Convolutional Neural Network as sensor fusion algorithm applied to IPIN2019 dataset

Gaetano (Luca) De Palma*, Antoni Peréz-Navarro*, Raul Montoliu Colas**

* Faculty of Computer Sciences, Multimedia and Telecommunication, Universitat Oberta de Catalunya (UOC), Barcelona, Spain, e-mail: [gdepalma,aperezn]@uoc.edu

** Institute of New Imaging Technologies, Jaume I University, Castellon,´ Spain e-mail: montoliu@ uji.es

Abstract. This work-in-progress explores the use of Convolutional Neural Net-works (CNNs) in a sensor fusion approach for indoor localization, a crucial component in computer vision, robotics, and navigation. CNNs have emerged in the last few years as sensor fusion algorithms, using deep learning to process multi-sensor data. We apply CNN to the IPIN 2019 competition dataset. The method consists of processing sensor data and transforming it into images, and training a CNN model for position estimation. Preliminary results show promise in specific scenarios, but the CNN approach struggles with generalization on diverse tracks.

Keywords. Indoor Localization, Sensor Fusion, Convolutional Neural Network

1. Introduction

Indoor localization faces challenges due to weak signals within buildings, leading to the ineffectiveness of Global Navigation Satellite Systems (GNSS). So it is necessary to use other available sensors to find the user's position indoor. In that, sensor fusion plays a crucial role, it enables the efficient utilization of multiple sensors to gather comprehensive and accurate information from the environment. While CNNs have been in existence

Published in "Proceedings of the 18th International Conference on Location Based Services (LBS 2023)", edited by Haosheng Huang, Nico Van de Weghe and Georg Gartner, LBS 2023, 20-22 November 2023 Ghent, Belgium.

This contribution underwent single-blind peer review based on the paper. https://doi.org/10.34726/5721 | © Authors 2023. CC BY 4.0 License.

for many years, their application as sensor fusion algorithms is a relatively recent development. CNNs excel in extracting relevant information by leveraging spatially hierarchical features from diverse and complex data sources, including multi-spectral or multi-modal data. Specifically, researchers have reported successful utilization of CNNs for feature extraction using multiple modalities like RGB and Infrared images (Hu 2018, Li 2023). The work in (Antsfeld 2020) largely inspired us. In this work-in-progress paper, we will discuss the application of CNN as a sensor fusion algorithm to an indoor localization context using the IPIN 2019 contest dataset (Potortí 2020). We present preliminary results and provide insight into upcoming research directions.

2. Methods

This work follows a classical machine learning method: data and labels are collected, a model is trained using them, and then tested with a validation set. The model is used to predict labels for new data, and performance is evaluated by comparing the predicted labels with the true ones (Bishop 2006). Performance measures such as RMSE, MSE, and displacement are calculated. The proposed method is shown in Fig. 1. It is applied to the IPIN 2019 dataset (Potortí 2020), that are the data collected using an Android smartphone and its build-in sensors. The data are composed of time, latitude, longitude and, sometimes, altitude in format of decimal degrees, and meters respectively. The time is in milliseconds. The data collected comes from sensors such as the accelerometer, gyroscope, magnetometer, attitude (AHRS), GNSS, barometer, and the Wi-Fi intensity, among others. The labels are the ground truth of the position.

The 2019 dataset is composed by 40 tracks, each track was walked four times (e.g.T01 01.txt, T01 02.txt, T01 03.txt, T01 04.txt) often back and forth. The IPIN organizers, moreover, provided a set of geo-referenced maps of the buildings where the data were collected (Moayeri 2016, UJIIndoorLoc 2023, Long-Term 2014).

The original data were pre-processed to separate sensor information, reduce data volume, and ensure portability across different configurations. Matplotlib and Python 3.10 were used. Data from each sensor were separated and saved in new files for easier management (e.g. 2019 T01 01 ACCE.csv for the accelerometer data, etc.). The SensorTime was used to synchronize data from different sensors, as recommended by the IPIN organizers.

The original input data were decimated and filtered to reduce both the data vol-ume and noise. All raw sensor data were down-sampled to 50 Hz, which is considered sufficient to capture user displacement. (Fig. 2 shows an example.)

In this work, the training data were augmented with the magnitude of each sensor (accelerometer, gyroscope, and magnetometer) and for each sensor an additional column with the calculated magnitude (e.g., [x, y, z, $\|v\|_2$]) was added.

CNNs are known for their capability in feature extraction from images, and in this work, we extended their application by transforming sensor signals into images. Using time windows of 1 second, we created images with 12 rows and 50 columns, representing sensor axes and their norm. The sampling rate was set at 50 samples per second, and padding was applied to ensure an integer number of images, thus enabling data conversion for CNN input.

This down-sampling reduce the amount of data available for training, to over-come that firstly, we reduced the time window to 0.5 seconds, doubling the available ”images” per track. Secondly, we iterated through randomly shuffled batches, using each training track 2 or 3 times. These modifications noticeably improved our research progress.

Fig. 1 From raw data to input images.

Ground truth data consist of latitude and longitude coordinates, converted to UTM coordinates. The study uses positional increments between images as labels, generating missing points based on constant speed assumption. Output labels represent user’s incremental position in three dimensions, requiring integration to reconstruct the user’s position during CNN’s prediction phase.

The calibration data are collected and recorded at the beginning of each track, with the purpose to help the calibration of the smartphone. After the first initial trials, and a look at those data, it was decided to cut off the first part (35 seconds) of each track. That has led some improvement to the results.

The network has been composed initially by the following layers :

```
layer1 = tf.keras.layers.Conv2D(32,(3,3),activation = 'relu',input shape =(12,50,1))
layer2 = tf.keras.layers.MaxPooling2D((2,2))
layer3 = tf.keras.layers.Flatten()
layer4 = tf.keras.layers.Dense(96,activation = 'tanh')
layer5 = tf.keras.layers.Dense(3,activation='tanh')
```

Different activation functions and varying numbers of neurons in dense layers were experimented with during the course of this work.

For training and validation, a dataset consisting of 40 tracks was prepared, with one track off allocated for validation. The data batch was shuffled, and training was performed for a maximum of 600 epochs, with a callback function to halt training if the loss did not improve for 20 consecutive epochs. Tensorflow 2.11 with Python interface, Adam optimizer, and mean squared error (MSE) loss function were used. The training process took approximately 20 minutes per iteration on a laptop with Intel Core i7-8550U CPU, 256 GB SSD, and 16.0 GB memory.

The validation set included 9 tracks from the IPIN 2019 dataset, with some tracks having floor transitions. The model's performance was evaluated on all tracks, including the validation set, using measures like standard deviation, percentile, root-mean-square error, and displacement. Integration was necessary to obtain the position, and mean square root was calculated to assess position accuracy.

3. Results and Experiments

In our experiments, we explored variations in neural network configuration, adjust-ing the number of nodes from 12 to 576. Notably, reducing network size led to longer training times and decreased performance. Increasing the number of nodes in the first layer up to 576 did not yield significant improvements. Drawing from '90s experiences, we introduced a second layer to the network, resulting in faster training and notable performance enhancements. Further improvements were achieved by changing the activation function to 'tanh'.

Even with a modest network complexity (24 nodes in first layer,12 in second) we are able to get a satisfactory result on one track, but failing to generalize to other tracks.

Increasing the number of nodes, gave us a more uniform results among tracks, sometime really good on a single track; the issue with generalization persisted.

To enhance the outcomes, we increased the number of nodes in the network (64, 96, 192, 384), resulting in some improvements but not significant enhancements. Adding a second layer improved results, although it increased training time. While the CNN showed more consistent performance across tracks, it still faced challenges in generalization.

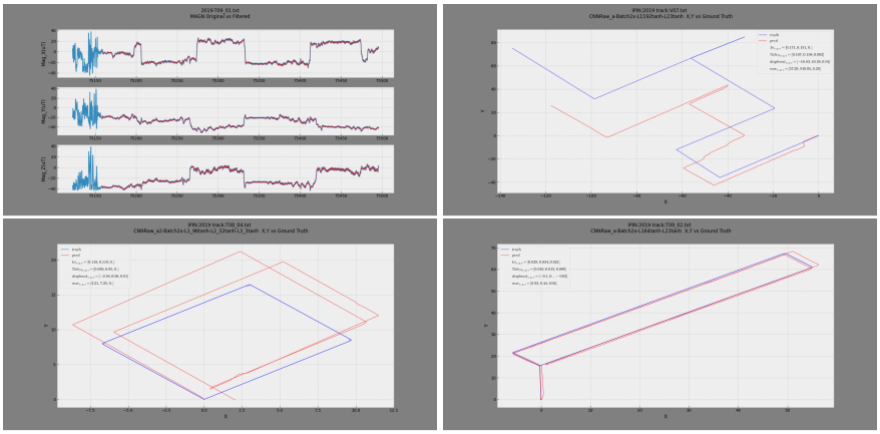

Fig. 2 Examples of input, typical prediction and Over fitting.

We observed unusually good results that raised suspicions about their accuracy. The first result was achieved by training the network multiple times on the same track, yielding remarkable performance on that specific track. The second result was obtained by training the network on a batch for two iterations, occasionally resulting in impressive performance. However, when attempting to generalize to other input tracks, the results consistently fell short of expectations (Fig. 2).

4. Conclusion

We aimed to use neural networks as a sensor fusion algorithm for indoor localization, which is a real-world problem. Recent hardware advancements enable the employment of powerful neural networks even in modern smartphones. The initial step focused on evaluating the effectiveness of convolutional neural networks (CNNs) as a sensor fusion algorithm.

A significant effort was dedicated to data preprocessing, involving the transformation of raw data from the IPIN dataset into a usable format. This

included re-constructing the ground truth data and converting sensor inputs into CNN-friendly images.

The selected CNN approach yielded varying outcomes and it lefts room for improvement. It showed satisfactory performance on specific tracks but struggled with generalization to new tracks. Converting sensor data into images for preprocessing led to a loss of important information and hindered the model's ability to make accurate predictions. The model occasionally is able to compensate for drift effects in the inertial sensors. The transformation process itself resulted in a significant degradation of the original data, reducing prediction reliability. To overcome these challenges, alternative methods, such as recurrent neural networks (RNNs) or LSTM (Long short-term memory), should be explored preserving the inherent characteristics of a dynamical system, thus enhancing generalization capabilities. To improve the model's generalization capabilities, it might be beneficial to expand the dataset used for training and that means to use the data already available from the other IPIN contest..

References

Antsfeld, L., Chidlovskii, B., & Sansano-Sansano, E. (2020). *Deep Smartphone Sensors-WiFi Fusion for Indoor Positioning and Tracking*. https://arxiv.org/abs/2011.10799v1

Bishop, C.M. (2006). *Information Science and Statistics* https://github.com/peteflorence/MachineLearning6.867/blob/master/Bishop/Bishop%20-%20Pattern%20Recognition%20and%20Machine%20Learning.pdf

Hu, M., Zhai, G., Li, D., Fan, Y., Duan, H., Zhu, W., & Yang, X. (2018). *Combination of near-infrared and thermal imaging techniques for the remote and simultaneous measurements of breathing and heart rates under sleep situation*. PLOS ONE, 13(1), e0190466. https://doi.org/10.1371/journal.pone.0190466

Li, M., Lu, Y., Cao, S., Wang, X., & Xie, S. (2023). *A Hyperspectral Image Classification Method Based on the Nonlocal Attention Mechanism of a Multiscale Convolutional Neural Network*. Sensors, 23(6), 3190. https://doi.org/10.3390/s23063190

Moayeri, N., Ergin, M. O., Lemic, F., Handziski, V., & Wolisz, A. (2016). PerfLoc (Part 1): *An extensive data repository for development of smartphone indoor localization apps*. 2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 1–7. https://doi.org/10.1109/PIMRC.2016.7794983

Potorti, F., Park, S., Crivello, A., Palumbo, F., Girolami, M., Barsocchi, P., Lee, S., Torres-Sospedra, J., Ruiz, A. R. J., Perez-Navarro, A., Mendoza-Silva, G. M., Seco, F., Ortiz, M., Perul, J., Renaudin, V., Kang, H., Park, S., Lee, J. H., Park, C. G., … Tsao, Y. (2020). *The IPIN 2019 Indoor Localisation Competition—Description and Results*. IEEE Access, 8, 206674–206718. https://doi.org/10.1109/ACCESS.2020.3037221

UJIIndoorLoc. (n.d.). Retrieved October 20, 2023, from https://archive.ics.uci.edu/dataset/310/ujiindoorloc

Long-Term Wi-Fi fingerprinting dataset and supporting material. (n.d.). (2014) https://doi.org/10.5281/ZENODO.1309317