



WiFiNet: WiFi-based indoor localisation using CNNs



Noelia Hernández*, Ignacio Parra, Héctor Corrales, Rubén Izquierdo, Augusto Luis Ballardini, Carlota Salinas, Iván García

Universidad de Alcalá, 28805 Alcalá de Henares (Madrid), Spain

ARTICLE INFO

Keywords:
Indoor localisation
WiFi
Fingerprinting
Deep learning
2010 MSC:
68T05
91E40
82C32
94A99

ABSTRACT

Different technologies have been proposed to provide indoor localisation: magnetic field, Bluetooth, WiFi, etc. Among them, WiFi is the one with the highest availability and highest accuracy. This fact allows for an ubiquitous accurate localisation available for almost any environment and any device. However, WiFi-based localisation is still an open problem.

In this article, we propose a new WiFi-based indoor localisation system that takes advantage of the great ability of Convolutional Neural Networks in classification problems. Three different approaches were used to achieve this goal: a custom architecture called WiFiNet, designed and trained specifically to solve this problem, and the most popular pre-trained networks using both transfer learning and feature extraction.

Results indicate that WiFiNet is as a great approach for indoor localisation in a medium-sized environment (30 positions and 113 access points) as it reduces the mean localisation error (33%) and the processing time when compared with state-of-the-art WiFi indoor localisation algorithms such as SVM.

1. Introduction

In the recent years, the number of applications and connected devices have grown exponentially (Fig. 1). According to CISCO, this trend is expected to continue in the following years reaching 28.5 billion IoT (Internet of Things) connected devices in 2022 (Thomas Barnett, Jain, Andra, & Khurana, 2018). Laptops, phones and tablets were the devices traditionally used for connectivity. However, in the recent years lots of other devices, such as smartwatches, fridges, even bulbs or toothbrushes, have joined the world of connected devices. This trend seeks to help the users in every aspect of their life (Atzori, Iera, & Morabito, 2010) with IoT applications in smart homes (Bergeron, Bouchard, Gaboury, & Giroux, 2018), cities (Islam, Uddin, Mukherjee, & Nirjon, 2018) or warehouses (Rabeah, Aditya, Alawwad, Molisch, & Behairy, 2016).

With smart sensors almost everywhere, IoT applications will require or benefit from a reliable and accurate localisation of certain devices. For instance, locating medical staff, medical equipment or even patients in hospitals would increase the quality and rapidity of the medical services (McAllister, El-Tawab, & Heydari, 2017). This localisation must be real-time, accurate both indoors and outdoors (Racko, Machaj, & Brida, 2018) and with low power consumption. So far, this localisation was

generally provided through the use of GPS (Global Positioning System) which has one main problem: it does not work indoors because of the NLOS (Non-line-of-sight) effect.

As a consequence, different technologies have been proposed to provide indoor localisation (Zafari, Gkelias, & Leung, 2019): magnetic field (Carrera-Villacrés, Zhao, Braun, Luo, & Zhao, 2018), Bluetooth (Ayyalasomayajula, Vasisht, & Bharadia, 2018), Kriz, Maly, and Tomáš (2016), WiFi (Wang, Gao, Mao, & Pandey, 2017; Luo et al., 2017), UWB (Ultra Wide Band) (Ridolfi, Van de Velde, Steendam, & De Poorter, 2016), RFID (Calderoni, Ferrara, Franco, & Maio, 2015) or visible light (Di Lascio, Varshney, Voigt, & Perez-Penichet, 2016; Hu et al., 2018) among others.

Wireless RSSI-based indoor localisation (based on Bluetooth, UWB or WiFi) is the most common due to its high availability (in both IoT devices and the environments), low consumption and high accuracy. Among them, WiFi is the one with the highest availability and highest accuracy at the cost of slightly higher power consumption (Sadowski & Spachos, 2018). This fact allows for an ubiquitous accurate localisation available for almost any environment and any device.

However, WiFi-based localisation is still an open problem. Since RADAR, the first method proposed in 2000 (Bahl & Padmanabhan, 2000), the research community has proposed different solutions to

* Corresponding author.

E-mail addresses: noelia.hernandez@uah.es (N. Hernández), ignacio.parra@uah.es (I. Parra), hector.corrales@uah.es (H. Corrales), ruben.izquierdo@uah.es (R. Izquierdo), augusto.ballardini@uah.es (A.L. Ballardini), carlota.salinasmaldo@uah.es (C. Salinas), ivan.garciad@uah.es (I. García).

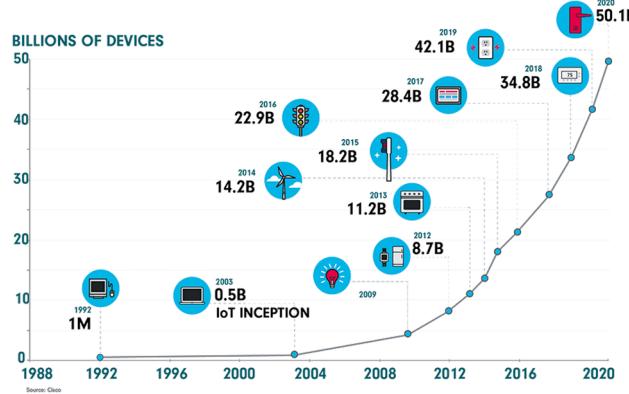


Fig. 1. Growth of IoT devices. Source: Cisco.

increase the localisation accuracy using methods based on state-of-the-art algorithms such as Random Forest (Jedari, Wu, Rashidzadeh, & Saif, 2015) or classifier ensembles (Torres-Sospedra et al., 2016; Hernández, Alonso, & Ocaña, 2017) achieving promising results.

But most of these methods, based on classic machine learning, have two main problems: their low scalability to big environments and their low generalisation ability which increases the localisation error when the number of site-surveyed positions is not high enough for the size of the environment. As a consequence, new methods were proposed to solve these problems seeking for a reduction of the site-survey effort (Hernández, Ocaña, Alonso, & Kim, 2017) or a self-maintenance of the system (Tao & Zhao, 2018). Despite all the efforts dedicated to indoor WiFi-based localisation research, there is still room for improvement.

In this article, we propose a new WiFi-based indoor localisation system that takes advantage of the CNNs (Convolutional Neural Networks) great performance in classification problems, looking for a reduction of the localisation error without forgetting about its generalisation ability and its scalability. Three different approaches were explored to achieve this goal: a custom architecture, called WiFiNet, designed and trained specifically to solve this problem, transfer learning using eight well-known pre-trained networks and, finally, feature extraction from pre-trained networks to be used along with classic classifiers.

As most CNNs are designed to classify images, the WiFi samples (the RSS (Received Signal Strength) from the WiFi APs (Access Points)) must be converted into images before training the network. The way the APs are ordered to form the image is key for the localisation procedure, as the CNNs will search for features taking small areas of the images as a whole. For this reason, the images will be created ordering the APs according to the order they appear in the collected data increasing its spatial relation. Once the images are created, they can be fed to the CNN for both training in an initial stage and classification during the use of the CNN for device localisation.

Three different experiments were designed to test the system: localisation at positions covered in the training dataset (the usual way of testing new localisation systems), localisation at positions non-existent in the training dataset (to evaluate its generalisation ability) and localisation under realistic conditions (walking around the environment). Test data was collected on a different time and date from the training data to capture the high variability of the WiFi signal over time. In addition, the processing time is analysed to evaluate the scalability of the system.

WiFiNet arises as a great approach for indoor localisation as it reduces the mean localisation error when compared with state-of-the-art algorithms (RMSE of 3.3 m compared with 4.4 m using SVM in a medium-sized environment with 30 positions and 113 APs). At the same time, WiFiNet's scalability lead us to think that it will be able to perform real-time localisation in bigger environments than state-of-the-art

algorithms.

The rest of the manuscript is structured as follows: Section 2 analyses the WiFi indoor localisation state of the art presenting the most important and recent work in the field. Section 3 presents the core of this work. It describes the different approaches to work with CNNs and describes WiFiNet, our custom architecture. Then, Section 4 goes in depth with the experimental evaluation, analysis of the results and discussion. Finally, the main contributions in this work along with some future research lines are summarized in Section 5.

2. State of the art

The objective of any indoor localisation system is to accurately estimate the position of a device inside a building using a map as a reference. This localisation is one of the key enabling technologies for IoT devices applications. However, indoor localisation is not solved yet despite all the efforts dedicated to find a solution in the last decades. As explained in the introduction, we can use a GPS to locate a device outdoors, but it can not be used for indoor localisation due to NLOS. This is why the research community has been working to find a solution for the indoor localisation problem using a different technology, being WiFi one of the most promising ones.

Two different kind of methods can be used for WiFi localisation: based on propagation models and based on RSS fingerprints.

The former use propagation models to describe how the signal is propagated in the environment (Mazuelas et al., 2009; Yang, Lee, & Moessner, 2016). These models are usually used to translate the RSS into a distance and then the estimated position of the device is obtained using lateration algorithms. The main problem of this method is that the exact location of every AP must be known, which is difficult in real environments. In addition, propagation models do not accurately adjust in indoor environments due to signal blockage. That is why propagation model-based methods are mainly used outdoors.

This leads to the use fingerprint-based methods in indoor environments. These use a fingerprint database to estimate the position of a device by means of different classification algorithms (Bahl & Padmanabhan, 2000; Hernández et al., 2017; Nabati, Ghorashi, & Shahbazian, 2020). The fingerprint database stores information, typically the RSS, at certain locations of the environment, modelling the characteristics of the signal so it can be compared in a later stage with new measurements collected at unknown positions.

These methods have been improving by using more complex and modern algorithms since their first appearance in the year 2000 with the RADAR algorithm (Bahl & Padmanabhan, 2000). Since then, almost every classification algorithm in the literature has been tested for this purpose, being the following the ones providing the lowest localisation errors and, as a consequence, the most used algorithms to be compared with new methods:

- K-Nearest Neighbours (KNN) (Kibler & Aha, 1987) is one of the simplest classification algorithms. It uses a voting mechanism to select the estimated class among the k nearest neighbours to the object using a distance measure. KNN was used in RADAR (Bahl & Padmanabhan, 2000) which is world-wide recognized as one of the pioneers in the research field of WiFi indoor localisation. As WiFi indoor localisation is highly environment-dependent, research results can not be directly compared. Instead, new methods are usually compared with a KNN-based method using its results as a baseline.
- Support Vector Machine (SVM) (Cortes & Vapnik, 1995) is a kernel-based classifier. It has been proved as one of the most accurate classic algorithms for WiFi indoor localisation (Wei, Hwang, & Lee, 2018).
- Classifier ensembles combine groups of machine learning systems, in the literature called weak learners, in order to yield better accuracy and lower error rates (Kuncheva, 2001). Their performance strongly relies on diversity of the weak learners (Brown, Wyatt, Harris, & Yao, 2005), in the way that ideally they make their errors on different

parts of the problem space. As a consequence, they are well-recognized machine learning tools capable of obtaining better performance than a single component classifier (Hernández et al., 2017; Lee, Kim, & Moon, 2019).

- Neural Networks (NN) are based on a network of neurons that adjust their weights during the learning process. These methods have been used for indoor localisation for decades now (Laoudias, Eliades, Kemppi, Panayiotou, & Polycarpou, 2009) introducing deeper and more complex networks in the recent years such as Deep Neural Networks (DNN) (Khatab, Hajihoseini, & Ghorashi, 2018) and Convolutional Neural Networks (CNNs) as it will be explained in Section 3.

3. CNNs for WiFi indoor localisation

Recently, CNNs, a special kind of multi-layer neural network including convolutional layers, have become a key method for classification specially in computer vision and natural language processing problems. In this area, CNNs have beaten traditional methods to the point that they are increasingly being used for classification problems.

However, CNNs goodness is infrequently tested in other areas. When training a CNN to solve a new problem, there are four different approaches that might be used:

- Design and train a new network architecture: This approach requires a deep understanding of CNNs architecture design, a big dataset and powerful hardware to train the new network. This method is usually chosen when the problem to solve is very different from the problems solved by the already existing networks and the number of available training images is high enough.
- Use of a pre-trained network (transfer learning): There is a wide range of pre-trained networks that can be used as a starting point to solve new problems. Most of these models have in common that they were designed to improve the performance classifying ImageNet (ImageNet, 2009), a large scale dataset (more than a million images) with 1000 different classes used in the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) (Russakovsky et al., 2015). In this approach, the knowledge learned during the previous training (low-level features extraction such as colors, edges, shapes, etc.) is maintained and only a fine-tuning will be done during the new training. By using transfer learning, the new training will be much faster and the number of images required in the new dataset much smaller than training a network from scratch.
- Train an existing network architecture from scratch: This approach re-uses the design of an already existing CNN, but the weights are recomputed to solve the new problem. The difference with the first approach is that there is no need to design a CNN architecture, but a high number of images are still needed to train the network. This approach is generally used when the problem is similar to an existing one but the previous method (transfer learning) is not providing the expected results.
- Use an existing pre-trained network to extract features and then using them as inputs for a classic classification algorithm: This approach takes advantage of the architecture and weights of pre-trained networks. The CNN is only used to extract features that will be used as input features for other classifiers (such as KNN or SVM). This approach is mostly used when the computing power, the time or the number of samples is insufficient to train a network but also to take advantage of the great ability of CNNs extracting significant features.

In this article, we propose to use CNNs to estimate the location of a device by using the RSS from the WiFi APs on the environment. To do so, we have followed three approaches: A new network architecture was designed and trained from scratch, different pre-trained networks were used to take advantage of transfer learning and finally, pre-trained

networks were also used to extract features to be used as inputs with some classic classifiers.

As all of these networks are designed to classify images, the first step will be to transform the RSS samples. Once the RSS samples are transformed into images, they can be used to train the network (or extract features). Naturally, the same transformation will be required to locate the device during the localisation stage.

3.1. Creation of images from RSS samples

The first step to create the images is to select the order in which the RSS samples from the different APs will be arranged in the images. This is not a trivial procedure, as the CNNs will apply the operations through the different layers on spatially-neighboring pixels, especially in their initial layers (Coates & Ng, 2012). As a consequence, the order of the APs can not be randomly selected, as the RSS information at neighboring positions of the image will be associated at the first layers of the CNN. In this paper, the order of the APs is selected according to the order they appear in the collected data increasing its spatial relation (Fig. 2).

As shown in the figure, the APs seen from the first position of the environment will be at the beginning of the data vector. Then, the APs seen from the closest position (that are not already in the vector) will be added. This procedure will continue until all the positions of the environment are analysed.

This way, the APs that are visible from the same position of the environment will be arranged on the same area of the image creating groups of neighbouring pixels with related information. Then, the data of each sample will be re-arranged following the selected order into an Y square matrix, with

$$Y = [\sqrt{N_{AP}}] \quad (1)$$

being N_{AP} the number of APs in the complete training dataset.

The next step is to transform the RSS values into pixel intensity values (i.e. from 0 to 255). Given that the collected RSS ranges from -99 dBm to -30 dBm, this transformation is performed by adding an offset of + 200 to the RSS, obtaining medium-ranged intensity values (from 101 to 170). If there is no RSS from an AP, its pixel intensity will be set to 0. This way, there is a high differentiation between non-seen APs and low RSS APs. Of course, if the total number of APs is not a perfect square, the image will have some pixels without an associated AP which will also be set to 0.

Finally, depending on the number of APs and the selected CNN, the images should be resized to agree with the input layer of the network (pre-trained networks input image size is around 224x224 pixels). Once the training RSS samples are transformed into images, they can be used to train a network following one of the approaches explained at the beginning of Section 3. The complete process is summarized in Fig. 3.

3.2. WiFiNet: a custom architecture

As explained at the beginning of the section, custom architectures are used when the problem to solve is highly different from the problems already solved by the existing networks and the number of available training samples is enough. However, knowing how many samples are enough to train a network in advance is almost impossible, being necessary to train and test the network to obtain this knowledge from the results. Thus, in practice, this method is used to solve new problems as is our case: Existing architectures and pre-trained networks are designed with the aim of identifying real objects (cars, animals, etc.) in the input images, but our goal is to obtain a location from RSS measurements which have no evident meaning in the image.

In this work, we propose a custom network architecture, called WiFiNet, based on ResNet bottleneck units to take advantage of its efficiency and accuracy.

Our network is formed by 13 convolutional layers (Conv) followed

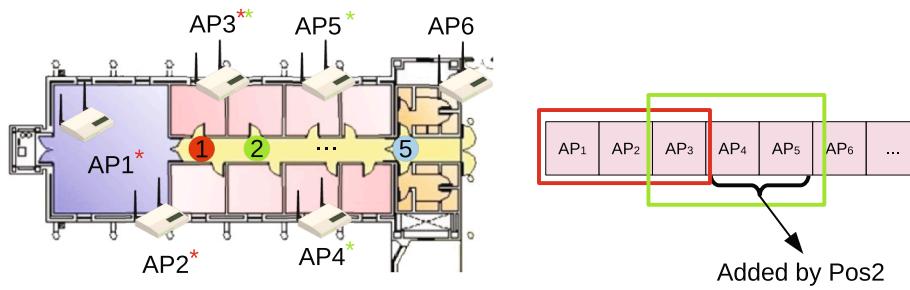


Fig. 2. APs order selection: The APs with a red * are seen from position 1 and the ones with a green * are seen from position 2.

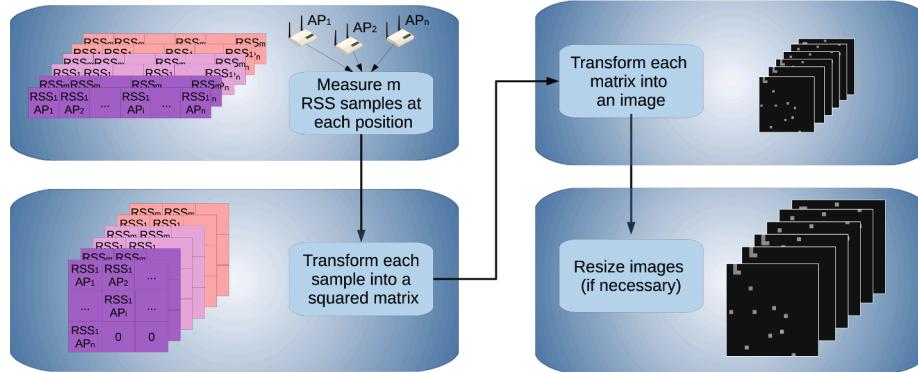


Fig. 3. Collecting and pre-processing data to convert RSS samples into images.

by a Batch Normalisation (BN) layer and a ReLU layer (Rectified Linear Unit) every three Conv + BN layers. Fig. 4 shows the complete architecture. As it can be seen, an initial input Conv + BN + ReLU block is followed by four Conv + BN + Conv + BN + Conv + BN + ReLU blocks in which the output of the different layers grows with their depth in the network. This decision was made to keep the weight of the pixels on the borders of the images the same as the rest of the pixels. This decision was validated experimentally as this network got the best results in terms of accuracy, mean error and processing time. The last layer of the network is a Fully Connected + Softmax layer that will use the activations of the previous layer to decide the most likely position in which the RSS sample was collected. As this network was specifically designed to solve this localisation problem, there is no need to resize the input images (as explained section 3.1).

3.3. Transfer learning using pre-trained networks

Transfer learning (TL) is a great approach to reduce the training

effort (in time, processing power and training dataset size) by using a pre-trained network as starting point to learn how to solve new problems. This way, we can take advantage of a working network architecture previously trained during hundreds of hours using thousand of images. In this approach, the initial layers of the network are maintained along with their weights to keep the knowledge obtained during the previous training (low-level features extraction such as colors, edges, shapes, etc.). The weights of these layers will be fine-tuned during the new training. The final layers of the network (classification layers) are replaced with “blank” ones configured to differentiate between the classes of the new problem. The weights of the classification layers will be adjusted from scratch during the training.

There are lots of pre-trained CNNs for classification that can be used for this purpose. The most popular ones are AlexNet (Krizhevsky, Sutskever, & Hinton, 2012), GoogLeNet (Szegedy et al., 2015), ResNet (He, Zhang, Ren, & Sun, 2016) or SqueezeNet (Iandola et al., 2016) but others such as NasNet Mobile (Zoph, Vasudevan, Shlens, & Le, 2017) or Inception ResNetv2 (Szegedy, Ioffe, & Vanhoucke, 2016) are deeper

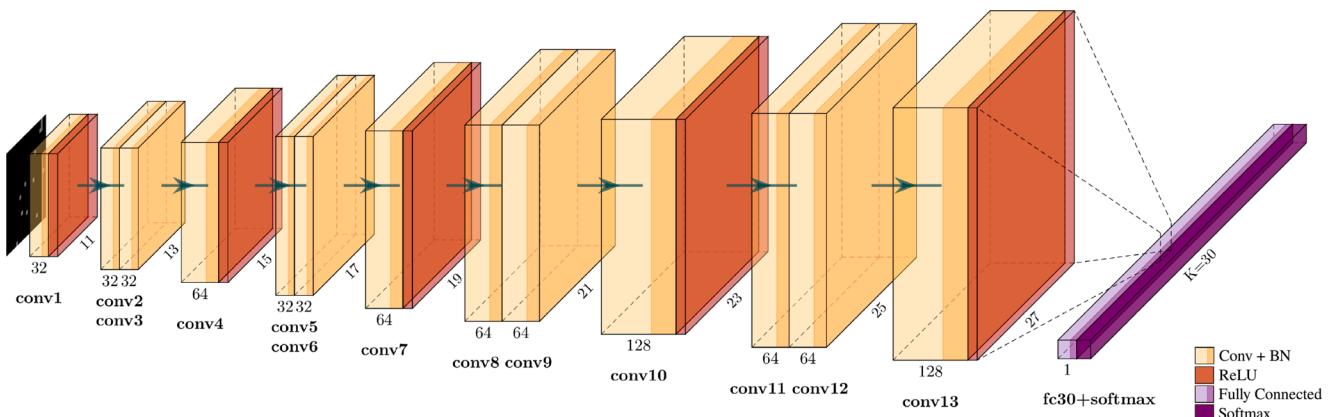


Fig. 4. WiFiNet: Custom architecture.

networks that are also worth testing. Their input layer is designed to use 224x224 or 227x227 square images. As a consequence, the input images must be resized to fulfill this requirement.

A summary of all the tested TL algorithms and configurations is shown in [Table 1](#) where sgdm is the Stochastic Gradient Descent with Momentum optimizer, LR is the Learning Rate and MB is the size of the mini-batch for each training iteration. As can be seen, the selected parameters are the same for all the CNNs, except for ResNet101, NasNet Mobile and Inception ResNetv2 in which the selected mini-batch size is smaller in order to fit the network in the GPU memory. Changing this parameter may lead to better generalization at the cost of higher computational time per epoch during the training.

3.4. Feature extraction from pre-trained networks

Feature extraction (FE) also takes advantage of the knowledge learned by a pre-trained network, but reduces the time, processing power and dataset size to a minimum since there is no need to retrain the network. With this approach, the activations (features) of the layer previous to the classification layers are directly used to train a classic classifier.

As CNNs provide an accurate solution for WiFi-based indoor localisation, and the dataset size in our problem is small (60 samples per class/position in comparison with thousand of images usually used per class) we considered that by using feature-extraction traditional methods could benefit from the CNNs ability to extract significant features improving their accuracy.

The combinations of two pre-trained CNNs with two classic classifiers were used to test this approach: ResNet18 and AlexNet were used to extract the features and then, SVM and subspaceKNN were trained using the extracted features. [Table 1](#) also contains the algorithms and configurations using feature extraction.

3.5. Classic classifiers: baseline

Finally, six machine learning algorithms were tested for comparison

Table 1
Summary of the algorithms and their configuration.

Algorithm	Short name	Configuration
WiFiNet	WiFiNet	sgdm, 10^{-3} LR, 120 MB
AlexNet (TL) (Krizhevsky et al., 2012)	AlexNet	sgdm, 10^{-3} LR, 120 MB
SqueezeNet (TL) (Iandola et al., 2016)	Squeeze	sgdm, 10^{-3} LR, 120 MB
ResNet18 (TL) (He et al., 2016)	ResN18	sgdm, 10^{-3} LR, 120 MB
ResNet50 (TL) (He et al., 2016)	ResN50	sgdm, 10^{-3} LR, 120 MB
ResNet101 (TL) (He et al., 2016)	ResN101	sgdm, 10^{-3} LR, 60 MB
GoogLeNet (TL) (Szegedy et al., 2015)	GoogLe	sgdm, 10^{-3} LR, 120 MB
NasNet Mobile (TL) (Zoph et al., 2017)	NasMob	sgdm, 10^{-3} LR, 60 MB
Inception ResNetv2 (TL) (Szegedy et al., 2016)	Inception	sgdm, 10^{-3} LR, 30 MB
SVM (AlexNet FE)	SVMAlex	Linear kernel
SubKNN (AlexNet FE)	sKNNAlex	Euclidean distance, K = 1
SVM (ResNet18 FE)	SVMRN18	Linear kernel
SubKNN (ResNet18 FE)	sKNNRN18	Euclidean distance, K = 1
KNN (Bahl & Padmanabhan, 2000)	KNN	Euclidean distance, K = 1
SVM (Wei et al., 2018)	SVM	Linear kernel
Bagged Trees (Lee et al., 2019)	BagT	MSE split criterion
AdaBoost (Feng et al., 2014)	BoostT	MSE split criterion
Subspace Discriminant (Hernández et al., 2017)	SubDisc	Linear Discriminant
Subspace KNN (Hernández et al., 2017)	SubKNN	Euclidean distance, K = 1

purposes. Two classic classifiers: K-Nearest Neighbours (KNN) ([Kibler & Aha, 1987](#)) and Support Vector Machines (SVM) ([Cortes & Vapnik, 1995](#)) and four Classifier Ensembles (CE): AdaBoost ([Freund & Schapire, 1997](#)), Bagged Trees ([Breiman, 1996](#)), Subspace KNN and Subspace Discriminant ([Ho, 1998](#)).

We have tested several configurations for each algorithm and, in the following sections, we present the results of the ones providing the best results. However, for the SVM algorithm similar results were obtained using both RBF and linear kernels, so in order to reduce the complexity of the system ([Hsu, Chang, & Lin, 2010](#)) we present the results using the Linear kernel.

The final configuration of all these algorithms can be found on [Table 1](#).

4. Results and discussion

This section presents the experiments carried out to validate the localisation algorithms. First, the experimental set-up is described. Then, the different experiments along with a critical discussion of the obtained results is provided.

Three different experiments are presented: training and test using measurements collected at the same locations (test data collected one week apart from training data), test using measurements collected at different positions from the training ones (collected two months apart from training data) and test using data collected in motion (collected two months apart from training data).

4.1. Experimental set-up

The proposed CNNs were tested in a complex real-world environment set at the University of Alcalá (Madrid, Spain) covering 3600m^2 ([Fig. 5](#)).

The training dataset was collected using the internal WiFi interface of a Toshiba Portégé Z830-10F laptop on 30 evenly distributed positions ([Fig. 5a](#)). The minimum distance between neighbour positions in the training dataset is 3 m, the maximum distance 7.77 m and the mean distance 4.46 m.

Test data was collected a week later on the training positions. The test dataset is completed with measurements collected two months after the training dataset at 67 new positions ([Fig. 5b](#)) and with data collected while walking around the environment ([Fig. 5c](#)). This temporal difference on the collected data allows to capture the high variability of the WiFi signal over time. At the same time, measurements collected in non-existent positions in the training dataset and during the trajectory allows to test the generalisation capability of the proposed localisation method under more realistic conditions both standing at fixed positions and in motion.

A total of 113 APs were detected during the collection of the training dataset. We have no information about their configuration nor their location. This way, the information from all the existing APs is used to create the input images resulting in 11×11 square images. [Fig. 6a](#) shows some examples of the images used to train and test the CNN.

4.2. Localisation at existing positions in the training dataset

In this experiment, the test positions are the same as the training positions. This way, the behaviour of the different algorithms is tested when the localisation is performed at known positions, which is the best-case scenario. The perfect localisation algorithm should obtain a RMSE (Root Mean Square Error) of 0 m with an accuracy of 100%.

[Fig. 7](#) shows the accuracy ([Fig. 7a](#)) and RMSE ([Fig. 7b](#)) using the algorithms summarized in [Table 1](#). Four different colours are used to help with the differentiation of the four different approaches (custom architecture, transfer learning, feature extraction and classic learners). [Table 2](#) shows a summary with the results of the best algorithms using each approach.

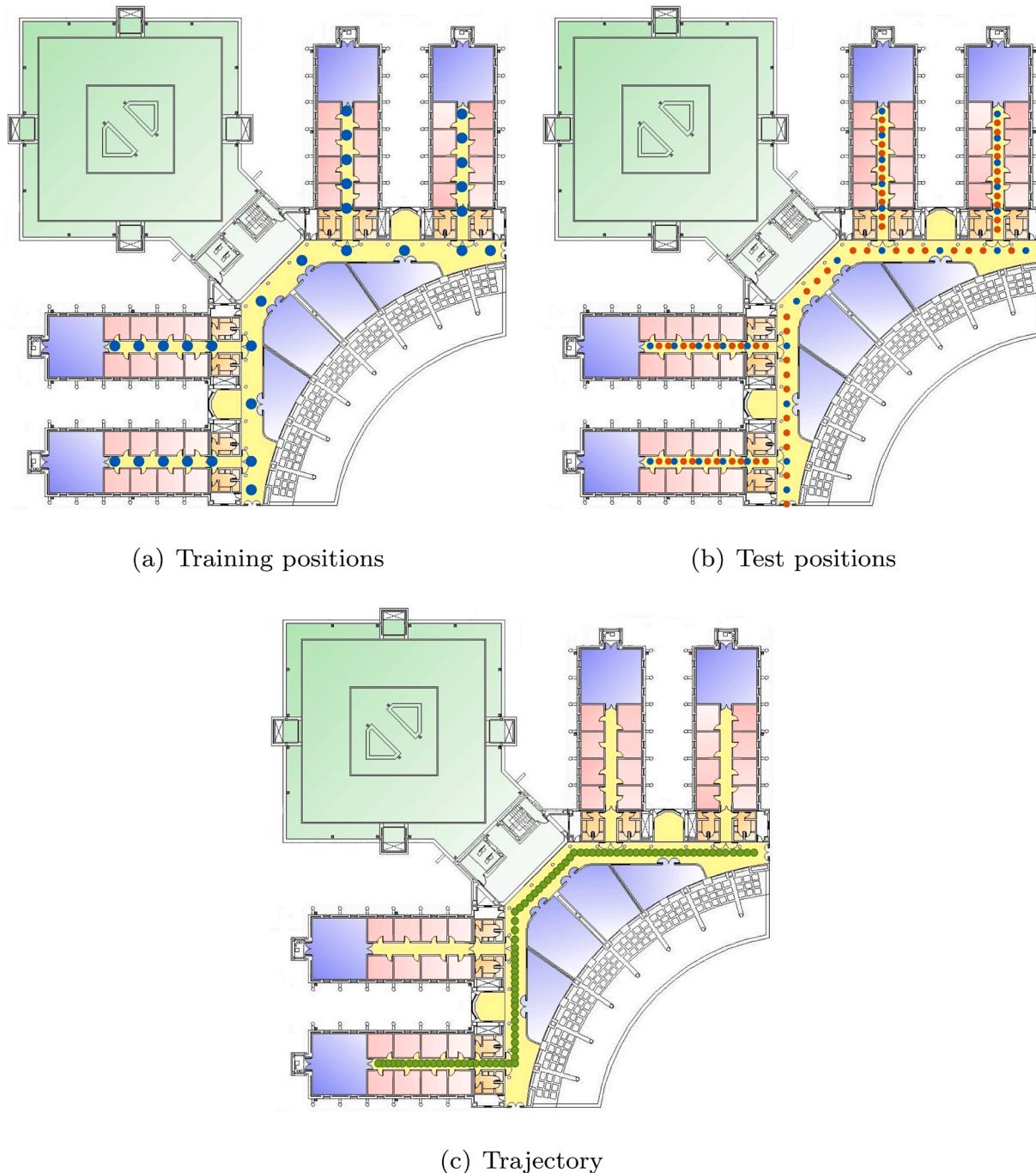


Fig. 5. Experimental environment located at the University of Alcalá. Training positions are represented with blue circles. Test positions different from the training ones are represented using red circles.

As can be seen, the highest accuracy and lowest RMSE is obtained using transfer learning using ResNet18 (24.6 cm), closely followed by ResNet101, ResNet50, GoogLeNet and WiFiNet (28 cm). However, feature extraction using ResNet18 (63.5 cm) does not seem to be as good as the use of the CNN itself. Among feature extraction algorithms, the lowest RMSE is obtained using SVM with AlexNet FE (34.9 cm) while using SVM (no FE) the RMSE is almost double (72.7 cm).

Looking at the results it can be concluded that WiFiNet and TL are the best approaches identifying already known locations with RMSE from 24.6 cm (ResNet18) to 28 cm (WiFiNet). The RMSE obtained using the best classic classifier is 250% higher than using WiFiNet (subspaceKNN (70 cm) and SVM (72.7 cm)).

4.3. Localisation at unknown positions: Generalization ability

This experiment is intended to evaluate the generalization ability of the different algorithms, as the test positions are different from the training ones. In this case, the accuracy of all the algorithms is always 0% as the test positions do not exist in the training data and, as a consequence, the RMSE can not be 0m. The best performing approach will be the one with the lower RMSE.

Fig. 8 shows the box plot (Fig. 8a) and RMSE (Fig. 8b) using the algorithms summarized in Table 1. As in the previous section, four different colours are used to help with the differentiation of the four different approaches. Table 3 shows a summary of the best results using each approach.

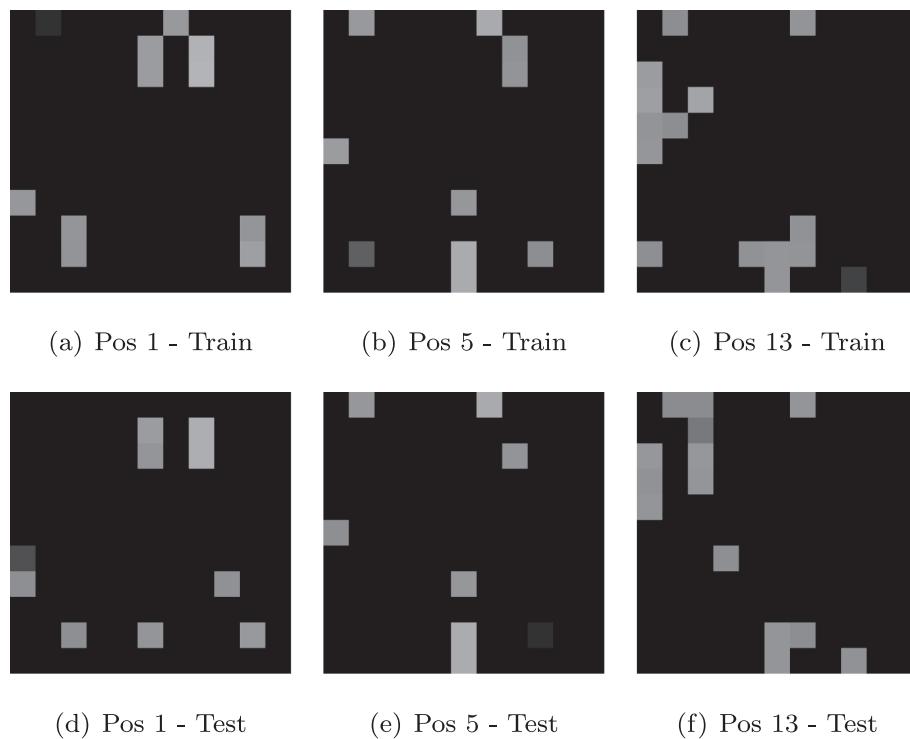


Fig. 6. Example of training and test images.

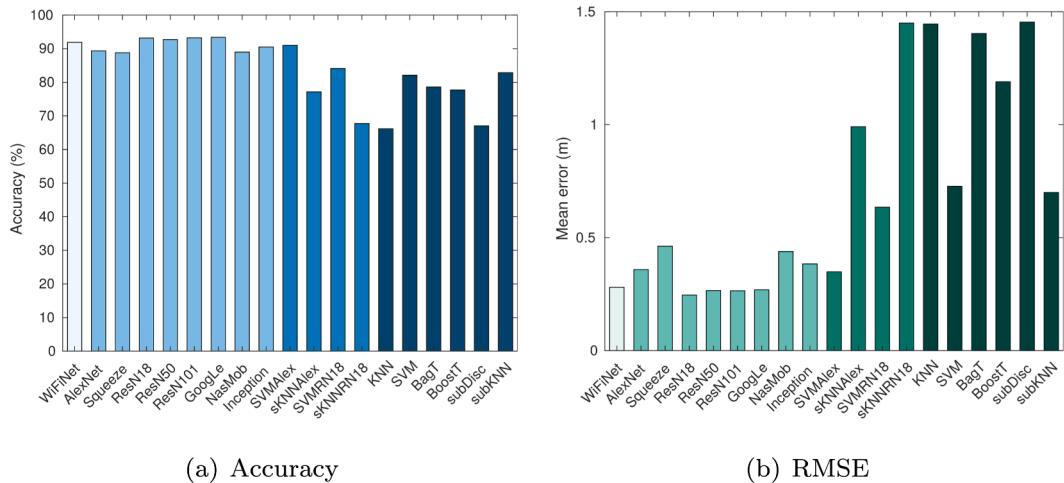


Fig. 7. Accuracy and RMSE using measurements collected at the same positions as the training data.

Table 2
Summary of the results locating at existing positions in the training dataset.

Algorithm	Accuracy	RMSE
WiFiNet	91.89%	28 cm
ResNet18 (TL)	93.17%	24.6 cm
SVMAlex (FE)	91%	35.9 cm
SVM	82.11%	72.7 cm

As can it be seen, the lowest RMSE and most compact box plot is now obtained using WiFiNet (3.5m) followed by ResNet18 (3.7m). The RMSE for the rest of CNNs used for TL is higher, especially in deeper networks (GoogleNet, NasNet Mobile and Inception ResNetv2) probably due to over-learning. In addition, feature extraction does not seem to be a good approach for localisation at unknown positions obtaining errors from 5.7m (SVM with AlexNet FE) to 6.7m (subspaceKNN with

ResNet18 FE). Finally, SVM is the best classic learner with a RMSE of 4m (around a 14% higher than using WiFiNet).

As in the previous section, WiFiNet and ResNet18 are the best locating the device at positions non-existent in the training dataset. However, even though SVM with AlexNet FE was one of the best when performing the localisation at known positions, the RMSE in this experiment shows that it is not good at generalising. On the other hand, the classic classifiers performance in this experiment shows that the difference with WiFiNet and ResNet18 shortens when the localisation is performed at unknown positions even though the mean localisation error at known positions was 250% higher than using WiFiNet.

4.4. Localisation in motion: Performance analysis of the proposed system

Finally, this experiment is aimed to evaluate the system under realistic conditions (i.e. walking around while collecting test measurements

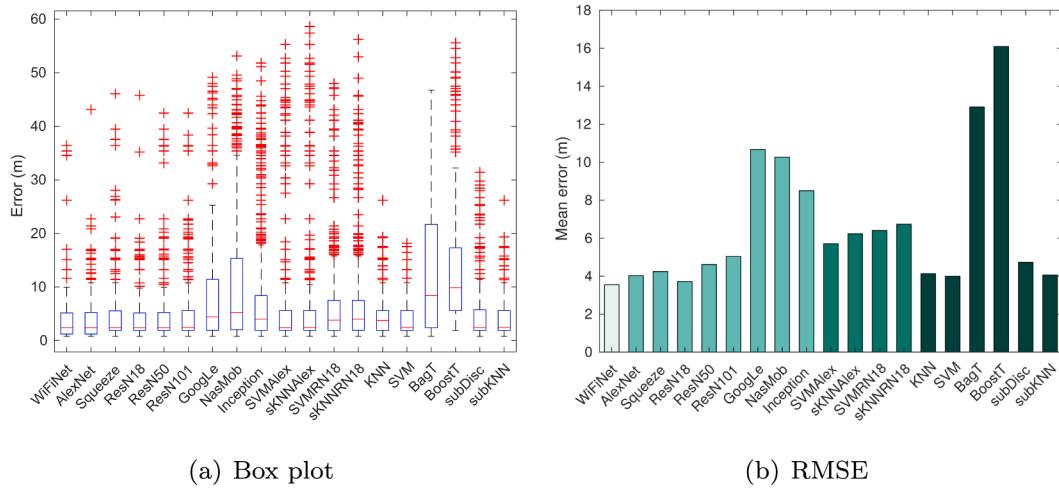


Fig. 8. Box plot and RMSE using measurements collected at positions not covered by the training data.

Table 3

Summary of the results locating at non-existent positions in the training dataset.

Algorithm	Mean Error	75th percentile
WiFiNet	3.5 m	5.1 m
ResNet18 (TL)	3.7 m	5.2 m
SVMAlex (FE)	5.7 m	5.6 m
SVM	4 m	5.6 m

in both known and unknown positions). To do so, the data collected following the trajectory shown in Fig. 5c was used. As in the previous section, the RMSE can not be 0 m since there are measurements collected at positions not covered in the training dataset.

Fig. 9 shows the box plot (Fig. 9a) and RMSE (Fig. 9b) using the algorithms summarized in Table 1. Again, four different colours are used to help with the differentiation of the four different approaches. Table 4 shows a summary of the best results using each approach.

As can be seen, WiFiNet is again the algorithm performing the localisation with the lowest RMSE (3.3 m). However, the differences with the rest of the algorithms are higher in this experiment. Using ResNet18, the second best algorithm so far, the RMSE is 24% higher and using SVM a 32% higher than using WiFiNet. At the same time, the 75th percentile using WiFiNet (5.0 m) is also lower than using the rest of the algorithms which means that the highest localisation errors are also lower using WiFiNet.

Table 4

Summary of the results locating a device in motion.

Algorithm	Mean Error	75th percentile
WiFiNet	3.3 m	5.0 m
ResNet18 (TL)	4.1 m	6.0 m
SVMAlex (FE)	4.3 m	5.7 m
SVM	4.3 m	6.4 m

As a conclusion, we can say that using WiFiNet reduces the RMSE when compared with other state-of-the-art methods when locating a device in motion. We can also conclude that even though TL (especially using ResNet18) provided almost the best results when locating the device at static positions of the environment (both known and unknown), they do not seem to adapt well to motion conditions.

4.5. Processing time: Localisation in real time

Another important characteristic that was not evaluated so far is the processing time. Here, we need to differentiate between the time required to train the system and the time spent locating one sample. Of course, the training time is important, but is not a requirement to achieve real-time localisation as the training is performed off-line and prior to the use of the localisation system.

On the other hand, the time required to obtain one location

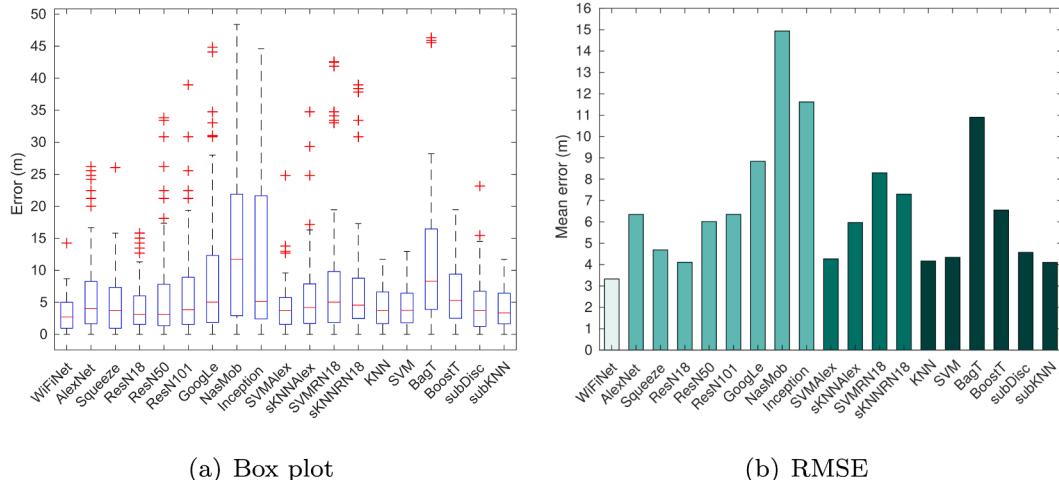


Fig. 9. Box plot and RMSE using the measurements collected while walking around.

estimation is critical, as the longer it takes the further the difference between two consecutive locations of the user and, as a consequence, the harder to track its position in the environment. Thus, the processing time must be lower than the WiFi acquisition time (which is usually up to 4 Hz - 250 ms) to achieve real-time localisation.

Fig. 10 shows the time required to classify one sample using the different algorithms. Real-time localisation is achieved by all the algorithms as the processing times for one sample are under 20 ms for all of them in a medium-size environment with 30 positions and 113 APs.

However, the processing time is highly dependent on the number of positions and the number of APs. As a consequence, the algorithms might not be scalable to bigger environments even though all the tested algorithms are able to work under real-time conditions in this scenario.

To evaluate how increasing the number of inputs and outputs affected each one of the systems in terms of processing time, two experiments were performed:

- In the first one, the number of inputs was increased by adding synthetic APs data to the original dataset. These synthetic APs were created by adding noise to the existing APs RSS. This new semi-synthetic dataset was composed of a total of 50172 APs to match the pre-trained CNNs that use 224x224 images.

Using this dataset WiFiNet needs 0,46 ms (twice the time required using 113 APs) while SVM needs 47,27 ms per sample (163 times the time required using 113 APs).

- To evaluate the impact on the processing time of the increase in the number of outputs (positions) we used the test dataset which is composed of 94 positions as explained in Section 4.1. This made the total number of outputs 94.

The change on the number of outputs barely affects the CNNs' processing time, as the only difference will be the connections on the last fc + softmax layers. However, for the classic classifiers an increase in the number of positions leads to a high increase in the processing time. For an environment with 94 positions WiFiNet needs 0.27 ms per sample (the same as in the 30 positions environment) while SVM needs 3.79 ms per sample (13 times the time required to locate 1 sample in the 30 positions environment).

Thus, WiFiNet is not only the localisation algorithm obtaining the lowest mean localisation error but the time it requires per location request is lower, leading us to think that it will be able to achieve real-time localisation in bigger environments than state-of-the-art methods such as SVM.

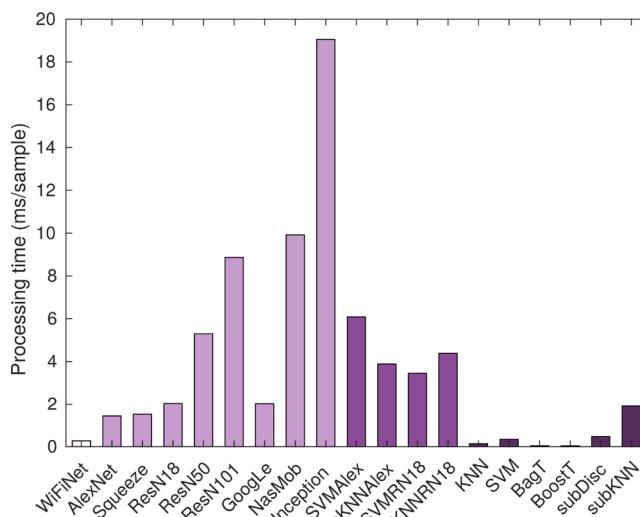


Fig. 10. Processing time per sample during localisation.

5. Conclusions

In this article we have presented a new method to estimate the location of a device using the WiFi RSS in indoor environments using CNNs. Three different approaches were evaluated: A new architecture, called WiFiNet, designed specifically to solve the problem and the most popular pre-trained networks using both transfer learning and feature extraction. In addition, the top performing classic algorithms for WiFi-based indoor localisation were tested to be used as a baseline.

Three experiments under different conditions were carried out to analyse the performance of the different methods: Using data collected in the same positions as in the training dataset to test their accuracy, using data collected in positions non-existent in the training dataset to evaluate their generalisation ability and using data collected while walking around to assess their performance under realistic conditions.

After analysing the results, we have reached the following conclusions:

- WiFiNet and ResNet18 (TL) are the best methods at learning how to identify static positions (both known and unknown).
- ResNet18 (TL) and SVM show a good generalisation ability when locating the device at static positions. However, they do not seem to be as good when locating the device under realistic conditions (in motion).
- WiFiNet is the best generalising and adapting to realistic conditions, achieving a RMSE of 3.3 m using measurements collected while walking around the environment.
- Even though TL (especially using ResNet18) provided almost the best results when locating the device at static positions of the environment (both known and unknown), TL does not seem to adapt well to motion conditions.
- Feature extraction does not seem to be a good approach especially regarding its low generalisation ability.

On the light of the results, we can conclude that using WiFiNet to perform WiFi-based indoor localisation arises as a good approach, reducing the RMSE when compared with other state-of-the-art methods, especially when locating a device under realistic conditions (3.3 m WiFiNet vs 4.4 m SVM). Moreover, WiFiNet is highly scalable, leading us to think that it will be able to achieve real-time localisation in bigger environments than other state-of-the-art methods such as SVM (0.46 ms per sample WiFiNet vs 47.27 ms per sample SVM using 50172 APs).

In the future, we plan on taking advantage of the great scalability of WiFiNet to improve localisation in bigger environments utilising the data collected in projects such as Radiocells.org ([Radiocells.org, 2009](#)) and obtaining "virtual" RSS samples for intermediate positions. To do so, we will use the CSE algorithm proposed in [Hernández et al. \(2017\)](#). This way, the resolution of the WiFiNet localisation system will be improved without the need of collecting measurements on additional positions. With this approach we expect to reduce the mean localisation error without increasing the processing time which was the main problem of the original CSE algorithm and other state-of-the-art methods.

CRediT authorship contribution statement

Noelia Hernández: Conceptualization, Methodology, Investigation, Software, Writing - original draft. **Ignacio Parra:** Conceptualization, Data curation, Writing - review & editing. **Héctor Corrales:** Methodology, Investigation. **Rubén Izquierdo:** Methodology, Investigation. **Augusto Luis Ballardini:** Software, Validation. **Carlota Salinas:** Writing - review & editing. **Iván García:** 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.

Acknowledgement

This work has received funding from the Comunidad de Madrid trough project CM/JIN/2019-012, the Spanish Ministry of Economy under project DPI2017-90035-R, the University of Alcalá through project CCG19/IA-077 and from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 754382. The Titan Xp used for this research was donated by the NVIDIA Corporation through the GPU Grant Program.

References

- Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. *Computer Networks*, 54, 2787–2805.
- Ayyalasomayajula, R., Vasishth, D., & Bharadwaj, D. (2018). BLoc: CSI-based accurate localization for BLE tags. In *Proceedings of the 14th International Conference on Emerging Networking EXperiments and Technologies* (pp. 126–138).
- Bahl, P., & Padmanabhan, V. N. (2000). RADAR: An in-building RF-based user location and tracking system. In *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies* (pp. 775–784).
- Bergeron, F., Bouchard, K., Gaboury, S., & Giroux, S. (2018). Tracking objects within a smart home. *Expert Systems with Applications*, 113, 428–442.
- Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24, 123–140.
- Brown, G., Wyatt, J., Harris, R., & Yao, X. (2005). Diversity creation methods: a survey and categorization. *Information Fusion*, 6, 5–20.
- Calderoni, L., Ferrara, M., Franco, A., & Maio, D. (2015). Indoor localization in a hospital environment using random forest classifiers. *Expert Systems with Applications*, 42, 125–134.
- Carrera-Villacrés, J., Zhao, Z., Braun, T., Luo, H., & Zhao, F. (2018). Discriminative learning-based smartphone indoor localization. CoRR, abs/1804.03961. <http://arxiv.org/abs/1804.03961>.
- Coates, A., & Ng, Y. A. (2012). Selecting receptive fields in deep networks. In *Proceedings of the 24th International Conference on Neural Information Processing* (pp. 2528–2536).
- Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20, 273–297.
- Di Lascio, E., Varshney, A., Voigt, T., & Perez-Penichet, C. (2016). Poster abstract: LocaLight - a battery-free passive localization system using visible light. In *2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)* (pp. 1–2).
- Feng, Y., Minghua, J., Jing, L., Xiao, Q., Ming, H., Tao, P., & Xinrong, H. (2014). Improved adaboost-based fingerprint algorithm for wifi indoor localization. In *2014 IEEE 7th Joint International Information Technology and Artificial Intelligence Conference* (pp. 16–19).
- Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55, 119–139.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 770–778).
- Hernández, N., Alonso, J. M., & Ocaña, M. (2017). Fuzzy classifier ensembles for hierarchical wifi-based semantic indoor localization. *Expert Systems with Applications*, 90, 394–404.
- Hernández, N., Ocaña, M., Alonso, J.M., & Kim, E. (2017b). Continuous space estimation: Increasing WiFi-based indoor localization resolution without increasing the site-survey effort. *Sensors*, 17 (1), 1–23.
- Ho, T. (1998). The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20, 832–844.
- Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2010). *A practical guide to support vector classification*.
- Hu, Y., Xiong, Y., Huang, W., Li, X.-Y., Yang, P., Zhang, Y., & Mao, X. (2018). Lightitude: Indoor positioning using uneven light intensity distribution. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, 2, 67:1–67:25.
- Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., & Keutzer, K. (2016). SqueezeNet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size. arXiv, <https://arxiv.org/abs/1602.07360>.
- ImageNet (2009). Image database. <http://www.image-net.org/> accessed on February 2020.
- Islam, B., Uddin, M., Mukherjee, S., & Nirjon, S. (2018). Rethinking ranging of unmodified BLE peripherals in smart city infrastructure. In *Proceedings of the 9th ACM Multimedia Systems Conference* (pp. 339–350).
- Jedari, E., Wu, Z., Rashidzadeh, R., & Saif, M. (2015). Wi-Fi based indoor location positioning employing random forest classifier. In *Proceedings of the 2015 International Conference on Indoor Positioning and Indoor Navigation* (pp. 1–5).
- Khatab, Z. E., Hajihoseini, A., & Ghorashi, S. A. (2018). A fingerprint method for indoor localization using autoencoder based deep extreme learning machine. *IEEE Sensors Letters*, 2, 1–4.
- Kibler, D., & Aha, D. (1987). Learning representative exemplars of concepts: An initial case study. In *Proceedings of the International Workshop on Machine Learning* (pp. 24–30).
- Kriz, P., Maly, F., & Tomáš, K. (2016). Improving indoor localization using bluetooth low energy beacons. *Mobile Information Systems*, 2016, 1–11.
- Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012). Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J.C. Burges, L. Bottou, & K.Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25* (pp. 1097–1105).
- Kuncheva, L. (2001). Combining classifiers: Soft computing solutions. *Pattern Recognition. From Classical to Modern Approaches chapter*, 15, 427–452.
- Laoudias, C., Eliades, D.G., Kemppi, P., Panayiotou, C.G., & Polycarpou, M.M. (2009). Indoor localization using neural networks with location fingerprints. In *Artificial Neural Networks – ICANN 2009* (pp. 954–963).
- Lee, S., Kim, J., & Moon, N. (2019). Random forest and wifi fingerprint-based indoor location recognition system using smart watch. *Human-centric Computing and Information Sciences*, 9, 1–14.
- Luo, C., Cheng, L., Chan, M. C., Gu, Y., Li, J., & Ming, Z. (2017). Pallas: Self-bootstrapping fine-grained passive indoor localization using wifi monitors. *IEEE Transactions on Mobile Computing*, 16, 466–481.
- Mazuelas, S., Bahillo, A., Lorenzo, R. M., Fernandez, P., Lago, F. A., Garcia, E., Blas, J., & Abril, E. J. (2009). Robust indoor positioning provided by real-time RSSI values in unmodified WLAN networks. *IEEE Journal of Selected Topics in Signal Processing*, 3, 821–831.
- McAllister, T. D., El-Tawab, S., & Heydari, M. H. (2017). Localization of health center assets through an IoT environment (LoCATE). In *2017 Systems and Information Engineering Design Symposium (SIEDS)* (pp. 132–137).
- Nabati, M., Ghorashi, S. A., & Shahbazian, R. (2020). Joint coordinate optimization in fingerprint-based indoor positioning. *IEEE Communications Letters*.
- Rabeah, N., Aditya, S., Alawwad, F., Molisch, A. F., & Behairy, H. (2016). A measurement-based blocking distribution for improving localization in warehouse environments. In *2016 International Conference on Electromagnetics in Advanced Applications (ICEAA)* (pp. 894–897).
- Racko, J., Machaj, J., & Brida, P. (2018). Ubiquitous smartphone based localization with door crossing detection. *Engineering Applications of Artificial Intelligence*, 75, 88–93.
- Radiocells.org (2009). Community project to collect information on cell tower and WiFi base stations. <https://radiocells.org/> accessed on February 2021.
- Ridolfi, M., Van de Velde, S., Steendam, H., & De Poorter, E. (2016). WiFi ad-hoc mesh network and MAC protocol solution for UWBT indoor localization systems. In *2016 Symposium on Communications and Vehicular Technologies (SCVT)* (pp. 1–6).
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115, 211–252.
- Sadowski, S., & Spachos, P. (2018). RSSI-based indoor localization with the Internet of Things. *IEEE Access*, 6, 30149–30161.
- Szegedy, C., Ioffe, S., & Vanhoucke, V. (2016). Inception-v4, inception-resnet and the impact of residual connections on learning. CoRR. <http://arxiv.org/abs/1602.07261>.
- Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 1–9).
- Tao, Y., & Zhao, L. (2018). A novel system for wifi radio map automatic adaptation and indoor positioning. *IEEE Transactions on Vehicular Technology*, 67, 10683–10692.
- Thomas Barnett, J., Jain, S., Andra, U., & Khurana, T. (2018). Cisco Visual Networking Index (VNI) Complete Forecast, Update 2017–2022. Technical Report Americas/EMEA Cisco Knowledge Network (CKN).
- Torres-Sospedra, J., Mendoza-Silva, G. M., Montoliu, R., Belmonte, O., Benítez, F., & Huerta, J. (2016). Ensembles of indoor positioning systems based on fingerprinting: Simplifying parameter selection and obtaining robust systems. In *Proceedings of the 2016 International Conference on Indoor Positioning and Indoor Navigation* (pp. 1–8).
- Wang, X., Gao, L., Mao, S., & Pandey, S. (2017). CSI-based fingerprinting for indoor localization: A deep learning approach. *IEEE Transactions on Vehicular Technology*, 66, 763–776.
- Wei, Y., Hwang, S., & Lee, S. (2018). IoT-aided fingerprint indoor positioning using support vector classification. In *2018 International Conference on Information and Communication Technology Convergence* (pp. 973–975).
- Yang, J., Lee, H., & Moessner, K. (2016). Multilateration localization based on singular value decomposition for 3D indoor positioning. In *Proceedings of the 2016 International Conference on Indoor Positioning and Indoor Navigation* (pp. 1–8).
- Zafari, F., Gkelias, A., & Leung, K. K. (2019). A survey of indoor localization systems and technologies. *IEEE Communications Surveys Tutorials*, 21, 2568–2599.
- Zoph, B., Vasudevan, V., Shlens, J., & Le, Q.V. (2017). Learning transferable architectures for scalable image recognition. CoRR,. <http://arxiv.org/abs/1707.07012>.