Technol. Mag.*, vol. 17, no. 2, pp. 37–45, May 2022.

[12] B. Vo *et al.*, "Multitarget tracking," in *Wiley Encyclopedia of Electrical and Electronics Engineering*. Wiley, Sep. 2015.

[13] S. Sun and Y. D. Zhang, "4D automotive radar sensing for autonomous vehicles: A sparsity-oriented approach," *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 4, pp. 879–891, May 2021.

[14] M. Zhou, Z. Yang, P. Chu, and J. Zhou, "An interference recognition embedded people counting method exploiting occlusion information," in *Proc. IEEE Int. Workshop Radio Freq. Antenna Technologies*, Shenzhen, China, May 2025, pp. 266–271.

[15] Z. Tian, C. Ye, and Y. Jin, "Device-free indoor tracking via joint estimation of DFS and AoA using CSI amplitude," in *Proc. Int. Conf. Microw. Millimeter Wave Technol.*, Nanjing, China, May 2021, pp. 1–3.

[16] K. Witrisal *et al.*, "High-accuracy localization for assisted living: 5G systems will turn multipath channels from foe to friend," *IEEE Signal Process. Mag.*, vol. 33, no. 2, pp. 59–70, Mar. 2016.

[17] S. Bartoletti, A. Conti, A. Giorgetti, and M. Z. Win, "Sensor radar networks for indoor tracking," *IEEE Wireless Communi. Lett.*, vol. 3, no. 2, pp. 157–160, Jan. 2014.

[18] S. Bartoletti, A. Conti, and M. Z. Win, "Device-free counting via OFDM signals of opportunity," in *Proc. IEEE Int. Conf. Commun. Workshops*, Kansas City, MO, USA, May 2018, pp. 1–5.

[19] ——, "Device-free counting via wideband signals," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1163–1174, Mar. 2017.

[20] C. A. Gómez-Vega, M. Z. Win, and A. Conti, "UWB localization-of-things via soft information: Network experimentation in indoor environment," in *Proc. IEEE Int. Conference Acoust., Speech Signal Process.*, Rhodes Island, Greece, Jun. 2023, pp. 1–5.

[21] A. Conti, M. Guerra, D. Dardari, N. Decarli, and M. Z. Win, "Network experimentation for cooperative localization," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 2, pp. 467–475, Jan. 2012.

[22] C. A. Gómez-Vega, O. Y. Kolawole, M. Hunukumbure, T. Mach, M. Z. Win, and A. Conti, "Device-free localization: Outdoor 5G experimentation at mmwaves," *IEEE Commun. Lett.*, vol. 27, no. 9, pp. 2353–2357, Jun. 2023.

[23] G. Torsoli, M. Z. Win, and A. Conti, "Beyond 5G localization at mmWaves in 3GPP urban scenarios with blockage intelligence," in *Proc. IEEE Position, Location Navigation Symp.*, Monterey, CA, USA, Apr. 2023, pp. 354–359.

[24] S. Perez-Gamboa, Q. Sun, and Y. Zhang, "Improved sensor based human activity recognition via hybrid convolutional and recurrent neural networks," in *Proc. Int. Symp. Inertial Sensors Syst.*, Kailua-Kona, HI, USA, May 2021, pp. 1–4.

[25] M. M. Hossain Shuvo, N. Ahmed, K. Nouduri, and K. Palaniappan, "A hybrid approach for human activity recognition with support vector machine and 1D convolutional neural network," in *Proc. App. Imagery Pattern Recognit. Workshop*, Washington DC, DC, USA, May 2020, pp. 1–5.

[26] N. Archana and K. Hareesh, "Real-time human activity recognition using ResNet and 3D convolutional neural networks," in *Proc. Int. Conf. Advances Comput., Commun., Embedded Secure Syst.*, Ernakulam, India, Oct. 2021, pp. 173–177.

[27] S. Bartoletti, A. Giorgetti, M. Z. Win, and A. Conti, "Blind selection of representative observations for sensor radar networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 4, pp. 1388–1400, Jan. 2015.

[28] M. Chiani, A. Giorgetti, and E. Paolini, "Sensor radar for object tracking," *Proc. IEEE*, vol. 106, no. 6, pp. 1022–1041, May 2018.

[29] A. Vaccari, M. Z. Win, and A. Conti, "Tracking and identification of targets via mmwave mimo radar," in *Proc. IEEE Military Commun. Conf.*, Washington, DC, USA, Oct. 2024, pp. 336–341.

[30] C. Iovescu and S. Rao, "The fundamentals of millimeter wave radar sensors," Texas Instruments, Application Note Rev. A, Jul. 2020.

[31] S. Rao, "Mimo radar," Texas Instruments, Application Report SWRA554A, Jul. 2018.

[32] A. Vaccari, "Methods for action recognition via mm-Wave MIMO radar," Master's thesis, Università di Ferrara, 2022.

[33] D. Yuan, H.-Y. Lin, J. Widmer, and M. Hollick, "Optimal and approximation algorithms for joint routing and scheduling in millimeter-wave cellular networks," *IEEE/ACM Trans. Netw.*, vol. 28, no. 5, pp. 2188–2202, Jul. 2020.

[34] M.-S. Kim, T. Ropitault, S. Lee, N. Golmie, H. Assasa, and J. Widmer, "A link quality estimation-based beamforming training protocol for IEEE 802.11ay MU-MIMO communications," *IEEE Trans. Commun.*, vol. 69, no. 1, pp. 634–648, Oct. 2021.

[35] E. Arribas *et al.*, "Optimizing mmWave wireless backhaul scheduling," *IEEE Trans. Mobile Comput.*, vol. 19, no. 10, pp. 2409–2428, Jun. 2020.

[36] A. Shastri *et al.*, "A review of millimeter wave device-based localization and device-free sensing technologies and applications," *IEEE Commun. Surv. Tut.*, vol. 24, no. 3, pp. 1708–1749, May 2022.

[37] S. Bartoletti, Z. Liu, M. Z. Win, and A. Conti, "Device-free localization of multiple targets in cluttered environments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 5, pp. 3906–3923, Nov. 2022.

[38] C. Fiandrino, H. Assasa, P. Casari, and J. Widmer, "Scaling millimeter-wave networks to dense deployments and dynamic environments," *Proc. IEEE*, vol. 107, no. 4, pp. 732–745, Feb. 2019.

[39] M. Scalabrin, G. Bielsa, A. Loch, M. Rossi, and J. Widmer, "Machine learning based network analysis using millimeter-wave narrow-band energy traces," *IEEE Trans. Mobile Comput.*, vol. 19, no. 5, pp. 1138–1155, Mar. 2020.

[40] R. Zhang, X. Jing, S. Wu, C. Jiang, J. Mu, and F. R. Yu, "Device-free wireless sensing for human detection: The deep learning perspective," *IEEE Internet Things J.*, vol. 8, no. 4, pp. 2517–2539, Sep. 2021.

[41] I. Nirmal, A. Khamis, M. Hassan, W. Hu, and X. Zhu, "Deep learning for radio-based human sensing: Recent advances and future directions," *IEEE Commun. Surv. Tut.*, vol. 23, no. 2, pp. 995–1019, Feb. 2021.

[42] M. Zhao *et al.*, "Through-wall human pose estimation using radio signals," in *Proc. IEEE Conf. Comp. Vision Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 7356–7365.

[43] T. Li, L. Fan, M. Zhao, Y. Liu, and D. Katabi, "Making the invisible visible: Action recognition through walls and occlusions," in *Proc. IEEE Int. Conf. Comput. Vision*, Seoul, Korea, Nov. 2019, pp. 872–881.

[44] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, "A survey of deep neural network architectures and their applications," *Neurocomputing*, vol. 234, pp. 11–26, Jul. 2017.

[45] C. M. Bishop and H. Bishop, *Deep Learning: Foundations and Concepts*. Springer, 2024.

[46] R. M. Schmidt, "Recurrent neural networks (RNNs): A gentle introduction and overview," arXiv preprint arXiv:1912.05911, 2019.

[47] S. Singh, K. Kumar, and S. Gupta, "Machine learning based indoor localization techniques for wireless sensor networks," in *Porc. Int. Conf. Advances Comput., Commun. Control Netw.*, Greater Noida, India, Dec. 2020, pp. 373–380.

[48] F. Morselli, S. Bartoletti, S. Mazuelas, M. Win, and A. Conti, "Crowd-centric counting via unsupervised learning," in *Proc. Int. Conf. Commun. Workshops*, Shanghai, China, May 2019, pp. 1–6.

[49] M. Gareis, C. Carlowitz, and M. Vossiek, "A MIMO UHF-RFID SAR 3D locating system for autonomous inventory robots," in *Proc. IEEE Int. Conf. Microw. Intell. Mobility*, Linz, Austria, Nov. 2020, pp. 1–4.

[50] C. A. Gómez-Vega, M. Z. Win, and A. Conti, "Neural network based node prioritization for efficient localization," in *Proc. IEEE Veh. Technol. Conf.*, Florence, Italy, Jun. 2023, pp. 1–5.

[51] I. Purwono *et al.*, "Understanding of convolutional neural network (CNN): A review," *International J. of Robot. and Control Syst.*, vol. 2, pp. 739–748, 01 2023.

[52] N. Shahriar, "What is convolutional neural network — CNN (deep learning)," *LinkedIn Pulse*. Available: https://www.linkedin.com/pulse/what-convolutional-neural-network-cnn-deep-learning-nafiz-shahriar/, Apr. 2023.

[53] S. w. Kang, M. h. Jang, and S. Lee, "Autoencoder-based target detection in automotive MIMO FMCW radar system," *Sensors*, vol. 22, no. 15, pp. 1–18, Jul. 2022, article no. 5552.