User Data Selection using CNN-Feature Extractor for Fingerprint Localization

Yohei Konishi¹, Satoru Aikawa¹, a), and Shinichiro Yamamoto¹
¹ Graduate School of Engineering, University of Hyogo, 2167 Shosha, Himeji-city, Hyogo 671-2280, Japan
a) aikawa@eng.u-hyogo.ac.jp

Abstract: This paper examines a method for fingerprint indoor localization that employs CNN. CNN is trained using AP information. The estimation accuracy of CNN improves as the number of AP information increases. However, gathering AP information is expensive. The problem can be solved using UD (User Data). The UD is unlabeled data because the measuring method does not know the exact location of the user. As a result, we can perform semi-supervised learning with the estimation result as the correct label. In this paper, we propose a method for selecting UD using a CNN-feature extractor.

Keywords: Fingerprint, indoor localization, CNN, semi-supervised learning

Classification: Navigation, guidance and control systems

References

[1] Satoru Aikawa, et al., “WLAN Finger Print Localization using Deep Learning,” IEEE APCAP Aug. 2018. DOI:10.1109/APCAP.2018.8538306.

[2] Linchen Xiao, et al., “A Deep Learning Approach to Fingerprinting Indoor Localization Solutions,” International Telecommunication Networks and Applications Conference (ITNAC), Nov. 2017 DOI: 10.1109/ATNAC.2017.8215428.

[3] Ayush Mittal et al., “Adapting Convolutional Neural Networks for Indoor Localization with Smart Mobile Devices,” GLSVLSI’18, Chicago, IL, USA, May 23–25, 2018. DOI: 10.1145/3194554.3194594

[4] Yusuke Miyamoto, et al., “User data selection scheme to reduce database update errors for fingerprint localization,” IEICE Communication Express, vol.10, no5(2021), 343–348. DOI: 10.1587/comex.2021XBL0051

[5] Satoru Aikawa, et al., “CNN Localization Using AP Inverse Position Estimation” 2019 IEEE Conference Antenna Measurements and Applications(CAMA2019), OCTOBER23-25,2019. DOI:10.1109/CAMA47423.2019.8959663

[6] Hideki Nakayama, “Image feature extraction and transfer learning using deep convolutional neural networks,” IEICE Technical Report, vol. 115, no. 146, SP2015-45, pp. 55–59, July.2015 (In Japanese).
1 Introduction

The importance of location information has recently grown. Navigation systems based on mobile devices such as smartphones have become increasingly popular. Because the satellite signal is blocked, the GPS method cannot estimate the exact location in indoor facilities.

As an indoor location estimation method, fingerprint localization based on RSSI (Received Signal Strength Indicator) of Wi-Fi has been proposed. Machine Learning methods, particularly CNN, are used to fingerprint localization with high accuracy [1,2,3]. CNN is trained using AP information with correct labels. The AP information consists of the AP identifier and RSSI of the observed AP at the map coordinates. In general, the more training data with correct labels is used for machine learning, the higher the estimation accuracy. However, measuring a large amount of AP information with correct labels is expensive. The problem can be solved using UD (AP information users measured) [4]. The UD is unlabeled data because the UD measuring method does not know the exact location of the user. We use semi-supervised learning with estimated UD result as the correct label. The estimated result, however, may be incorrect. As a result, it is necessary to select UD that is estimated correctly and use it for CNN training. In this study, we propose a UD selection method for CNN training, and the effect is demonstrated using data collected in a building.

2 Fingerprint localization

2.1 Indoor Fingerprint localization using CNN

This section describes the use of CNN for indoor fingerprint localization. CNN is primarily used to classify images. First, the coordinates in the location estimation area are arbitrarily set. The coordinates are referred to as preset coordinates from now on. The following step is to measure the AP information, also known as preset-AP information. The AP information measured at the preset coordinates is referred to as the preset-AP information. Then, using preset information, input images for CNN are generated. The observed AP is first arranged on a 2D image. To create the input image, the RSSI values obtained from the AP are treated as pixel values. The image that is created is used as training data for CNN [5]. When performing the indoor localization, a user’s terminal measures AP information as UD (User Data) and feeds the resulting image to CNN. The number of neurons in CNN’s output layer is equal to the number of preset coordinates in this study. The final layer, which employs the Softmax function, returns the user’s existence probability for each coordinate. Finally, the coordinate with the greatest probability is chosen as the estimated result.
2.2 Semi-supervised learning using UD

The more preset-AP information measured in a CNN-based localization method, the better the estimation accuracy. However, measuring a large amount of preset-AP information is expensive. On the other hand, if there are enough navigation system users, it is possible to collect a large amount of UD. To estimate the user's location, a CNN trained with the preset-AP information is used. The estimated result is assumed to be the correct UD label and UD can be used to train CNN. Using both preset-AP information and the UD, the proposed method can perform semi-supervised learning. Semi-supervised learning is a training method that uses both labeled and unlabeled data. However, the labels assumed for the UD could be incorrect. As a result, if all the UD are used, the estimation accuracy of the CNN may not be improved. At the preset coordinate, the preset-AP information is measured. UD, on the other hand, is measured at any point where the user is present. As a result, if the user is far from the preset coordinates, the estimated results are more likely to be incorrect.

As a result, it is critical to select UD that is correctly estimated. The chosen UD is used for semi-supervised learning. Semi-supervised learning with the chosen UD is referred to as UD-learning from now on.

3 UD selection for semi-supervised learning

3.1 Feature value extracted from CNN

CNN is employed in order to extract the feature value of input images. The trained CNN is fed an image. As the feature value of the input image, the output values of each neuron in the middle layer are extracted. If the feature values of the two input images are similar, they are close [6]. UD is chosen in a proposed scheme if the feature values of UD and preset-AP information are close.

Fig. 1(a). depicts the CNN configuration used in this study. The output values in the middle layer before the output layer of the CNN were used as feature values of the input image to select the UD.

3.2 UD selection using CNN-feature extractor

The UD closest to the preset coordinates should be added to the CNN's training data. In contrast, if the measured position is significantly different from the estimated coordinates, the UD should not be included in the training data. The feature value extracted from the preset-AP information is similar to the feature value of the UD measured near the preset coordinate. The feature value of the UD measured away from the preset coordinates, on the other hand, is different.

First, as described in Section 2.1, preset-AP information is measured. CNN is trained using images generated from the preset-AP information. Following the training of CNN, the feature values are extracted from all of the training data. The extracted feature values of all preset coordinates are then gathered as illustrated in Fig. 1(b), and the average value is used as the feature value for each preset coordinate.

The location estimation for the UD using CNN selects a coordinate. At that point, the UD feature value is extracted from CNN. The feature value of UD is then
compared to the preset coordinate selected by location estimation, as shown in Fig. 1 (c). If the two feature values are similar, the UD is used as CNN training data. In [7], Euclidean distance was used to calculate the similarity between two feature values, as shown in Eq. (1). The feature values are $x$ and $y$, and the Euclidean distance is $d$. The smaller the Euclidean distance value, the closer the two feature values are. In this study, Cosine similarity is also used to calculate the similarity, as shown in Eq. (2). As Cosine similarity approaches one, the two feature values are more similar.

\[
d(x, y) = \sqrt{(x_1 - y_1)^2 + \cdots + (x_n - y_n)^2}
\]

(1)

\[
\cos(x, y) = \frac{x \cdot y}{\|x\|\|y\|}
\]

(2)

(a) Extracting feature value using CNN

(b) Feature value at each coordinate

(c) comparison to feature value of UD

**Fig. 1.** Method of UD selection
3.3 Verification of UD-learning

In the verification environment as shown in Fig.2, UD-learning was implemented. TensorFlow is used to implement this verification and the estimation result is slightly altered because initial values of CNN’s weights are determined randomly. As a result, this verification was carried out ten times with the average value of each estimation result shown below.

![Measurement environment](image)

(a) Measurement environment

(b) Measurement conditions

| Measurement place            | University building corridor |
|------------------------------|------------------------------|
| The number of coordinates   | preset coordinates: 11      |
|                              | UD coordinates: 121         |
| Coordinate interval         | preset: 3 m                 |
|                              | UD: 0.25 m                  |
| The number of measurements  | 40 for each preset coordinate|
|                              | 100 for each UD coordinate  |

Fig.2. verification environment

Fig.3(a) depicts the average error with respect to the Euclidean distance and Cosine similarity thresholds. The CNN trained solely with preset-AP information is depicted as “only preset” in Fig. 3(a). As shown in Fig. 3(a), UD-learning improves estimation accuracy over “only preset.” In this verification, the average error was 1.74m when the Euclidean distance threshold was set to 6.5. Furthermore, the average error was 1.69 m when the Cosine similarity threshold was set to 0.95. As a result, using Cosine similarity to select UD is slightly better than using Euclidean distance. The number of chosen UD increases as the threshold of Euclidean distance is set larger or that of Cosine similarity is smaller. However, the average error value is not significantly lower than only preset. The reason for this is most likely that by setting a high threshold, many UD far from the preset coordinates are used for UD-learning.

Fig. 3(b) depicts the CDF of the estimation error for the test data with a Euclidean distance threshold of 6.5 and a threshold of Cosine similarity set to 0.95. Fig.3(b) shows that the estimation error of UD-learning is smaller than “only preset”. Furthermore, UD-learning estimation accuracy using Cosine similarity is slightly better than using Euclidean distance. The effectiveness of the proposed UD selection method based on Euclidean distance or Cosine similarity was confirmed by these experimental results. Furthermore, as shown in Fig 3, Cosine similarity
outperforms Euclidean distance.

![Graph](image)

(a) Estimation accuracy with changing threshold

![Graph](image)

(b) CDF of UD-learning

**Fig. 3.** Verification result

### 4 Conclusion

The more preset-AP information measured in fingerprint localization using CNN, the better the estimation accuracy. However, measuring a large amount of preset-AP information is expensive. We proposed a method for selecting UD for semi-supervised learning using an extracted feature value from CNN in this study. The similarity of feature values between preset-AP information and UD was calculated using Euclidean distance and Cosine similarity. UD-learning can improve estimation accuracy more than CNN trained solely with preset-AP information. Furthermore, based on these experimental results, we confirmed that using Cosine similarity is superior to using Euclidean distance in UD selection.

### Acknowledgments

This work was supported by JSPS KAKENHI Grant Number 21K04065.