Information Fusion 91 (2023) 173–186

Contents lists available at ScienceDirect

Information Fusion

journal homepage: www.elsevier.com/locate/inffus

Full Length Article

A novel deep learning approach using blurring image techniques for Bluetooth-based indoor localisation

Reewos Talla-Chumpitaz [b,1], Manuel Castillo-Cara [a,1,*], Luis Orozco-Barbosa [c], Raúl García-Castro [a]

[a] *Universidad Politécnica de Madrid, Madrid, Spain*
[b] *Universidad Nacional de Ingenieria, Lima, Peru*
[c] *Albacete Research Institute of Informatics, Universidad de Castilla-La Mancha, Albacete, Spain*

ARTICLE INFO

Keywords:
Indoor positioning
Fingerprinting localisation
Metaheuristic algorithm optimisation
Image blurring technique
Convolutional Neural Network
Image generation

ABSTRACT

The growing interest in the use of IoT technologies has generated the development of numerous and diverse applications. Many of the services provided by the applications are based on knowledge of the localisation and profile of the end user. Thus, the present work aims to develop a system for indoor localisation prediction using Bluetooth-based fingerprinting using Convolutional Neural Networks (CNN). For this purpose, a novel technique was developed that simulates the diffusion behaviour of the wireless signal by transforming tidy data into images. For this transformation, we implemented the technique used in painting known as blurring, simulating the diffusion of the signal spectrum. Our proposal also includes the use and a comparative analysis of two dimensional reduction algorithms, PCA and *t*-SNE. Finally, an evolutionary algorithm was implemented to configure and optimise our solution with the combination of different transmission power levels. The results reported in this work present an accuracy of close to 94%, which clearly shows the great potential of this novel technique in the development of more accurate indoor localisation systems.

1. Introduction

The development of wireless networks and devices equipped with multiple sensors [1], and their connection to storage centres and data processing over the Internet, has led to the implementation of the Internet of Things (IoT) [2]. The growing interest in the use of IoT technologies has led to the development of numerous and diverse applications [3,4]. The proper functioning of these applications requires the control of huge flows of data generated by mobile devices to the intelligent decision-making centres deployed in the cloud [5]. Many of the services provided by the applications are based on knowledge of the localisation and profile of the end user [6].

Therefore, the main effort focuses on the main components of the development of localisation systems for IoT applications [7]: (i) the analysis of the signal that transmitters emit to the receiver and how to adequately treat the generated data [8,9]; and (ii) algorithms for processing wireless signals that allow indoor localisation [10,11].

Regarding the first point, the design and development of wireless-based fingerprint localisation techniques presents major challenges, as indoor propagation of signals is highly sensitive to the multipath fading effect [12,13]. It is also widely recognised that the capabilities of the surveying devices will play a major role in the quantity, quality, and time of the effort invested to produce valuable Received Signal Strength Indicator (RSSI) fingerprints [14,15].

On the second point, one of the main areas of research is the characterisation of the behaviour of wireless signals [9,16]. The results of different investigations show that the use of classification algorithms in the signal characterisation process is capable of greatly improving the indoor localisation mechanisms based on performance [13,17].

Since there are many learning methods to solve this problem, this work will focus on the development of a novel prediction technique using Convolutional Neural Networks (CNN) [18]. However, since the input format for a CNN is images, this input does not match the format of data obtained by IoT devices, which are tidy data (also called tabular data), based on RSSI [16,19]. One of the main contributions of this work lies in the transposing of tidy data [20] into images with a novel preprocessing technique that represents the behaviour of a wireless signal in the image construction process [19,21].

[*] Corresponding author.
E-mail addresses: reewos.talla.c@uni.pe (R. Talla-Chumpitaz), manuel.castillo.cara@upm.es (M. Castillo-Cara), luis.orozco@uclm.es (L. Orozco-Barbosa), r.garcia@upm.es (R. García-Castro).
[1] This authors have contributed equally to this work.

https://doi.org/10.1016/j.inffus.2022.10.011
Received 18 May 2022; Received in revised form 28 September 2022; Accepted 10 October 2022
Available online 17 October 2022

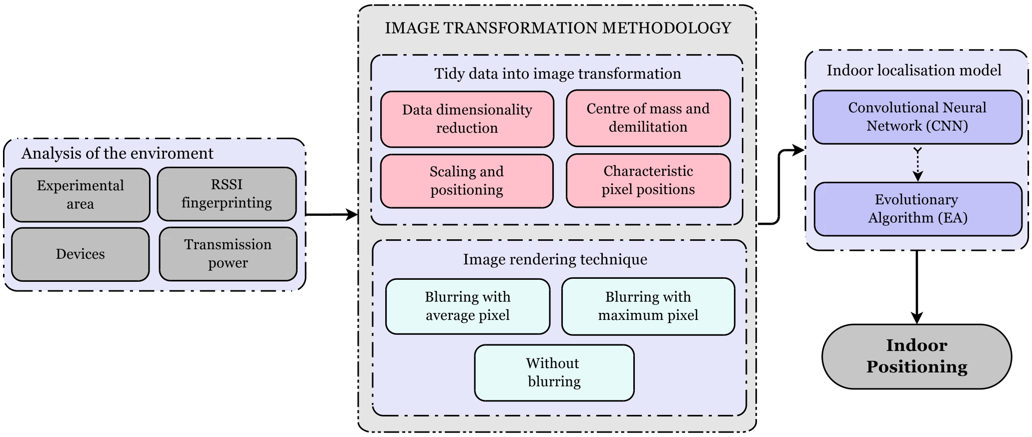


IMAGE TRANSFORMATION METHODOLOGY
Tidy data into image transformation
Data dimensionality reduction
Centre of mass and demilitation
Scaling and positioning
Characteristic pixel positions
Image rendering technique
Blurring with average pixel
Blurring with maximum pixel
Without blurring
Analysis of the enviroment
Experimental area
RSSI fingerprinting
Devices
Transmission power
Indoor localisation model
Convolutional Neural Network (CNN)
Evolutionary Algorithm (EA)
Indoor Positioning

Fig. 1. Overall schema proposal.

In other words, our proposal starts by transforming the tidy data, consisting of RSSI samples spatially distributed in a target area, into images to enable the implementation of the classification process to be performed by a CNN [19,22]. The proposal uses the methods and mechanisms of the article [23] as the starting point for developing the overall processing schema. In the first step of our proposed schema, two techniques of dimensional reduction are used: t-SNE [24] and PCA [25] for transforming the tidy data into images. Then, an evolutionary algorithm is then used to find the best combination of results using different transmission powers (here on in referred to as TxPower) by the transmitter devices. The structure of the overall processing schema introduced in this work to improve the indoor localisation process accuracy can be summarised as follows:

- Use of two dimensionality reduction algorithms to convert tidy data into images, spatial distributed BLE RSSI readings, namely the t-SNE and PCA algorithms.
- Use of the blurring painting technique to emulate signal degradation in the image. This and the previous step make for the main contributions of herein, allowing the classification to be performed using a CNN.
- Use of a two-branch convolutional neural network for Bluetooth indoor localisation.
- Optimisation of the BLE indoor localisation accuracy using metaheuristic algorithms.

The remainder of this paper is organised as follows. Section 2 reviews the recent Bluetooth localisation literature in two sections: indoor localisation fingerprint and techniques developed to construct images from tidy data. Section 3 specifies our indoor environment and the devices used as transmitters and receivers, depicted in Fig. 1 with blocks called “Analysis of the Environment”. Subsequently, Section 4 “Image Transformation Methodology” describes, step-by-step, the details of the mechanisms and techniques used to create images from the tidy data collected. This section also covers the preprocessing methodology and the importance of the blurring painting technique to emulate the behaviour of the signal. Section 5 presents our first set of results using symmetric and asymmetric TxPower configurations of the beacons with a parallel CNN and evolutionary algorithm, represented in “Indoor Localisation Model” block. Finally, Section 6 presents our conclusions and future work directions.

2. Related work

This section introduces the main terminology and an analysis of related works for a better understanding and development of the proposed solution.

2.1. Indoor localisation fingerprint

A large number of RSSI-based localisation studies have been reported in the literature. Among the most popular wireless technologies, Bluetooth Low Energy (BLE) has been the object of many studies. BLE devices (here on in referred to as beacon) use 40 2-MHz channels to broadcast information. The protocol uses short message duration to reduce power consumption. To avoid interference between devices, since Wi-Fi and BLE use the same 2.4 GHz band, BLE4.0 uses channels labelled 37 (2402 MHz), 38 (2426 MHz), and 39 (2480 MHz) [9,26]. Various authors have developed beacon performance testing for positioning [7], leading to the publication of recommendations on the placement and density of the beacons, transmission intervals, fingerprint space layout and sampling intervals in physical learning spaces for sustainable eLearning environments.

However, indoor localisation systems based on a single technology have exhibited limitations [4,27], such as the drift of inertial navigation and fluctuation of the received strength of the Bluetooth signal, making them unable to provide reliable positioning. In order to overcome the shortcomings of a single source of information, various authors have developed multi-sensor solutions by benefiting from the use of the different sensors currently encountered in most mobile devices, together with algorithms or other sources of information, such as particle filters, indoor space or other various wireless technologies maps [8].

Kriz et al. [28] reported a localisation experiment using a set of Wi-Fi access points (AP) accompanied by BLE devices. Their localisation mechanism was based on Weighted-Nearest Neighbours in the Signal Space algorithm. The main objectives of their study were to improve indoor localisation accuracy by introducing the use of beacons and deploying a system that constantly updates the RSSI levels reported by the mobile devices (receivers). During their experiments, they varied two parameters of the beacons: the duration of the RSSI signal scan, and the density. However, throughout all their experiments, the power of the beacons was set to their maximum value.

A more in-depth study on localisation with beacons and prior to the one described in this article is that described in [9]. The authors carried out an exhaustive study on two main areas to be taken into account when performing indoor localisation based on wireless signals. First, they explored mitigating the negative effect of multipath fading on localisation by using different transmission powers. Second, they developed a methodology to determine the best combination of transmission powers used by the different beacons. Their solution was based on the use of metaheuristics. Their study showed a great improvement in terms of localisation accuracy with respect to previous results reported in the literature.

Regarding the data processing methods being used in the development of indoor fingerprinting-based indoor localisation environments,

Table 1
Comparison of the present work with other existing works in the literature. The last row (*This work*) shows the characteristics and results of the work presented in this article.

Article	Method	Algorithm	Accuracy	Mean error	Test area
[34]	Triangulation	Signal Coverage Density Method	50%	8.39 m	13 m × 15 m
[26]	Fingerprinting	*k*-Nearest Neighbour	95%	2.6 m	50 m × 15 m
[35]	Fingerprinting	*k*-Nearest Neighbour	92.5%	2 m	17.5 m × 9.6 m
[36]	Fingerprinting	Weighted Pass Loss	–	1.39 m	19 m × 16.2 m
[37]	Triangulation	Single Direction Outlier Removal	–	1.5 m	5.4 m × 8 m
[38]	Proximity	Kalman/Particle Filter	–	0.708 m	7 m × 6 m
[39]	Fingerprinting	*k*-Nearest Neighbour	90%	2.6 m	17 m × 3 m
[40]	Proximity	Gaussian Process Regression	–	5.2 m	18 m × 12 m
[41]	Fingerprinting	*k*-Nearest Neighbour	–	1.27 m	7.28 m × 7.24 m
[7]	Fingerprinting	*k*-Nearest Neighbour	89.65%	2.23 m	16 m × 14.5 m
[7]	Fingerprinting	*k*-Nearest Neighbour	87.59%	2.17 m	13 m × 11.7 m
[7]	Fingerprinting	*k*-Nearest Neighbour	92.19%	1.89 m	15 m × 19 m
[7]	Fingerprinting	*k*-Nearest Neighbour	90.07%	1.1 m	9.5 m × 7 m
[42]	Fingerprinting	Linear Discriminant Analysis	79.34%	–	10 m × 10 m
[8]	Proximity	Particle Filter	–	1.32 m	52.5 m × 52.5 m
[43]	Fingerprinting	Deep Reinforcement Learning	–	4.3 m	60 m × 54.86 m
[44]	Fingerprinting	Multilayer Perceptron	–	0.78 m	11 m × 12 m
[45]	Fingerprinting	Long Short-Term Memory	91.1%	–	7 m × 7 m
[46]	Fingerprinting	Multilayer Perceptron	–	0.45 m	31 m × 21 m
[47]	Proximity	Path Loss Model	–	0.844 m	10.8 m × 7.3 m
[47]	Proximity	Path Loss Model	–	0.661 m	5.6 m × 5.9 m
[33]	Fingerprinting	*k*-Nearest Neighbour	77.89%	0.51 m	9.3 m × 6.3 m
[9]	Fingerprinting	Gradient Boost Machine	86%	0.384 m	9.3 m × 6.3 m
This work	Fingerprinting	Convolutional Neural Network	93.04%	0.148 m	9.3 m × 6.3 m

involve three main types of algorithmic methods [3]: probabilistic methods, such as particle filter [8], deterministic methods, such as Support Vector Machine [9], and neural networks, such as CNN [16]. Most studies currently focus on the use of CNNs to improve the accuracy and processing of indoor localisation solutions [29].

2.2. Bluetooth indoor localisation: Techniques and results

Hint to reviewers: This entire subsection is new.

In the past few years, we have witnessed the emergence of the so-called wireless-based indoor localisation mechanisms, which aim to respond to the ever-increasing demands for indoor positioning requirements [30,31]. Under this paradigm, designers and developers are being challenged to design and develop systems incorporating a large number of data sources and intelligent data-intensive processing and visualisation mechanisms capable of meeting the location requirements of end users and applications.

Among the large number of emerging applications, many organisations and research centres are focusing their efforts on the design and development of robust-indoor wireless localisation mechanisms [32]. Depending on the wireless technology, the use of a technique or algorithm may be more suitable or feasible with respect to others [30, 32].

This work focuses on the development of Bluetooth localisation mechanisms through fingerprint classification processes. Table 1 lists the most significant works reported in the literature. For each method, the table shows the methods, algorithms and the performance results obtained in a controlled test area. The last row (*This work*) reports the characteristics and results of the work reported herein. Moreover, the penultimate two rows report the results of two of our previous works [9,33] making use of the same RSSI database and test area.

As shown in Table 1, a large number of previous works based on fingerprinting methods use the *k*-Nearest Neighbour (*k*-NN) algorithm. Although *k*-NN is the most widely used algorithm, the vast majority of works on obtaining numeric, i.e. tidy data, from transmitters focus on classic machine learning algorithms for the classification process [42]. It can also be observed that the test area is usually a small environment; this is done in order to study the impact of the signal on the classification process in a fully controlled environment [7]. An example of this is the prior data clean-up processing required before the classification process, where auto-encoders have been shown to be an effective process for this purpose [35]. The use of a controlled environment allows us to focus on the use of novel techniques and, above all, to see the impact of important elements in the classification process, such as the position of the transmitters [37], the direction of the signal [7], transmission channels [40], mitigating the multipath fading effect [9,33] or the use of multiple sensors from smartphones [46], among others.

As seen in the table, our work shows better performance results, Accuracy and Mean Error metrics that all of the previous works reported in the literature. Likewise, comparing *This work* with our previous works [9,33] for which we have used the data collected in the same experimental area, we observe a significant improvement in both metrics. The proposal introduced in this paper makes use of an unstudied method for Bluetooth-based indoor localisation, consisting on the transformation of tabular data into images to be able to make use of CNN-based classification process. This differs from previous works in which the use of CNNs for indoor localisation using wireless signals was been based on the combination of obtaining images through cameras with wireless data [48].

2.3. CNN for indoor localisation fingerprint

One of the main aspects when developing CNNs is to be able to have the data reinforced in a preprocessing phase [19]. In terms of the classification technique implemented when using images in the context and classification process, CNNs predominate. The works developed in terms of the use of a neural network architecture are varied, but the vast majority focus on having an input image obtained in the sample collection phase [49].

A different research work from the classic one of collecting samples from images and classifying them with CNNs is that of Hsieh et al. [50]. The authors present an approach based on CNNs using transmission channel quality metrics, mainly RSSI and Channel State Information (CSI). By testing different methods, they conclude that with a one-dimensional CNN with CSI they obtain better metrics than other models by reducing complexity.

More specifically, we find another more sophisticated research work on image integration [48]. The authors propose a new localisation algorithm in which wireless signals and images are combined. The novelty of this work lies in first obtaining a coarse-grained estimation based on the visualisation of wireless signals by fingerprinting.

Additionally, the authors perform a matching process to determine the correspondences between two- and three-dimensional pixels based on the images collected. Based on their results, the method through the combination of visual and wireless data significantly improves the localisation metrics and robustness.

Therefore, the present work was based on the technique used in [23], in which the authors focus on developing a method for the transformation of structured to unstructured data. In this case, it would be from tidy data to images for prediction using CNNs. More specifically, in this research, the model created with five datasets (1 cancer, 1 vowel, 1 text, and 2 artificial) are evaluated. On the other hand, for the transformation, dimensional reduction algorithms are used with the purpose of obtaining coordinates generated for each feature in the dataset. From the generated coordinates, it generates an image pattern where a pixel represents the feature and the value of each pixel would be given by the value of the feature. Finally, the generated images would go through a CNN with parallel layers, obtaining the final classification result. The accuracy obtained is higher than 90% for most of the datasets used in this work.

As can be seen, the use of different machine/deep learning techniques is widely developed, especially in the use of known algorithms. Therefore, the present work describes a novel method for developing an indoor localisation technique in which structured data (tidy data) are transformed into unstructured data (images) to use a neural network architecture. For the transformation of the data, the methodology described in the article [23] used, while, for the signal specifications, we used the methodology described in [9].

3. Background: Devices and methods

In the design and development of BLE localisation systems, it is widely recognised that the capabilities of the surveying devices will play a major role in the quantity, quality, and time of effort invested to produce valuable RSSI fingerprints. Details about the area, transmitter/receiver, survey campaigns, Multipath Fading (MPF) and intraday signal attenuation were analysed in previous works. Hence, we discuss additional information about BLE4.0 signal characterisation and conduct an in-depth analysis about the impact of different materials/structures on RSSI [9,15].

3.1. Experimental area

Our experiments were carried out in the laboratory of our research institute. We placed four beacons at each of the four corners of a rectangular area 9.3 m×6.3 m. The fifth beacon was placed in the middle of one of the longest edges of the room. Fig. 2 depicts the experimental area in which the five beacons are labelled as “Be07”, “Be08”, “Be09”, “Be10” and “Be11”. We divided the experimental area into 15 sectors of 1 m^2, each separated by a guard distance of 0.5 m. A 1.5 m-wide strip was left around the experimental area. This arrangement will allow us to better differentiate the RSSI level of the joint sectors when reporting our results. Measurements were performed by placing the mobile device at the centre of each on of the 15 sectors, as described below. The shortest distance between a beacon and a receiver was limited to 1.5 m. Fig. 3 shows four views taken from each of the four corners of the laboratory. As shown in the figure, we placed beacons “Be10” and “Be11” in front of a window, Figs. 3(d) and 3(b), respectively, while all the other beacons were placed in front of the opposite plasterboard wall. We further noted that beacon “Be08” was placed at the left edge of the entrance door, close to the corridor with a glass wall (Fig. 3(c)).

Table 2
Structure of the RSSI for Tx04 obtained in the data acquisition phase. An instance of the RSSI for the different sectors (in total 15) is shown as an example.

Be07	Be08	Be09	Be10	Be11	Sector
−65	−61	−74	−73	−67	1
−60	−57	−83	−62	−69	2
−66	−70	−78	−63	−73	3
…	…	…	…	…	…
−58	−66	−71	−73	−69	14
−60	−62	−73	−69	−57	15

3.2. Transmitter and receiver devices

For this experiment, JAALEE beacon devices were used [51]. According to the specifications of the five beacons used in our experiments, they may operate at one of eight different TxPower levels. Following the specifications, the TxPower levels are labelled in consecutive order from highest to lowest level as TxPower = 0 × 01, TxPower = 0 × 02, …, TxPower = 0 × 08 (here on in referred to as Tx01, …, Tx08) (ultra wide range TxPower: 4 dBm to −40 dBm), although Tx07 and Tx08 were discarded since they did not adequately cover the signal spectrum in the entire area. During our experiments, we conducted several measurement campaigns by fixing the TxPower level of all beacons at the beginning of each campaign. Additionally, all measurements were performed under line-of-sight conditions.

As receiver, we used a Raspberry Pi equipped with a USB BLE4.0 antenna [52], hereinafter referred to as BLE4.0 antenna. Finally, the dataset can be downloaded from [53].

3.3. Acquisition phase data format

In the data acquisition phase, the RSSI obtained from the five beacons will be denoted as the features, X. The output value, i.e., target, is the sector in which the person is located, denoted as Y. Note that we have from Tx01 to Tx06. Thus, a sample of the input data for Tx04 can be seen in Table 2. Note that the features, X, are the RSSI values called from “Be07” to “Be11”, and the target, Y, is the feature called “Sector”. It can be seen that the input data is in tidy data format.

Since the main purpose is to convert the tidy data into images for the use of a CNN, a partition of the samples must be made, i.e., for train/test. The partition ratio would be 70% for training and 30% for testing.

3.4. Convolutional neural network architecture

For the development of a CNN, the structure of the incoming and outgoing data must be understood. The resulting images from the previous process, when imported, must be converted into a data matrix of size $p \times p \times 1$ (one channel since the images are given in black and white), which will be the input to the CNN. On the other hand, we have the test data, which are related to the sector being a classification problem, where each sample is converted into a vector of size C. Note that C is the number of classes, in this case, 15, referring to the sectors of the experimental area.

In this context, Fig. 4 shows the architecture of the parallel CNN developed in this work. The result of this CNN is maximised the results by testing empirically different configurations of it. We will work with a CNN with 2 parallel branches, having the same input for both and four blocks. Each block consists of 4 different layers, with all 4 blocks having the same configuration. Thus, the initial layer of each block is a convolutional layer followed by a batch normalisation layer that resets and rescales the results obtained by the previous layer. The next is a ReLU activation layer and the last is a pooling layer. The difference between one branch and the other is the size of the kernel filter and the type of pooling. Maximising the results yielded a configuration of a filter of size $(3, 3)$ and MaxPooling for the first branch and, for

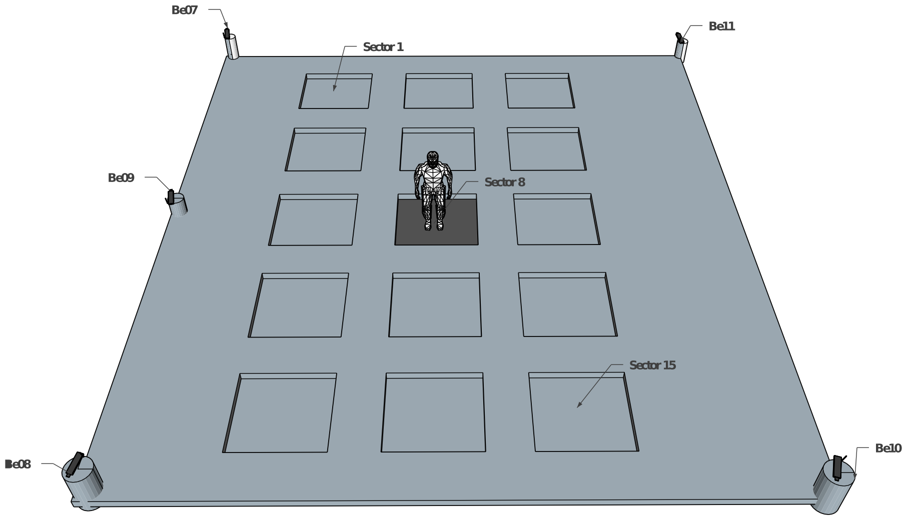
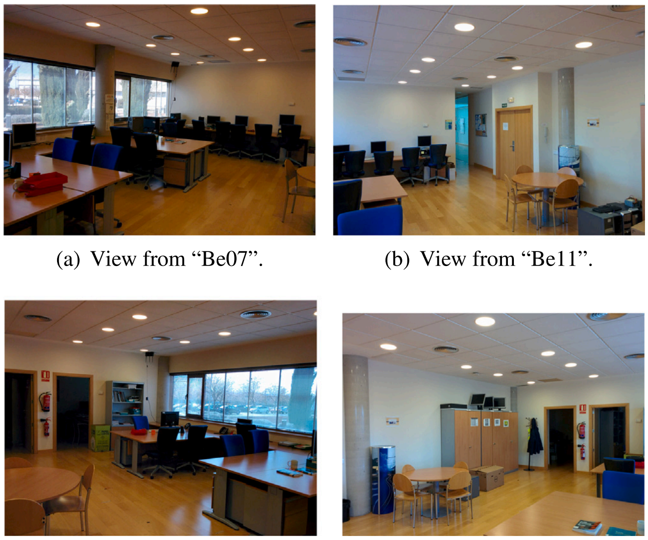


Fig. 2. Beacon indoor experimental area setup.


(a) View from “Be07”.

(b) View from “Be11”.

(c) View from “Be08”.

(d) View from “Be10”.

Fig. 3. Pictures of each of the four corners of the laboratory.

the second branch, $(5,5)$ and AveragePooling. For both branches, the condition that the number of filters for the convolution layers is 16, 32, 64 and 64 for each block and in that order is satisfied.

Therefore, the two branches converge into one (concatenate). The Flatten layer follows, which arranges the data, flattening the matrices created from the previous layers. Subsequently, 1 dense layer of 256 neurones with ReLU activation and 3 dense layers with Sigmoid activation (128, 64 and 32 neurones, respectively) are used. The last layer of the CNN has Softmax activation, which aims to provide values (probabilities) to classify the image, and has 15 neurons due to the number of classes (sectors) in the experimental area.

4. Methodology on image transformation

As discussed in the aforementioned sections, one of the main contributions of this work is to develop and formalise a methodology for converting tidy data into images. This section describes this methodology and the mathematical formalisation that justifies this novel mechanism for indoor localisation fingerprint.

4.1. Tidy data into image transformation

One of the first translations to be performed is to convert the RSSI obtained for each Bluetooth device in the same image, i.e., the corresponding sample (row) in the sampling phase (see Table 2). To develop tidy data into image transformation, the data processing shown in the pipeline detailed in Fig. 5 must be performed. As can be seen, tidy data is converted into image data through a two-dimensional space in X and Y. This translation to two-dimensional space will allow us to build an image of characteristic pixels for each sample of the dataset, i.e., each data sample is converted to a two-dimensional space. Note that, for data transformation, we used the basic methodology (although adapted to this particular research) set out in [23].

As seen from Fig. 5, the pipeline to convert each tidy data sample into a two-dimensional space consists of five main processing tasks:

1. Data dimensionality reduction: The initial matrix is transposed. Note that each feature is represented with a different colour per comprehension, although the resulting figure will consist of two channels, i.e., black and white. Therefore, this task makes use of a dimensionality reduction algorithm. In our case, we explored two such algorithms: PCA and t-SNE.
2. Centre of mass and delimitation: Having obtained the coordinates, the centre of gravity of the points is determined and the area is subsequently delimited.
3. Scaling and pixel positions. The matrix is transposed, scaled and the values are rounded to integer value positions.
4. Characteristic pixel positions: The values obtained would be the positions of the characteristic pixels for the creation of the image pattern.

As mentioned, this work is based on the methodology presented by Sharma et al. [23] for the transformation of tidy data into image. However, the authors do not provide a formal specification of the steps to obtain the positions of the characteristic pixels from the initial tidy data. In the following paragraphs, not only do we provide a precise description of the steps of our solution, but we also fully analyse, in Section 4.2, the significance of the tidy data into image patterns obtained when applying the two dimensionality reduction algorithms, i.e., PCA and t-SNE, and blurring painting technique in the image generation process. Our analysis fully identifies the shortcomings of the t-SNE algorithm and provides a mechanism to avoid the overlapping of characteristic pixels.

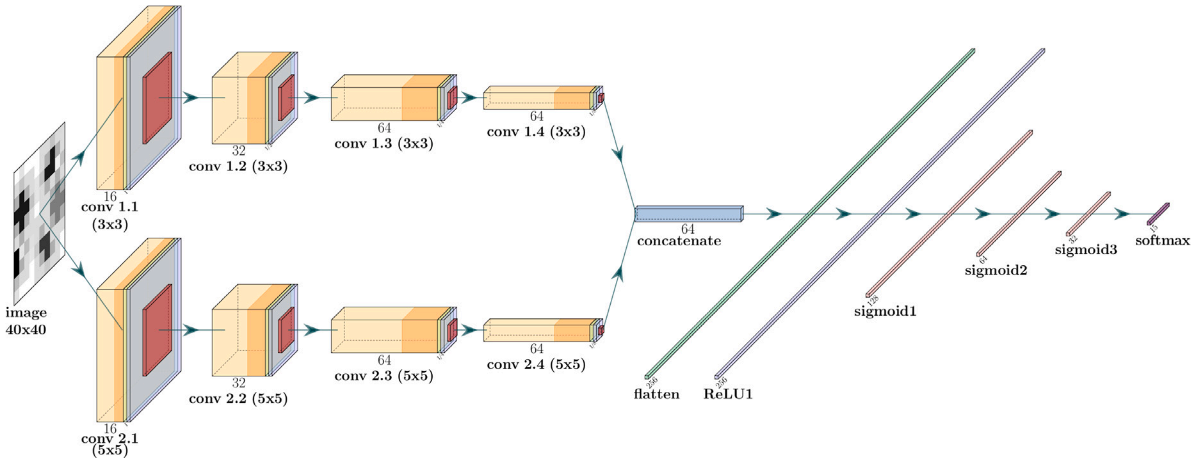


Fig. 4. Convolutional neuronal network architecture.

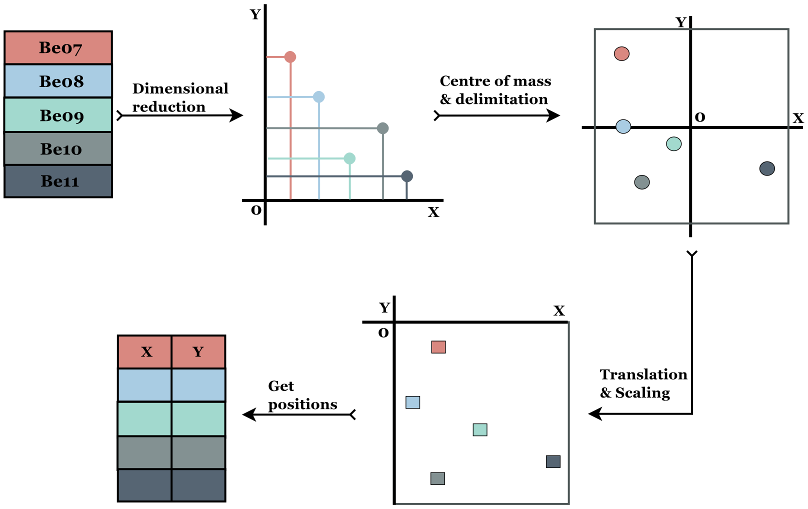


Fig. 5. Process for obtaining the feature coordinates from the transpose matrix. It can be seen that the tidy data are converted into a two-dimension matrix through the different phases with their respective techniques, i.e., delimitation, translation, and so on.

Dimensional reduction

This first task, not shown in Fig. 5, takes as input the initial data matrix,denoted by $A \in \mathbb{R}^{\mathrm{m}\times\mathrm{n}}$, where m is the number of samples (rows) and n is the number of features (columns).

$$A = \begin{bmatrix} a_{11} & \dots & a_{1n} \\ a_{21} & \dots & a_{2n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{bmatrix} \tag{1}$$

The matrix A is transposed, denoted by $A^t \in \mathbb{R}^{\mathrm{n}\times\mathrm{m}}$:

$$A^t = \begin{bmatrix} a_{11} & a_{21} & \dots & a_{m1} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \dots & a_{mn} \end{bmatrix} \tag{2}$$

The transposed matrix A^t is then reduced by applying one of the two dimension reduction algorithms (t-SNE or PCA). The resulting matrix, denoted as ARD, is reduced from m columns to two columns. Therefore, the resulting matrix $B \in \mathbb{R}^{\mathrm{n}\times 2}$ can be simply denoted as:

$$ARD(A^t, 2) = B = \begin{bmatrix} b_{11} & b_{12} \\ \vdots & \vdots \\ b_{n1} & b_{n2} \end{bmatrix} \tag{3}$$

In the case that the use of the t-SNE algorithm be preferred, we should take care to avoid the overlapping of characteristic pixels. This condition may arise due to the stochastic nature, and results in the loss of information, see Section 4.2 for an in-depth analysis. Accordingly in this case, the following three processing tasks should be performed when using the t-SNE algorithm.

- Given the matrix A^t, a new matrix $D \in \mathbb{R}^{\mathrm{kn}\times 2}$ is created by overlapping k times the matrix A^t:

$$D = \begin{bmatrix} A^t_1 \\ \vdots \\ A^t_k \end{bmatrix} = \begin{bmatrix} a^{(1)}_{11} & a^{(1)}_{21} & \dots & a^{(1)}_{m1} \\ \vdots & \vdots & \ddots & \vdots \\ a^{(1)}_{1n} & a^{(1)}_{2n} & \dots & a^{(1)}_{mn} \\ a^{(2)}_{11} & a^{(2)}_{21} & \dots & a^{(2)}_{m1} \\ \vdots & \vdots & \ddots & \vdots \\ a^{(k)}_{1n} & a^{(k)}_{2n} & \dots & a^{(k)}_{mn} \end{bmatrix} \tag{4}$$

- Dimension reduction is performed as specified by (3):

$$t-\mathrm{SNE}(D, 2) = D' = \begin{bmatrix} b'_{11} & b'_{12} \\ \vdots & \vdots \\ b'_{n1} & b'_{n2} \\ b'_{(n+1)1} & b'_{(n+1)2} \\ \vdots & \vdots \\ b'_{(kn)1} & b'_{(kn)2} \end{bmatrix} \tag{5}$$

- A linear transformation is performed $T : \mathbb{R}^{\mathrm{kn}\times 2} \rightarrow \mathbb{R}^{\mathrm{n}\times 2}$, such that T is defined as follows:

$$T(D') = \left[b_{ij} = \frac{\sum_{h=1}^{k} b'_{((h-1)n+i)j}}{k} \right] = B \tag{6}$$

As seen from the above steps, the procedure basically consists of replicating the matrix A_t k times before applying the *t-SNE* algorithm, and then reducing the number of rows of kn to n by averaging over $(h-1)\cdot n + i$ rows, where $h \in [1,k]$, for each $i \in [1,n]$

Centre of mass and delimitation

In this second task, each pair of elements in a row is considered as a point p_i where $i \in [1,n]$, corresponds to the coordinates X and Y the elements of the first and second column, respectively:

$$p_i = (b_{i1}, b_{i2}) \rightarrow P = \{p_1, \dots, p_n\} \tag{7}$$

We proceed to find the centre of mass of the points p_i, represented as p_{MC}, where all points have the same mass.

$$p_{MC} = (\frac{\sum_{i=1}^{n} b_{i1}}{n}, \frac{\sum_{i=1}^{n} b_{i2}}{n}) \tag{8}$$

Then, the maximum integer distance from the farthest point to the centre of mass, denoted d_{max}, is calculated:

$$d_{max} = \lceil \max_{\{i\in[1,n]\}} \|\vec{p_i} - \vec{p_{MC}}\| \rceil \tag{9}$$

where $\|\|$ is the norm function, $\lceil\rceil$ is the ceil function, and $p^{\rightarrow}$ is a vector pointing from the origin of the coordinates to point p.

The section in the Cartesian plane containing the characteristic points $p_i, \forall i \in [1,n]$ is delimited, denoted as S.

$$S = \{(x,y) : x,y \in \mathbb{R}/x \in [p_{MCx} - d_{max}, p_{MCx} + d_{max}],\\ y \in [p_{MCy} - d_{max}, p_{MCy} + d_{max}]\} \tag{10}$$

A linear transformation $F_1 : S \rightarrow T$ is performed, such that F_1 is defined as follows:

$$F_1(x,y) = (x - p_{MCx} + d_{max}, y - p_{MCy} - d_{max}); \forall (x,y) \in S \tag{11}$$

Using this maximum distance d_{max}, the square is bounded, so each side of the image square will be a size of $2 \times d_{max}$; where the set T is defined as follows:

$$T = \{(x,y) : x,y \in \mathbb{R}/x \in [0, 2\times d_{max}], y \in [-2\times d_{max}, 0]\} \tag{12}$$

In our case, the translation was made to the fourth quadrant because of its similarity to the order of the matrices (upper left position as origin). In this case, the values of the coordinates are negative, but are solved in later steps (see expressions (13) and (14)).

Scaling and pixel positions

This task starts with a linear transformation, the scaled coordinates of the features, $F_2 : T \rightarrow U$ defined as follows:

$$F_2(x,y) = (\lfloor x \times \frac{pixel-1}{2\times d_{max}} \rceil, \lfloor |y \times \frac{pixel-1}{2\times d_{max}}| \rceil); \forall (x,y) \in T \tag{13}$$

where $pixel$ is the number of pixels of the resulting image and $||$ and $\lfloor\rceil$ are the absolute value and rounding functions (solely to integer value positions), respectively.

In turn, the set U is defined as:

$$U = \{(x,y) : x,y \in \mathbb{N}/x \in [0, pixel-1], y \in [0, pixel-1]\} \tag{14}$$

Consequently, having characteristic points p_i, we transform them into characteristic pixel positions q_i:

$$q_i = F_2(F_1(p_i)) \rightarrow Q = \{q_1, \dots, q_n\} \tag{15}$$

Then the values of the initial matrix A, should be scaled over the interval $[0,255]$. This is done by first obtaining the global minimum (see Eq. (16)) and global maximum (see Eq. (17)) of matrix A:

$$a_{min} = \min_{\{i\in[1,m];j\in[1,n]\}} \{a_{ij}/a_{ij} \in A\} \tag{16}$$

$$a_{max} = \max_{\{i\in[1,m];j\in[1,n]\}} \{a_{ij}/a_{ij} \in A\} \tag{17}$$

We proceed to normalise the values of the matrix A by means of the linear transformation $G_1 : R^{m\times n} \rightarrow R^{m\times n}$, being defined as follows:

$$G_1(X) = \left[\frac{x_{ij} - a_{min}}{a_{max} - a_{min}} \right] \in R^{m\times n} = [y_{ij}] \in R^{m\times n} = Y \tag{18}$$

the scale over the $[0,255]$ is finally completed by means of the linear transformation $G_2 : R^{m\times n} \rightarrow R^{m\times n}$, defined as follows:

$$G_2(Y) = [\lfloor 255 \times y_{ij} \rceil] \in R^{m\times n} = [z_{ij}] \in R^{m\times n} = Z \tag{19}$$

In summary, by applying the linear transformations to matrix A:

$$C = G_2(G_1(A)) \tag{20}$$

where $C \in R^{mtimesn}$ is the matrix containing the final values, after normalisation and scaling to $[0,1]$, to be placed at the corresponding positions in the characteristic pixels. For this specific procedure, the `MinMaxScaler` class of the *scikit-learn* Python library was used.

Characteristic pixel positions

With the purpose of ordering the data to feed the CNN, a linear transformation $H : R^{m\times n} \times R^{n\times 2} \rightarrow R^{m\times pixel\times pixel}$ must be performed, as follows:

$$H(Z,Q) = \left[w_{ijk} = \begin{cases} z_{i\lambda}, & \text{if } (j,k) = q_\lambda \in Q, \lambda \in [1,n] \\ 0, & \text{otherwise} \end{cases} \right] \\ \in R^{m\times pixel\times pixel} \tag{21}$$

where, $Z \in R^{m\times n}$ and $Q \in R^{n\times 2}$.

4.2. On the significance of image pattern recognition

In this section, an analysis of the patterns in the images obtained will be performed based on the methodology explained in Section 4. When creating images, we have to take into account that the scenes should contain as much information as possible to be processed by a CNN in order extract the significant features for classification. Therefore, it is important to identify the main parameters used during the process of converting tidy data into images and their impact on the representation of the pixels in the scene. Hence, in this section, four key parameters are analysed.

On the significance of dimensionality reduction algorithms

Concerning the image pattern produced, and in particular, the positions of the characteristic pixels, we ascertained that dimensionality reduction algorithms (PCA and t-SNE) are affected by different inputs to the transformation process, such as the sample size and the seed (see Table 3).

Table 3 shows five images of the same sample when applying the two reduction algorithms and five different seeds. Images with labels (a), (b) and (c) were obtained using the PCA algorithm with seeds to 1, 7, and 23, respectively. In turn, the images labelled (d) and (e) were obtained using the t-SNE algorithm with seeds to 7 and 23, respectively. Note that we have not included the image for the case of the t-SNE algorithm with $seed = 1$. Our results show that, for this case, there is a large overlap of pixels. Our results reported only 4 pixels in an image. From the table, we see that there are clear differences in the resulting image patterns. Furthermore, the PCA always yields, when applied to a given data instance, the same pattern regardless of the random seed. On the contrary, in the case of the t-SNE algorithm, the pattern varies for each seed. We can then conclude that, for PCA, being invariant to

Table 3
Comparison of the patterns generated in the image creation by the two-dimensionality reduction algorithms (Alg.): PCA and *t*-SNE. In the different images, the impact of the choice of seeds can be seen when generating the image of the same sample.

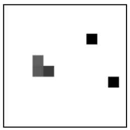
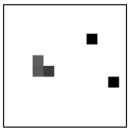
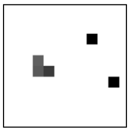
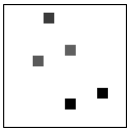
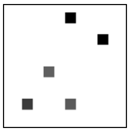
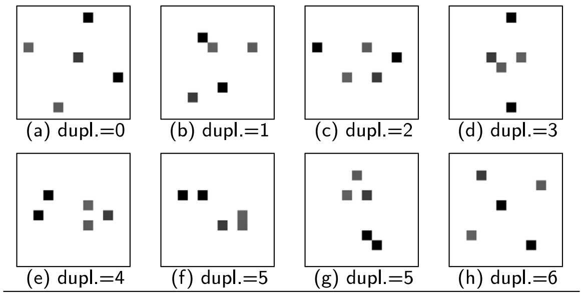
Alg.	seed=1	seed=7	seed=23
PCA	 (a)	 (b)	 (c)
t-SNE	—	 (d)	 (e)

Table 4
Variations of the image pattern using consecutive duplication (dupl.). The duplication values analysed in the image range from 0 to 6.


(a) dupl.=0 (b) dupl.=1 (c) dupl.=2 (d) dupl.=3
(e) dupl.=4 (f) dupl.=5 (g) dupl.=5 (h) dupl.=6

the random seed, no optimisation or change is required. On the other hand, *t*-SNE does require some optimisation or at least some way to avoid overlapping of the characteristic pixels, as we will explain below based on the findings reported in [24].

On the significance of t-SNE parameters

As observed in Table 3, the PCA algorithm is indifferent to the given seed, i.e., the samples have the same pattern; whereas *t*-SNE, is vulnerable to the seed, i.e., the result depends on the number of seeds. In this context, using the initial default parameters, there is the possibility of overlapping one characteristic pixel with another. We should then take into account that the characteristic pixels must not be superimposed, as the information of a pixel would be lost, not obtaining the 5 points in the two-dimensional space.

The authors of Wattenberg et al. [24] mention the need for the use of parameters for optimisation. Furthermore, *t*-SNE is mentioned as a way to visualise the data, but not as an indication of any relationship between point-to-point distances. In this case, this would not be a problem, since *t*-SNE is not used for that purpose in this work but only allows us to obtain a visual pattern. According to the authors, the following parameters are recommended to be analysed for an optimal result (taking into account that the parameters not specified have the value by default):

- *n_components*: Defines the number of final components. In our case, these components correspond to the coordinates of each pixel.
- *random_state*: Defines the random seed. Several values should be used to evaluate their impact on the patterns of the images as previously above (see 3).
- *perplexity*: Indicates how easy it is to predict the probability distribution. Its default value is 30. By modifying this parameter to 50, the result was that the variation of the patterns per random seed is not as abrupt as presented in Table 3. Moreover, the result is similar, but has slight variations, such as the rotation of the pixels with respect to the central axis.

On the significance of the overlapping pixels for t-SNE

This process will only be used when *t*-SNE is used as the dimensionality reduction algorithm. As seen in Table 3, *t*-SNE is vulnerable to the chosen seed value, while PCA remains constant in the characteristic pixel representation. This makes sense due to the stochastic nature of *t*-SNE which requires an analysis of the seed set-up [24]. An improper set-up may result in two duplicate pixels falling on the same two-dimensional point, representing fewer than the five characteristic pixels required as input data. Note that our proposal requires N characteristic pixels in a dataset. In our case, we must have $N = 5$ characteristic pixels in the resulting image.

However when designing a solution, we should realise that the patterns of the images generated should operate properly, regardless of the seed being used. In other words, we should provide a solution allowing us to overcome the stochastic nature of *t*-SNE. Our proposal involves the reproduction of k instances of the transposed matrix and before applying the *t*-SNE algorithm. In this way, we make use of k coordinate registers for characteristic pixel i, and then we average the k registers and plot their average in the two-dimensional plane.

Table 4 presents an example when applying the different images obtained by consecutive duplication: (a) duplication is 0 and there would be only 5 samples; (b) duplication is 1 and there would be 10 samples; (c) duplication is 2 and there would be 20 samples; (d) duplication is 3 and there would be 40 samples; (d) duplication is 4 and there would be 80 samples; (f) duplication is 5 and there would be 160 samples; (g) duplication is 5 and there would be 160 samples and, with respect to the previous one, another random seed is used; and (h) duplication is 6 and there would be 320 samples. Some similarity can be observed between (c), (e) and (f). On the other hand, (a) and (h) are also similar, but this would be an indication not to increase further, i.e., not to make more than 6 consecutive duplications. Due to the similar structure to the PCA, option (e) could be chosen, i.e., duplicated four consecutive times. As a summary of the above, the duplication value was set to four as in our empirical experiments this value fulfilled the purpose of maximising the model output.

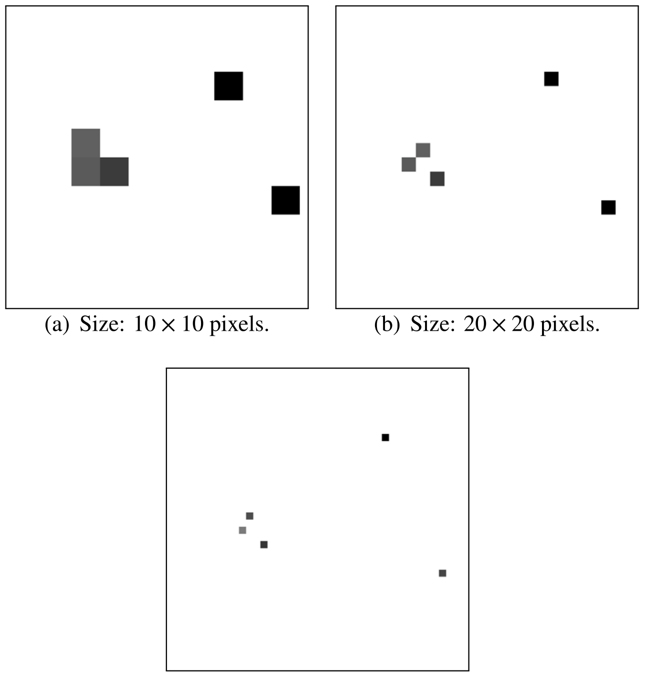
As mentioned, with the pixel duplication technique, we fulfil two definitions generated by *t*-SNE: (i) Stabilising the image pattern regardless of the seed used; and (ii) Combating overlapping pixels in a two-dimensional representation. While it is true that the image can be created with fewer characteristic pixels due to the issue of overlapping pixels, the results yielded slightly worse final performance, so it was decided to condition the experiment with the duplication of pixels.

On the significance of pixel-image size

For the development of this experiment, one of the challenges was also to find a suitable size for the pixels in the images. Summarising the tests carried out, the preliminary results of using images of 10 × 10, 20 × 20, and 40 × 40 pixels will be presented below.

Fig. 6 depicts these preliminary training results for the three image sizes used in pixels: 10 × 10 (see Fig. 6(a)); 20 × 20 (see Fig. 6(b)); and 40 × 40 (see Fig. 6(c)). All images are scaled. As can be seen, it would be difficult to visually compare samples in 40 × 40 (see Fig. 6(c)), compared to samples 10 × 10 (see Fig. 6(a)). This is due to the ratio of characteristic pixels to the total number of pixels: 10 × 10 (see Fig. 6(a)) has a ratio of 5%, while 20 × 20 (see Fig. 6(b)) has 1.25% and 40 × 40 (see Fig. 6(c)) has 0.3125%.

Therefore, the percentage of characteristic pixels in total pixels must be neither too low nor too high. If it is too low, the values will go unnoticed; if it is too high, the characteristic pixels may overlap, regardless of the dimensionality reduction algorithm used. In this work, the percentage of 5% of characteristic pixels with respect to the total



(a) Size: 10 × 10 pixels.

(b) Size: 20 × 20 pixels.

(c) Size: 40 × 40 pixels.

Fig. 6. Comparison of scaled images with respect to the number of pixels for the different matrix sizes for the images.

pixels, i.e., 10 × 10 pixels (see Fig. 6(a)), was chosen as it maximised the performance of the final CNN model. Hence, to increase the convolutional layers, the original image, which is 10 × 10 pixels (see Fig. 6(a)), was scaled to 40 × 40 pixels (see Fig. 6(c)), resulting in the same original pattern but with more pixels. This allows the necessary convolutions to be made to improve the prediction.

In this context, the comparison of the training and test for images of an initial size of 10 × 10 and 20 × 20 pixels is shown in Fig. 7. We can observe that the accuracy of the test curve for an image of 20 × 20 pixels is very unstable, varying up to approximately 30% in a single epoch of difference. This reflects great uncertainty, and, therefore, a totally random model that does not generalise. At the same time, using images of 10 × 10 pixels, we see fully smoothed, stable curves with almost no bias, indicating robust learning of the model.

Summarising, we can note that the size of the image will be closely linked to the number of characteristic pixels. In our case, for the five characteristic pixels, size 10 × 10 adequately optimises CNN learning. Note that if more wireless points are used, a larger two-dimensional space should be used, which would lead to a rescaling of the scene.

4.3. Image rendering techniques

As the procedure used was to convert tidy data into image, a CNN was developed that can carry out the classification process and, therefore, the localisation model. In order to evaluate the localisation spectrum in a wireless environment, two techniques was developed as case studies to be evaluated: one with and one without blurring of the characteristic pixel.

Case 1: Characteristic pixels without blurring

In this case, we fill in the data A_{scale} at their corresponding characteristic pixel positions in the matrix M. The matrix M is then converted into an image. This is done for each A_{scale} sample. All other values of the matrix M are zeros, so it is not significant in this case.

Therefore, for the creation of images from the tidy data, the characteristic pixel position and the scale generated by the normalisation are

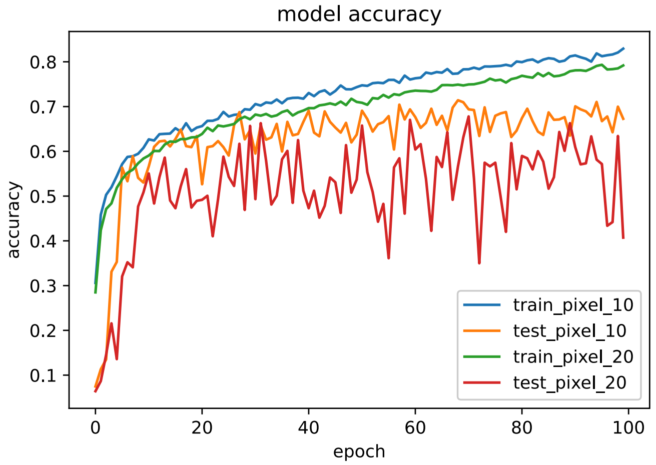


Fig. 7. Comparison of accuracy vs. epoch plots using different initial image sizes (in pixels). (i) The blue-coloured curve represents the training curve for the size of 10 × 10; (ii) the orange-coloured curve represents the test curve for the size of 10 × 10; and (iii) the green-coloured curve represents the training curve for the size of 20 × 20; and (iv) the red-coloured curve represents the test curve for the size of 20 × 20 pixels. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

used. Fig. 10(a) shows an example of the results without the blurring technique.

Case 2: Characteristic pixels with blurring

This procedure was created empirically, based on the technique of blurring in the plastic arts, specifically in drawing and painting. This technique is often used to soften and extend strokes so that the transition from intense to faint is uniform. Then, by making an analogy with the characteristic pixels as spots on a canvas, these spots can be blurred so that they cover more area without losing the intensity of the original characteristic pixel.

Therefore, the use of the blurring technique of the characteristic pixels is proposed to enlarge their area. To do this, the thickness, number, and intensity of the strokes must be defined. Each stroke would be a circumference around the pixel and the previous stroke with a thickness called *distance*, represented as r; and the number of strokes or circles would be denoted as *steps*, represented as C_x, see Fig. 8. Moreover, the intensity of this stroke or circumference would be determined by the encompassed area of the circumference, with the centre (the characteristic pixel) being the initial intensity, represented as P. Additionally, the intensity fades as you move away from the characteristic pixel. The intensity of the stroke or circumference, i, where $1 \leq step \leq total\_steps$, would be given by the following equation:

$$intensity_i = amplitude \times \frac{intensity_0}{\pi \times (total\_steps \times distance)^2} \tag{22}$$

where,

- $intensity_0$ is the value of the characteristic pixel;
- *amplitude* is a constant that would allow the intensity to extend along the strokes. For this study, we put the amplitude with the value π, simplifying the value π of the denominator and improving the blurring technique;
- *distance* is a percentage value that can range from 0 to 1. For this study, the distance 0.1 (10%) was chosen, corresponding to one pixel in the image of size $10x10$ pixels; and
- *total_steps* is the number of steps for image fading. It was set to four since, in the following steps, the intensity value is very small, close to zero. Accordingly, the value $i \in [0, total\_steps]$.

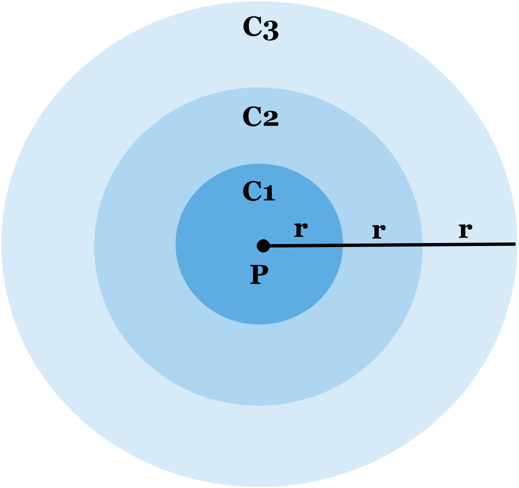


Fig. 8. Representation of the blurring technique, where r would be the distance, P the localisation of the characteristic pixel, and the colours would represent the intensity for each stroke or circumference. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

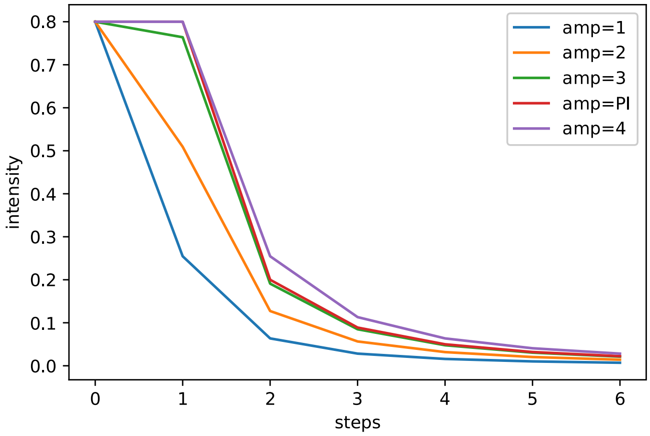


Fig. 9. Comparison of intensities by distance for different amplitudes. The amplitudes evaluated are $amp = [1, 2, 3, \pi, 4]$.

Fig. 9 shows the rationale for the parameters chosen as an example. Therefore, a comparison of the intensities per step is made for the amplitudes $amp = [1, 2, 3, \pi, 4]$ and $intensity_0 = 0.8$. In the figure, the distance is represented over all the steps interval, i.e. as a continuous function. However, in practice, the distance value is only defined for a discrete, integer value, of steps (pixels).

Finally, by using this technique, it is possible to determine the step size where the blurring of two characteristic pixels overlap. In our case, we experimented with two possible solutions for the representation of the scene with overlapping pixels (see Figs. 10):

- **Average value**: An alternative to the above is to average the intensity of both, being the one already in the matrix and the new one, see Fig. 10(b).
- **Maximum**: The highest scoring value pixel of the overlapping pixels is chosen, i.e., if the value of the pixel in the matrix is lower than the value of the incoming pixel, it is replaced, otherwise it is kept, see Fig. 10(c).

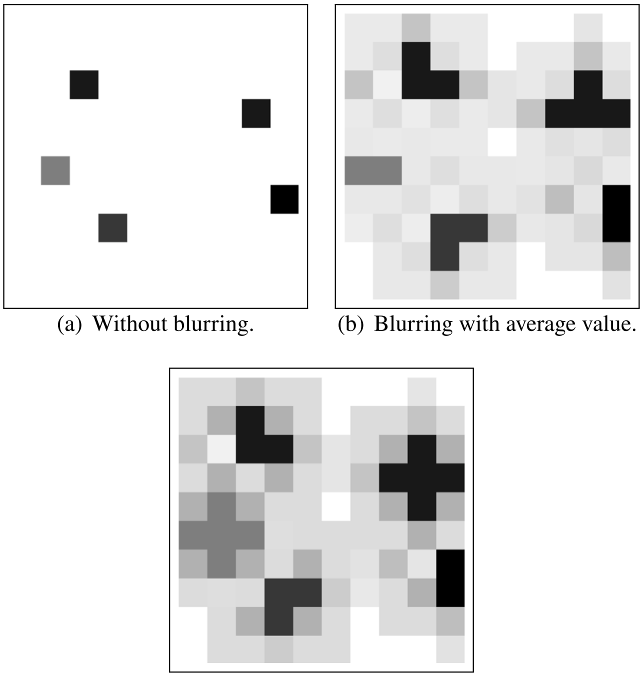

(a) Without blurring.
(b) Blurring with average value.
(c) Blurring with maximum value.

Fig. 10. Example of a scene with the different techniques used. For this image, a sample of the Tx01 dataset, sector 9 and PCA as the dimensionality reduction algorithm was used.

5. Experimental results

Having analysed the CNN input, we will now show the results obtained from the different procedures explored above, both the generated images and the CNN predictive result. First, we will look at the baseline results with symmetric TxPower setup and then discuss the results using metaheuristic optimisation to find the best combination of TxPower.

5.1. Image pattern recognition

In this section, we will present the samples generated using PCA and t-SNE in each dataset, with and without the blurring technique. The same seed was used in all evaluations for the separation of train and test data, i.e., for the same dataset, the same conditions are used, varying only the use of blurring and the type of overlapping.

Figs. 11 and 12 show the generation of three sample images for the different cases under study, for PCA and t-SNE, respectively. These sample images are clear examples of the resulting images that will be the input to CNN.

5.2. Symmetric transmission power setups

This first analysis will allow us to identify which of the two dimensionality reduction algorithms combined with three different pixel representation techniques provides the best results. By best results, we mean not only the final localisation results, but also the setting and configuration of the algorithms and techniques used in the data processing tasks. Accordingly, we define three cases: (i) without blurring; (ii) with maximum blurring; and (iii) with average blurring.

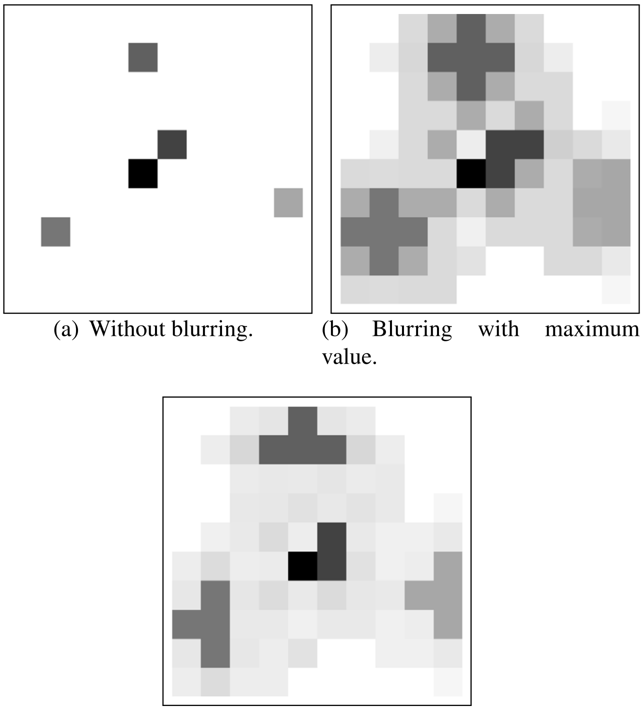

(a) Without blurring. (b) Blurring with maximum value. (c) Blurring with average value.

Fig. 11. Image samples for PCA depend on the use of the blurring technique. For all images, Tx04 was used for sector 4 of the experimental area.

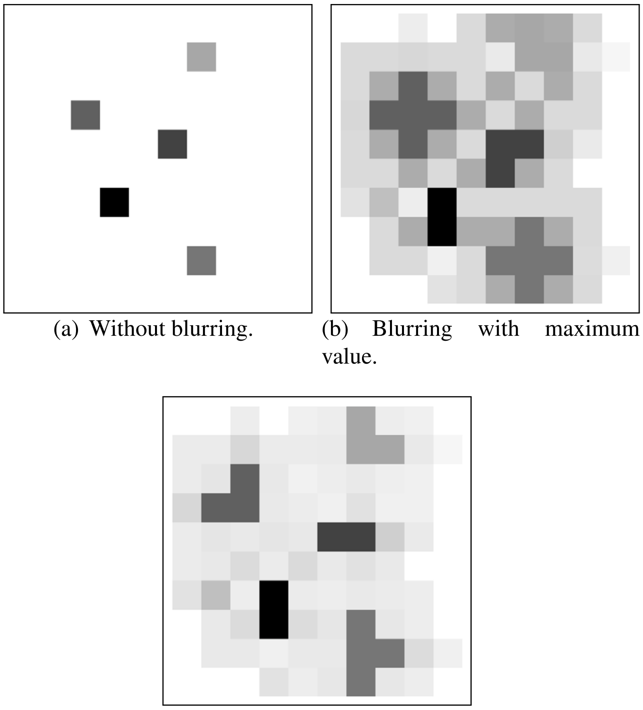

(a) Without blurring. (b) Blurring with maximum value. (c) Blurring with average value.

Fig. 12. Image samples for t-SNE depends on the use of blurring technique. For all images, Tx04 was used for sector 4 of the experimental area.

Table 5
Results in accuracy (Acc), loss, and mean error (ME) without the blurring technique for each TxPower in the test dataset for PCA and t-SNE. Best results are shown in bold.

TxPower	Algorithm	Acc	Loss	ME (m)
Tx01	*t*-SNE	67.49	1.1517	0.7661
Tx01	PCA	68.15	1.1422	0.7669
Tx02	*t*-SNE	68.49	1.1675	0.8093
Tx02	PCA	66.83	1.1508	0.8749
Tx03	*t*-SNE	69.16	1.1485	0.9334
Tx03	PCA	71.04	1.0431	0.8784
Tx04	*t*-SNE	68.71	1.0054	0.7937
Tx04	PCA	69.27	0.9831	0.7775
Tx05	*t*-SNE	65.68	1.2827	1.0863
Tx05	PCA	68.57	1.1744	1.0118
Tx06	***t*-SNE**	**77.38**	**0.9195**	**0.6470**
Tx06	**PCA**	**76.04**	**0.9288**	**0.6768**

Table 6
Results in accuracy (Acc), loss and mean error (ME) using the average blurring technique for each TxPower in Test dataset for PCA and t-SNE. Best results are shown in bold.

TxPower	Algorithm	Acc	Loss	ME (m)
Tx01	*t*-SNE	79.68	0.7302	0.4916
Tx01	PCA	81.67	0.6618	0.4267
Tx02	*t*-SNE	79.30	0.8020	0.5331
Tx02	PCA	79.08	0.8098	0.5154
Tx03	*t*-SNE	82.52	0.7189	0.5452
Tx03	PCA	85.24	0.6254	0.4569
Tx04	*t*-SNE	83.43	0.6354	0.4270
Tx04	PCA	84.52	0.5756	0.3894
Tx05	*t*-SNE	78.20	0.8251	0.6289
Tx05	PCA	78.92	0.7753	0.6158
Tx06	***t*-SNE**	**89.26**	**0.4578**	**0.3041**
Tx06	**PCA**	**90.50**	**0.4093**	**0.2738**

Case 1: Without blurring

Table 5 shows the results obtained from the six TxPower, i.e., Tx01, …, Tx06, without the blurring technique. Each dataset is evaluated with PCA and t-SNE. The results are shown for accuracy, loss and mean error (m). From the results, it is clear that the power level plays a major role on all counts (metrics). The results also show that the best results are obtained for the lowest power level used in our experiments, i.e., Tx06. The results also show that for or a given transmission power level, both dimension reduction algorithms PCA and t-SNE report similar values. This proves the effectiveness of the proposed extra processing introduced in our proposal to overcome the shortcomings of the t-SNE algorithm.

Case 2: With average blurring

Table 6 shows the results obtained from the six TxPowers, i.e., Tx01, …, Tx06, using the average blurring technique. Each dataset is evaluated with PCA and t-SNE. The results are shown in accuracy, loss and mean error (m).

The results show a substantial improvement with respect to the values reported for the previous case. In terms of the accuracy, the results reported show an improvement of more than 10% and a mean error less than half for all the transmission powers under study. Similarly to the previous case, the best results are obtained for the transmission power Tx06 for both reduction algorithms with a remarkable improvement in all metrics: an accuracy of more than 14%, a loss and a mean error of less than half those reported when no blurring is applied. We also note that the best results obtained for the two dimension reduction algorithms are very similar.

Case 3: With maximum blurring

Table 7 shows the results obtained from the six TxPowers, i.e., Tx01, …, Tx06, using maximum blurring technique.

Similarly to the previous case, the use of the blurring technique generates much better results compared to the case when no blurring

Table 7
Results in accuracy (Acc), loss and mean error (ME) using the maximum blurring technique for each TxPower in Test dataset for PCA and *t*-SNE. Best results are shown in bold.

TxPower	Algorithm	Acc	Loss	ME (m)
Tx01	*t*-SNE	81.55	0.6829	0.4413
Tx01	PCA	81.45	0.6596	0.4403
Tx02	*t*-SNE	79.78	0.7840	0.5249
Tx02	PCA	80.09	0.7808	0.5059
Tx03	*t*-SNE	85.02	0.6092	0.4488
Tx03	PCA	85.57	0.5939	0.4437
Tx04	*t*-SNE	84.07	0.5822	0.4058
Tx04	PCA	85.36	0.5692	0.3678
Tx05	*t*-SNE	79.58	0.7999	0.5877
Tx05	PCA	79.78	0.7656	0.5996
Tx06	***t*-SNE**	**89.58**	**0.4265**	**0.2938**
Tx06	**PCA**	**90.33**	**0.4173**	**0.2778**

is applied. However, when comparing the results with respect to the previous case, average blurring, the results are very similar for each one of the transmission powers under study. Once again, the best results are reported for the case when the transmission power is set to Tx06. We notice that the best results when using one of the two blurring techniques show an improvement of more than 14% in terms of the accuracy and a loss and mean error of less than half those reported for the case when no blurring technique is applied. These results clearly show the effectiveness of using the blurring technique.

From our results, it is clear that the use of blurring for image generation considerably improves the performance of the developed model compared to the use of the characteristic pixels alone. This is mainly due to the fact that, in the produced images, the values of the non-characteristic pixels are mostly non-zero and that the characteristic pixels cover more space, similar to a signal propagation. This setting is fully exploited by the convolution process.

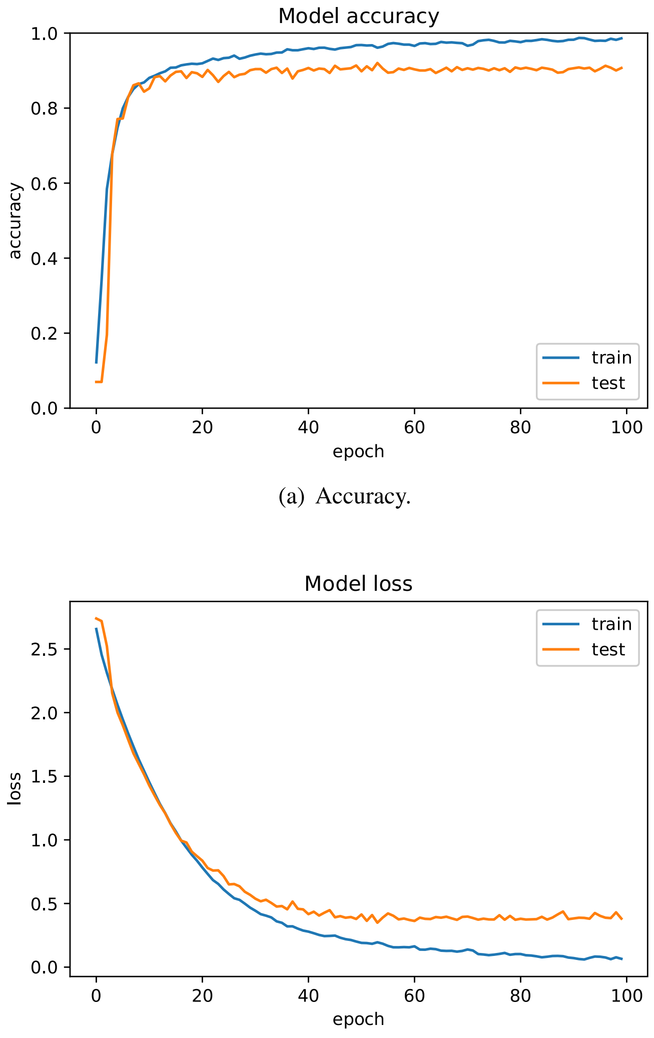
For case 1, without blurring, we can observe that the values are between 67.49% and 77.38%. However, these values can be improved by using blurring, even reaching 90% for both dimensionality reduction algorithms. This experimental study demonstrates that having more information of a scene makes the classification process more orderly by maximising the metrics.

Regarding the dimension reduction algorithms, both algorithms generate very similar results. They operate in a very similar manner, enabling valuable information to be extracted and improving the learning process. This is demonstrated by the fact that the accuracy for most of the TxPower cases increased more than 10% when the blurring technique was applied, e.g., the Tx06 dataset using PCA went from 89% to 90% (average blurring) of accuracy.

This analysis can be verified with the learning curve. Fig. 13 shows an example for the accuracy and loss metric using maximum blurring with PCA. From the figure, we can see that using 100 epochs generates an orderly learning. In fact, the number of epochs could have been reduced to 60–70. It should be noted that the *t*-SNE and PCA plots of the different results have the same verification.

5.3. Asymmetric TxPower configuration: Metaheuristic optimisation

In [9], the authors proposed the use of an asymmetric power configuration of the beacons as a means to improve the performance of wireless-based indoor localisation mechanisms. They also made use of metaheuristics in order to determine the best power setting of the beacons, i.e., the one yielding the best performance results. In this section, we carry out a similar study which should allow us to show the great benefits of our proposal. Our main aim is to show that the solution proposed in this work provides results close to the optimal setting. It is worth mentioning that the search for the optimal transmission power configuration of the various beacons requires high computational resources. Even though metaheuristic optimisation algorithms may prove effective in computing the optimal asymmetric power transmissions setup, the results reported in this section should show that the image blurring techniques provides us with results that are close to the optimal ones.



(a) Accuracy.

(b) Loss.

Fig. 13. Loss and accuracy learning curves with respect to the epochs of the CNN model, using PCA and maximum blurring for train and test dataset.

In this section, we make use of an Evolutionary Algorithm (EA) specifically a Genetic Algorithm (GA) [9,11]. EA is a generic population -based metaheuristic optimisation algorithm that uses mechanisms inspired by biological evolution, such as reproduction, mutation, recombination, and selection.

The operation of the EA algorithm is depicted in Algorithm 1 [54, 55], where: `population` denotes a list of individuals (value: 100); `cxpb` is the probability of crossing two individuals (value: 0.5); `mutpb` is the mutation probability (value: 0.2); `ngen` is the number of generations (value: 20); the `bNGF()` (*buildNextGenerationFrom*) function computes the next generation population, applying crossing and mutation operations to the selected population, validates the new individuals, and computes the statistics of this new population.

Moreover, the methods of this evolutionary algorithm are as follows: chromosome representation with a list of integers, flat crossover, random mutation and tournament method with size four. Finally, an adaptive mutation probability was empirically established as a function

Algorithm 1 Evolutionary Algorithm.

```
$evaluate(population)$
for $g$ $in$ $range$ $(ngen)$ do
    $population = select(population, len(population))$
    $offspring = bNGF(population, cxpb, mutpb)$
    $evaluate(offspring)$
    $population = offspring$
end for
```

Table 8
Best two results for PCA and t-SNE in accuracy (Acc), loss and mean error (ME) using the asymmetric TxPower combination and the average (Aver.) and maximum (Max.) blurring technique. Best result are shown in bold.

TxPower	Alg.	Blurring	Acc	Loss	ME (m)
[3-6-6-3-1]	**PCA**	**Max.**	**93.94**	**0.254**	**0.148**
[3-6-1-3-5]	t-SNE	Max.	93.35	0.283	0.157
[3-6-1-3-2]	PCA	Aver.	92.91	0.364	0.161
[3-4-1-2-2]	t-SNE	Aver.	92.54	0.373	0.173

of the generation number according to the following equation [56]:

$$P_m(i) = 0.2 \cdot e^{\frac{1-i}{10}} \tag{23}$$

where,

- i is the generation; and
- P_m is the mutation probability.

Table 8 shows the results obtained with GA for each classification model. It is important to note that the value of the evaluation function used in both algorithms is the accuracy obtained for each specific classification model of each individual in the population, i.e., for each candidate solution to the problem.

In this final result, we can observe that PCA with maximum blurring has a better combination result in the evaluated metrics. Likewise, applying t-SNE with maximum blurring, we can see that it is quite close to PCA in the metrics.

As a final discussion of the process, it can be said that both dimensionality reduction algorithms perform very similarly in terms of results. Likewise, using the maximum or average blurring technique has no impact on the final results since they are very similar in all the cases evaluated. Therefore, as a recommendation, it could be determined to use PCA for the process of transforming tidy data into image due to the representation of the characteristic pixels in the image. In turn, t-SNE, due to its stochastic nature, must be more carefully configured with respect to the analysis of the parameters and their impact on the image created.

6. Conclusions and open challenges

This article explored the transformation of tidy data to images supplemented by blurring image techniques to process BLE signals with the aim of developing wireless indoor localisation mechanisms. Our results show that the process successfully contributed to mitigating the effect of Multipath Fading (MPF). We have also show that the use of optimisation techniques based on metaheuristics may be used to determine the best TxPower configuration setting of the BLE devices.

In addition, images created from tidy data were used as input to this CNN. To this end, the partial contributions that enabled the development of this work are the use of a CNN with two parallel branches for localisation as a base learning model. Hence, to achieve this transformation, two-dimensionality reduction algorithms such as t-SNE and PCA were evaluated. In this context, two cases of image generation were tested, namely: (i) using the characteristic pixels; and (ii) applying the blurring technique by simulating the signal fading. The blurring method demonstrates that generating an image emulates the signal fading, maximising the result of the analysed metrics, and reducing the mean error. Moreover, to improve the results with asymmetric TxPower, an evolutionary algorithm was used, which demonstrates the best combination of TxPowers, further maximising the final results and bringing the solution closer to a minimum error.

Finally, regarding the open challenges, one of the main points to develop is to be able to set the blurring technique in accordance with the signal degradation itself and not empirically, as was done in this work. Using channel propagation models to develop the blurring of the characteristic pixels in the image and similarity with the signal's own degradation with respect to the device may lead to a hybrid communications-based blurring approach to the problem. Additionally, one of the main evaluation challenges of the developed model is not only to see how it performs in larger environments, but also to see what the impact of the classification process is like in multi-level environments, e.g. in a multi-storey building.

CRediT authorship contribution statement

Reewos Talla-Chumpitaz: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization. **Manuel Castillo-Cara:** Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Visualization, Writing – original draft, Writing – review & editing. **Luis Orozco-Barbosa:** Conceptualization, Investigation, Methodology, Supervision, Writing – original draft, Writing – review & editing. **Raúl García-Castro:** Investigation, Methodology, Funding acquisition, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

This scientific experiment contains the following extended data:

- Dataset used in this analysis [53].
- Python code used in the analysis to develop all of this work [57].

Acknowledgements

The research leading to these results has been funded by the European Commission H2020 project "Construction-phase diGItal Twin mOdel (COGITO)", Grant No. 958310; by the Spanish Ministry of Science, Education and Universities, the European Regional Development Fund and the State Research Agency, Grant No. PID2021-123627OB; and by the CYTED, Grant No. 520rt0011. Manuel Castillo-Cara is supported by a postdoctoral research grant (María Zambrano) of the Next Generation EU (NGEU) programme and the Spanish Ministry of Universities.

References

[1] M. Mercuri, I.R. Lorato, Y.-H. Liu, F. Wieringa, C.V. Hoof, T. Torfs, Vital-sign monitoring and spatial tracking of multiple people using a contactless radar-based sensor, Nat. Electron. 2 (6) (2019) 252–262.
[2] Y. Yang, Multi-tier computing networks for intelligent IoT, Nat. Electron. 2 (1) (2019) 4–5.
[3] P. Roy, C. Chowdhury, A survey of machine learning techniques for indoor localization and navigation systems, J. Intell. Robot. Syst. 101 (3) (2021) 1–34.
[4] A.B.M.M. Rahman, T. Li, Y. Wang, Recent advances in indoor localization via visible lights: A survey, Sensors 20 (5) (2020).
[5] C. Real Ehrlich, J. Blankenbach, Indoor localization for pedestrians with real-time capability using multi-sensor smartphones, Geo-Spatial Inform. Sci. 22 (2) (2019) 73–88.

[6] P. Roy, C. Chowdhury, D. Ghosh, S. Bandyopadhyay, JUIndoorLoc: A ubiquitous framework for smartphone-based indoor localization subject to context and device heterogeneity, Wirel. Pers. Commun. 106 (2) (2019) 739–762.

[7] C.Y.T. Kwok, M.S. Wong, S. Griffiths, F.Y.Y. Wong, R. Kam, D.C. Chin, G. Xiong, E. Mok, Performance evaluation of ibeacon deployment for location-based services in physical learning spaces, Appl. Sci. 10 (20) (2020) http://dx.doi.org/10.3390/app10207126.

[8] J. Chen, B. Zhou, S. Bao, X. Liu, Z. Gu, L. Li, Y. Zhao, J. Zhu, Q. Lia, A data-driven inertial navigation/bluetooth fusion algorithm for indoor localization, IEEE Sens. J. (2021) 1, http://dx.doi.org/10.1109/JSEN.2021.3089516.

[9] J. Lovón-Melgarejo, M. Castillo-Cara, O. Huarcaya-Canal, L. Orozco-Barbosa, I. García-Varea, Comparative study of supervised learning and metaheuristic algorithms for the development of bluetooth-based indoor localization mechanisms, IEEE Access 7 (2019) 26123–26135, http://dx.doi.org/10.1109/ACCESS.2019.2899736.

[10] Y.-Y. Chen, S.-P. Huang, T.-W. Wu, W.-T. Tsai, C.-Y. Liou, S.-G. Mao, UWB system for indoor positioning and tracking with arbitrary target orientation, optimal anchor location, and adaptive NLOS mitigation, IEEE Trans. Veh. Technol. 69 (9) (2020) 9304–9314.

[11] P. Bencak, D. Hercog, T. Lerher, Indoor positioning system based on bluetooth low energy technology and a nature-inspired optimization algorithm, Electronics 11 (3) (2022) 308.

[12] S. Fan, Y. Wu, C. Han, X. Wang, Siabr: A structured intra-attention bidirectional recurrent deep learning method for ultra-accurate terahertz indoor localization, IEEE J. Sel. Areas Commun. 39 (7) (2021) 2226–2240.

[13] M. Zhou, Y. Li, M.J. Tahir, X. Geng, Y. Wang, W. He, Integrated statistical test of signal distributions and access point contributions for Wi-Fi indoor localization, IEEE Trans. Veh. Technol. 70 (5) (2021) 5057–5070, http://dx.doi.org/10.1109/TVT.2021.3076269.

[14] F. Gu, X. Hu, M. Ramezani, D. Acharya, K. Khoshelham, S. Valaee, J. Shang, Indoor localization improved by spatial context—A survey, ACM Comput. Surv. 52 (3) (2019) 1–35.

[15] S. Ishida, Y. Takashima, S. Tagashira, A. Fukuda, Design and initial evaluation of bluetooth low energy separate channel fingerprinting, in: T. Matsuo, T. Mine, S. Hirokawa (Eds.), New Trends in E-Service and Smart Computing, Springer International Publishing, Cham, 2018, pp. 19–33.

[16] J. Liu, B. Jia, L. Guo, B. Huang, L. Wang, T. Baker, CTSLoc: an indoor localization method based on CNN by using time-series RSSI, Cluster Comput. (2021) 1–12.

[17] N.M. Tiglao, M. Alipio, R.D. Cruz, F. Bokhari, S. Rauf, S.A. Khan, Smartphone-based indoor localization techniques: State-of-the-art and classification, Measurement 179 (2021) 109349.

[18] R.S. Sinha, S.-H. Hwang, Comparison of CNN applications for RSSI-based fingerprint indoor localization, Electronics 8 (9) (2019) http://dx.doi.org/10.3390/electronics8090989.

[19] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, G. Kasneci, Deep neural networks and tabular data: A survey, 2021, arXiv preprint arXiv:2110.01889.

[20] H. Wickham, Tidy data, J. Stat. Softw. 59 (10) (2014) 1–23, http://dx.doi.org/10.18637/jss.v059.i10.

[21] L. Vasquez-Espinoza, M. Castillo-Cara, L. Orozco-Barbosa, On the relevance of the metadata used in the semantic segmentation of indoor image spaces, Expert Syst. Appl. 184 (2021) 115486, http://dx.doi.org/10.1016/j.eswa.2021.115486.

[22] L.v.d. Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (Nov) (2008) 2579–2605.

[23] A. Sharma, E. Vans, D. Shigemizu, K.A. Boroevich, T. Tsunoda, DeepInsight: A methodology to transform a non-image data to an image for convolution neural network architecture, Sci. Rep. 9 (1) (2019) 1–7.

[24] M. Wattenberg, F. Viégas, I. Johnson, How to use t-SNE effectively, Distill (2016) http://dx.doi.org/10.23915/distill.00002.

[25] D. Kim, D. Joo, J. Kim, TiVGAN: Text to image to video generation with step-by-step evolutionary generator, IEEE Access 8 (2020) 153113–153122, http://dx.doi.org/10.1109/ACCESS.2020.3017881.

[26] R. Faragher, R. Harle, Location fingerprinting with bluetooth low energy beacons, IEEE J. Sel. Areas Commun. 33 (11) (2015) 2418–2428.

[27] Q. Niu, M. Li, S. He, C. Gao, S.-H. Gary Chan, X. Luo, Resource-efficient and automated image-based indoor localization, ACM Trans. Sensor Netw. 15 (2) (2019) 1–31.

[28] P. Kriz, F. Maly, T. Kozel, Improving indoor localization using bluetooth low energy beacons, Mob. Inf. Syst. 2016 (2016).

[29] Y. Oh, H.-M. Noh, W. Shin, C-CNNLoc: Constrained CNN for robust indoor localization with building boundary, Electron. Lett. 57 (10) (2021) 422–425.

[30] F. Zafari, A. Gkelias, K.K. Leung, A survey of indoor localization systems and technologies, IEEE Commun. Surv. Tutor. 21 (3) (2019) 2568–2599.

[31] G.S.d. Blasio, A. Quesada-Arencibia, C.R. García, J.C. Rodríguez-Rodríguez, Bluetooth low energy technology applied to indoor positioning systems: An overview, in: International Conference on Computer Aided Systems Theory, Springer, 2019, pp. 83–90.

[32] H. Obeidat, W. Shuaieb, O. Obeidat, R. Abd-Alhameed, A review of indoor localization techniques and wireless technologies, Wirel. Pers. Commun. 119 (1) (2021) 289–327.

[33] M. Castillo-Cara, J. Lovón-Melgarejo, G. Bravo-Rocca, L. Orozco-Barbosa, I. García-Varea, An empirical study of the transmission power setting for bluetooth-based indoor localization mechanisms, Sensors 17 (6) (2017) 1318, http://dx.doi.org/10.3390/s17061318.

[34] J.J. Diaz, R.d.A. Maues, R.B. Soares, E.F. Nakamura, C.M. Figueiredo, Bluepass: An indoor bluetooth-based localization system for mobile applications, in: The IEEE Symposium on Computers and Communications, IEEE, 2010, pp. 778–783.

[35] C. Xiao, D. Yang, Z. Chen, G. Tan, 3-D BLE indoor localization based on denoising autoencoder, IEEE Access 5 (2017) 12751–12760.

[36] Z. Chen, Q. Zhu, Y.C. Soh, Smartphone inertial sensor-based indoor localization and tracking with ibeacon corrections, IEEE Trans. Ind. Inf. 12 (4) (2016) 1540–1549.

[37] A. Mussina, S. Aubakirov, RSSI based bluetooth low energy indoor positioning, in: 2018 IEEE 12th International Conference on Application of Information and Communication Technologies (AICT), 2018, pp. 1–4.

[38] F. Zafari, I. Papapanagiotou, M. Devetsikiotis, T. Hacker, An ibeacon based proximity and indoor localization system, 2017, arXiv preprint arXiv:1703.07876.

[39] G.S. de Blasio, J.C. Rodríguez-Rodríguez, C.R. García, A. Quesada-Arencibia, Beacon-related parameters of bluetooth low energy: Development of a semi-automatic system to study their impact on indoor positioning systems, Sensors 19 (14) (2019) http://dx.doi.org/10.3390/s19143087.

[40] J. Powar, C. Gao, R. Harle, Assessing the impact of multi-channel BLE beacons on fingerprint-based positioning, in: 2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2017, pp. 1–8, http://dx.doi.org/10.1109/IPIN.2017.8115871.

[41] S. Subedi, J.-Y. Pyun, Practical fingerprinting localization for indoor positioning system by using beacons, J. Sensors 2017 (2017).

[42] F. Subhan, S. Saleem, H. Bari, W.Z. Khan, S. Hakak, S. Ahmad, A.M. El-Sherbeeny, Linear discriminant analysis-based dynamic indoor localization using bluetooth low energy (BLE), Sustainability 12 (24) (2020) 10627.

[43] M. Mohammadi, A. Al-Fuqaha, M. Guizani, J.-S. Oh, Semisupervised deep reinforcement learning in support of IoT and smart city services, IEEE Internet Things J. 5 (2) (2018) 624–635.

[44] H. Rizk, M. Torki, M. Youssef, CellinDeep: Robust and accurate cellular-based indoor localization via deep learning, IEEE Sens. J. 19 (6) (2018) 2305–2312.

[45] B. Bhattarai, R.K. Yadav, H.-S. Gang, J.-Y. Pyun, Geomagnetic field based indoor landmark classification using deep learning, IEEE Access 7 (2019) 33943–33956.

[46] A. Belmonte-Hernández, G. Hernández-Peñaloza, D.M. Gutiérrez, F. Alvarez, SWiBluX: Multi-sensor deep learning fingerprint for precise real-time indoor tracking, IEEE Sens. J. 19 (9) (2019) 3473–3486.

[47] S. Sadowski, P. Spachos, RSSI-based indoor localization with the internet of things, IEEE Access 6 (2018) 30149–30161.

[48] J. Jiao, F. Li, W. Tang, Z. Deng, J. Cao, A hybrid fusion of wireless signals and RGB image for indoor positioning, Int. J. Distrib. Sens. Netw. 14 (2) (2018) 1550147718757664.

[49] F. Alhomayani, M.H. Mahoor, Deep learning methods for fingerprint-based indoor positioning: a review, J. Locat. Based Serv. 14 (3) (2020) 129–200.

[50] C.-H. Hsieh, J.-Y. Chen, B.-H. Nien, Deep learning-based indoor localization using received signal strength and channel state information, IEEE Access 7 (2019) 33256–33267.

[51] JAALEE Inc., Beacon IB0004-N plus, 2018, Link: https://www.jaalee.com/. [[Online: accessed 18-may-2022].

[52] Trendnet, Micro bluetooth USb adapter, 2018, Link: https://www.trendnet.com/products/USB-adapters/TBW-107UB/. [Online: accessed 18-may-2022].

[53] M. Castillo-Cara, Bluetooth indoor localization dataset, 2022, http://dx.doi.org/10.5281/zenodo.6343083, [Online: accessed 18-may-2022].

[54] T. Bäck, D.B. Fogel, Z. Michalewicz, Evolutionary Computation 1: Basic Algorithms and Operators, Vol. 1, CRC Press, 2000.

[55] F.-A. Fortin, F.-M. De Rainville, M.-A. Gardner, M. Parizeau, C. Gagné, DEAP: Evolutionary algorithms made easy, J. Mach. Learn. Res. 13 (2012) 2171–2175.

[56] S. Sivanandam, S. Deepa, Introduction to Genetic Algorithms. Chapter 3: Terminologies and Operators of GA, Springer International Publishing, 2008, http://dx.doi.org/10.1007/978-3-540-73190-0.

[57] M. Castillo-Cara, R. Talla-Chumpitaz, Deep learning indoor localization from tidy data into synthetic images, 2022, http://dx.doi.org/10.5281/zenodo.6557364, [Online: accessed 18-may-2022].