CNN based Indoor Localization using RSS Time-Series

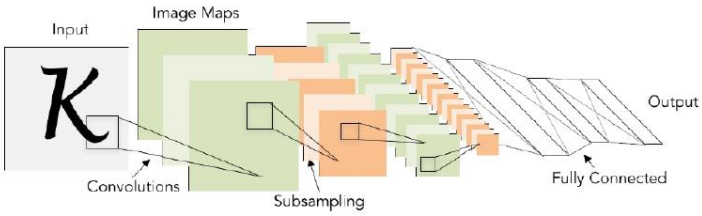
Yang Yang Meng Wei

Contents

- Introduction of CNN
- System Architecture
- Four Approaches
- Data Processing
- Achievements
- Problems

Basic Concepts of CNN

The Basic Structure of CNN

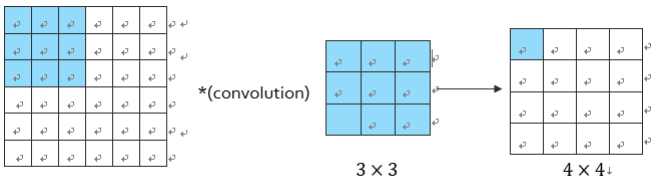


Why CNN?

Temporal Dependency

Processing data has grid-like topology. Such as time-series data and image data.

Convolution



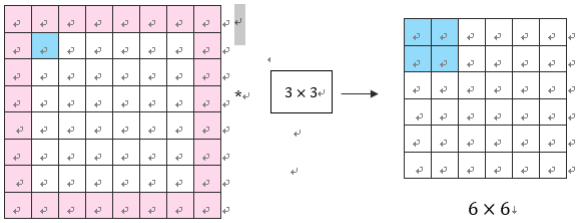
The size of filter is 3×3 and the possible combination of 3×3 in 6×6 is 4×4. In general, if we have a $n \times n$ and a $f \times f$ filter, the dimension of the output will be $(n - f + 1) \times (n - f + 1)$.

Disadvantage

1. The image will shrink after every convolutional operator;
2. Pixels on the corners or on the edges will be throwed away;

Padding

To avoid the disadvantages of convolution, padding the image.



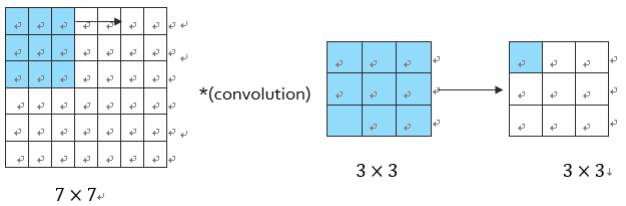
Valid convolution: no padding
Same convolution: pad so that output size is the same as the input size

$$p = \frac{f-1}{2}$$

(#the filter is always odd)

Strided Convolution

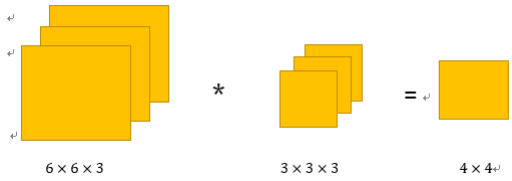
Stride=2;



In general, the output dimension is $\frac{n+2p-f}{s} \times \frac{n+2p-f}{s}$

A*(B*C)=(A*B)*C

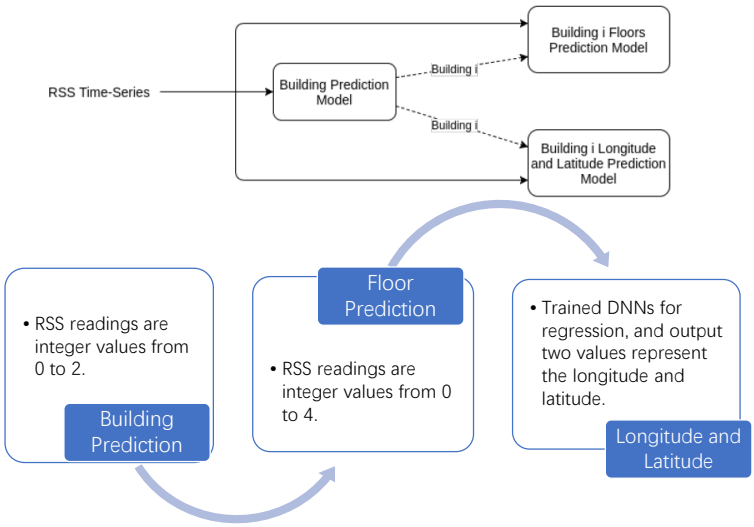
Convolutions or RGB Images



$6 \times 6 \times 3$ is the $\mathbf{height} \times \mathbf{width} \times \mathbf{channel}$
Different parameters get different features detector

In general, the output dimension can be got by:
$\mathrm{n} \times \mathrm{n} \times n_c * \mathrm{f} \times \mathrm{f} \times f_c = (\mathrm{n} - \mathrm{f} + 1) \times (\mathrm{n} - \mathrm{f} + 1) \times n_c'$
n_c'=the number of the filter

System Architecture Using Time-Series



Four Approaches

Floor Prediction Accuracy

Input	Convolutional Layers	Hidden Layers Units	Accuracy
Single RSS Vector	-	128, 128, 128	99.9 %
Averaged RSS Time-Series	-	256, 256, 256	97.9%
Concatenated RSS Time-Series	-	8, 8, 8	100%
RSS Time-Series Image	1 layer with 3 out channels and 2 × 2 kernel	8, 8	100%

The Fourth Approach

$$\begin{bmatrix} RSS_{1,1} & RSS_{1,2} & \dots & RSS_{1,T} \\ RSS_{2,1} & RSS_{2,2} & \dots & RSS_{2,T} \\ \vdots & \vdots & \vdots & \vdots \\ RSS_{N,1} & RSS_{N,2} & \dots & RSS_{N,T} \end{bmatrix}$$

T: RSS vectors, equals 10 in this model
N: RSS values from the N WLAN access points, equals 520 in UJIIndoorLoc dataset

Data processing

drawing a grid of D×D (m2) cells on the area

$$\underbrace{\begin{matrix} A_{11} & \dots & A_{1T} \\ \vdots & \ddots & \vdots \\ A_{D1} & \dots & A_{DT} \end{matrix}}_{D} \Big\} D$$

assigning each record from the 19937 records to a cell based on its longitude and latitude
sorted them based on their timestamp and put them in groups of T records
Each cell contains a $\mathrm{T} \times N$ matrix

$$\begin{bmatrix} RSS_{1,1} & RSS_{1,2} & \dots & RSS_{1,T} \\ RSS_{2,1} & RSS_{2,2} & \dots & RSS_{2,T} \\ \vdots & \vdots & \vdots & \vdots \\ RSS_{N,1} & RSS_{N,2} & \dots & RSS_{N,T} \end{bmatrix}$$

#Each group is less than or equal to S seconds

- After training, the best parameter to get the largest data set is D = 3 m, T = 10, and S = 60 seconds
- gives 5484 records of 10 consecutive readings from the same cell
- Each group of T records can then be used as an RSS time-series sample

Achievements

The Localization Accuracy for Four Models

	Input	Building Prediction Accuracy	Floor Prediction Accuracy	Localization Mean Error (m)
Traditional Methods	Single RSS Vector	100%	91.42%	10.25
	Averaged RSS Time-Series	99.91%	96.90%	4.39
	Concatenated RSS Time-Series	100%	99.66%	3.87
	RSS Time-Series Image	100%	100%	2.77

Improved the localization performance successfully.
Time-series of RSS readings is more robust to estimate location.
Time-series data affects the model accuracy.

Problems

- The size of RSS image is 520*10. The dataset is too small to train the model.
- The measurement method is not convenient.
- How to choose the value of T is not given in detail, and the source code.

END