# Kalman Filter와 CNN을 이용한 거리 측정 최적화 방안 연구

수학과 201621120 최동헌
수학과 201621136 이재협
수학과 201621138 허창현
수학과 201621148 백현규

# Contents

1 이론적 배경

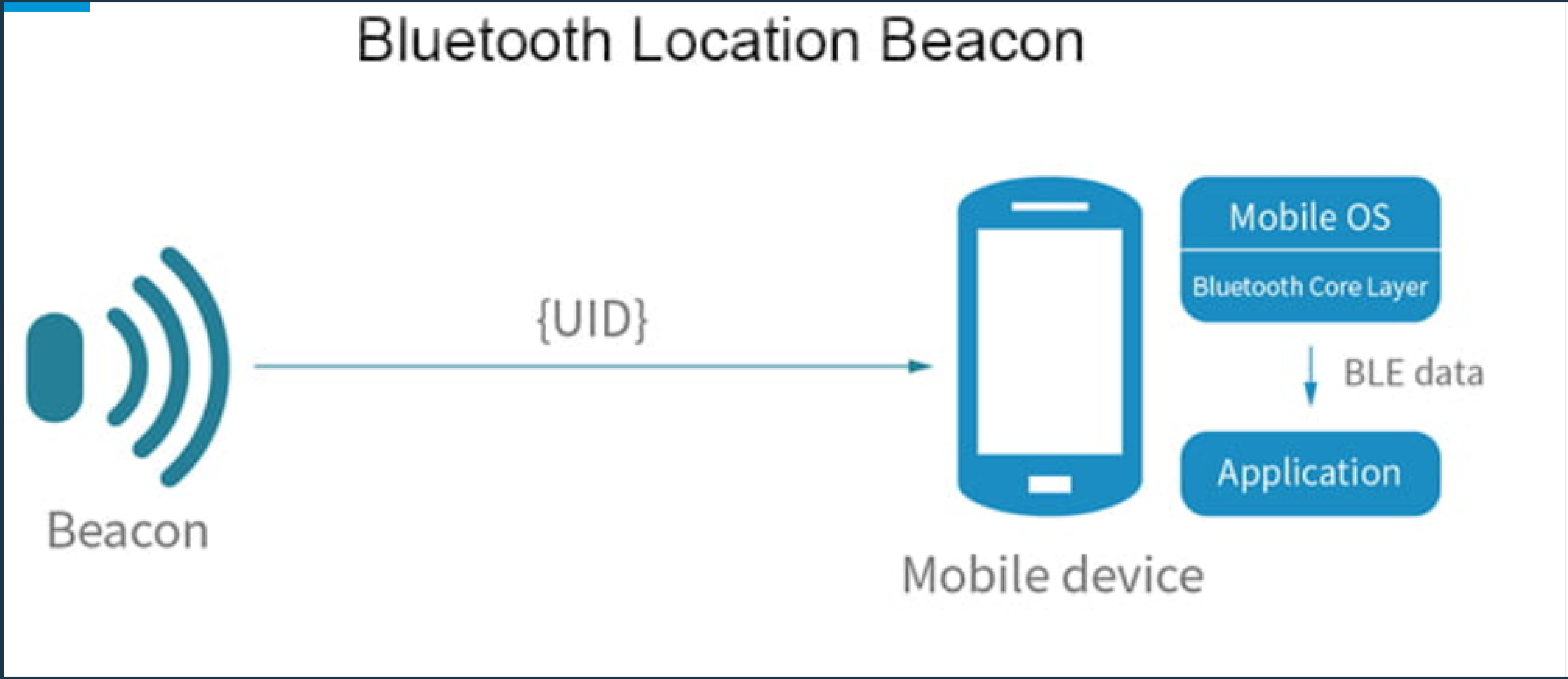
- Blutooth Beacon
- Kalman Filter
- Convolutional Neural Network (CNN)

2 연구 방법

3 결과

4 참고문헌

# 1. Blutooth Beacon and RSSI



# 1. 이론적 배경 - Blutooth Beacon

$$RSSI = -10n\log_{10}(d) + A,$$

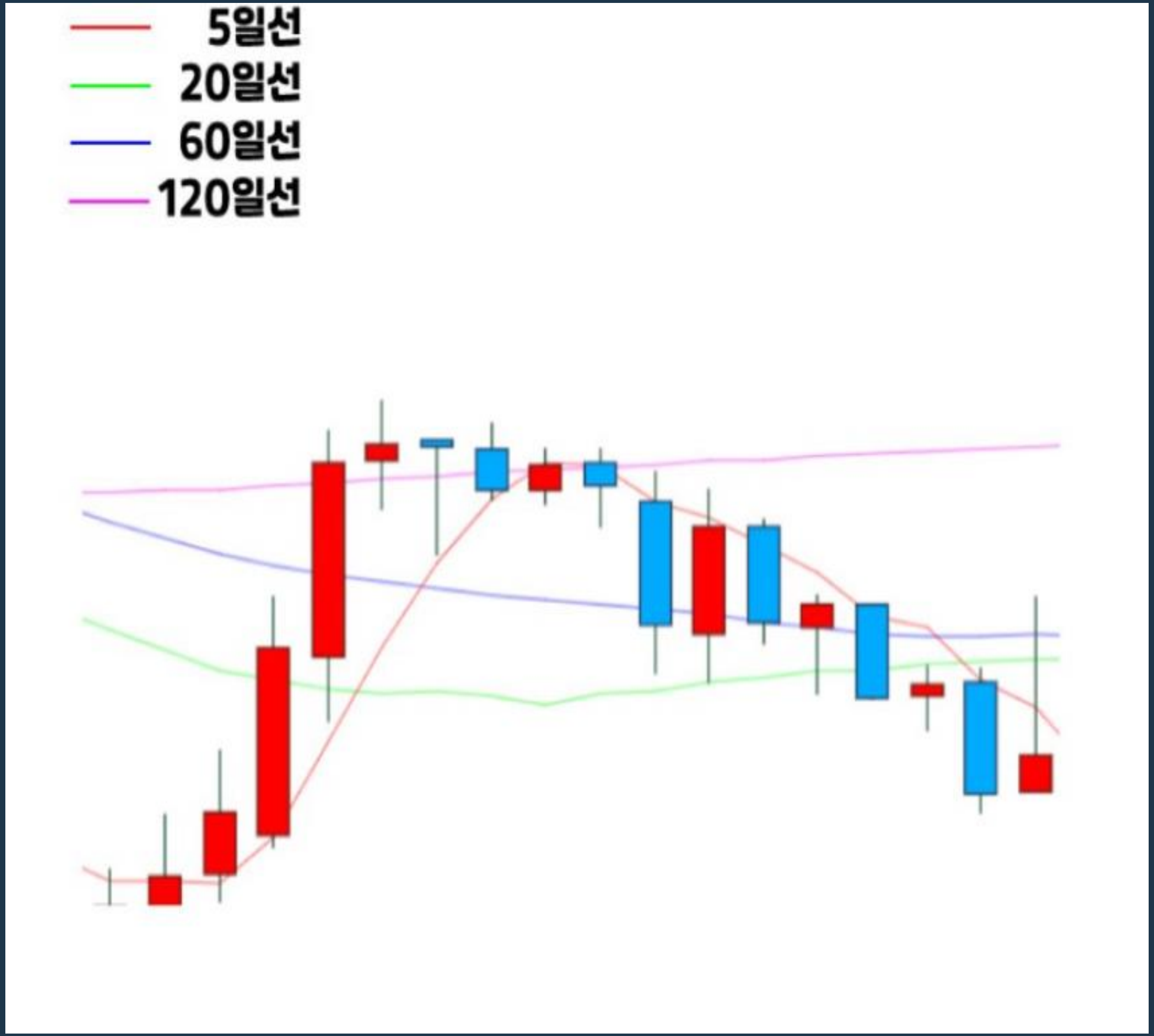
- RSSI를 통해 거리를 측정하는 데 노이즈로 인해 정확도가 낮음.
- RSSI값이 1만 변해도 거리는 몇 m씩 변함
- 정확도를 높일 수 있는 방법을 연구.

# 1. 이론적 배경 - Kalman Filter

$$\bar{x}_k = \alpha \bar{x}_{k-1} + (1-\alpha) x_k$$

- 1차 저주파 필터
- α는 weight를 뜻함.

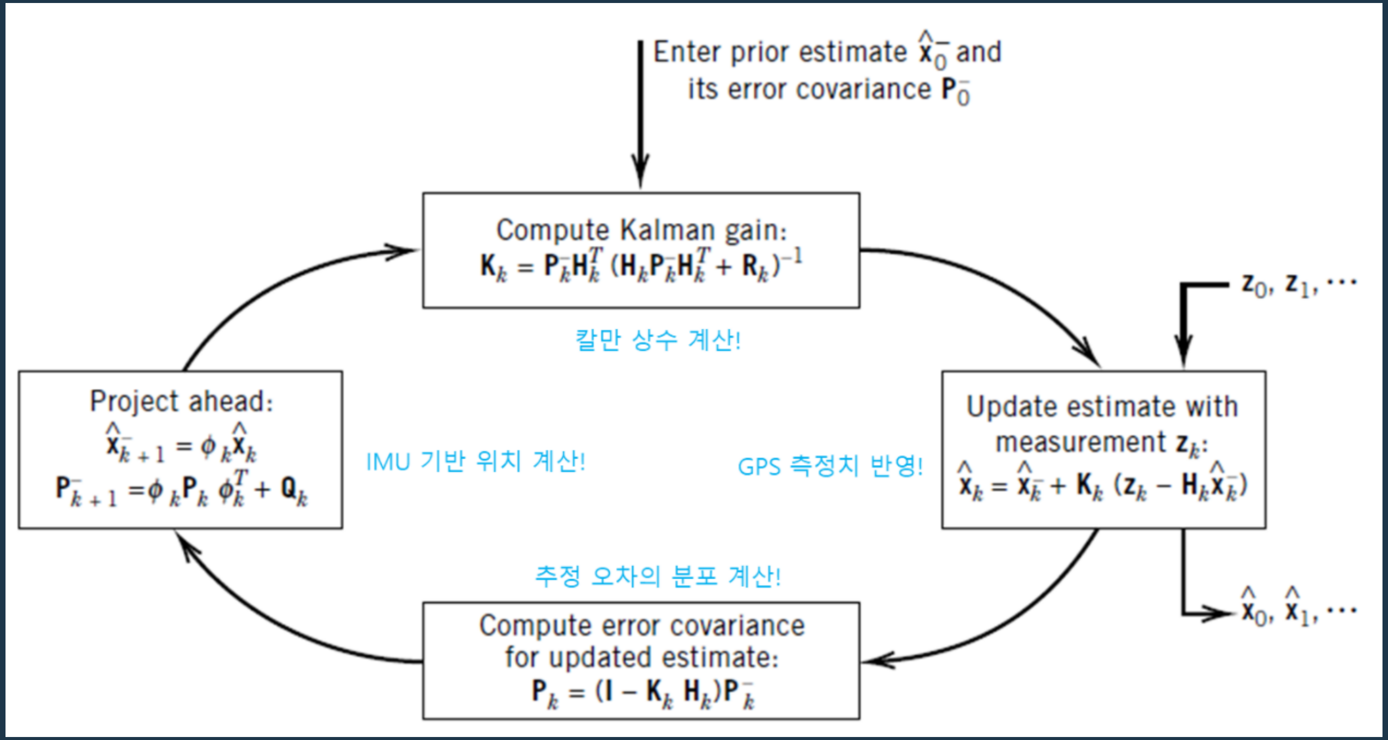
# 1. 이론적 배경 - Kalman Filter




- 일정 기간 주가의 산술 평균값을 이어서 만든 선
- 주식에서 일정기간의 평균을 의미 있는 보조 지표로 활용

칼만 필터

# 1. 이론적 배경 - Kalman Filter



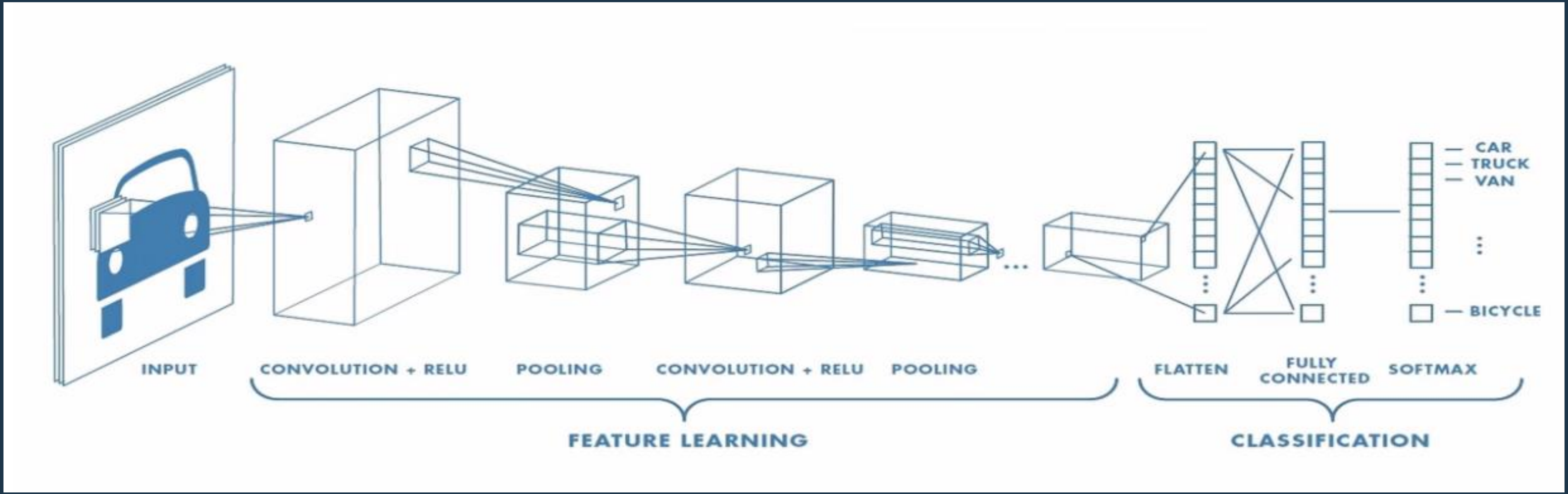

1) 한 센서로 추정값 계산

2) 추정 오차를 최소화 하는 칼만 상수 계산

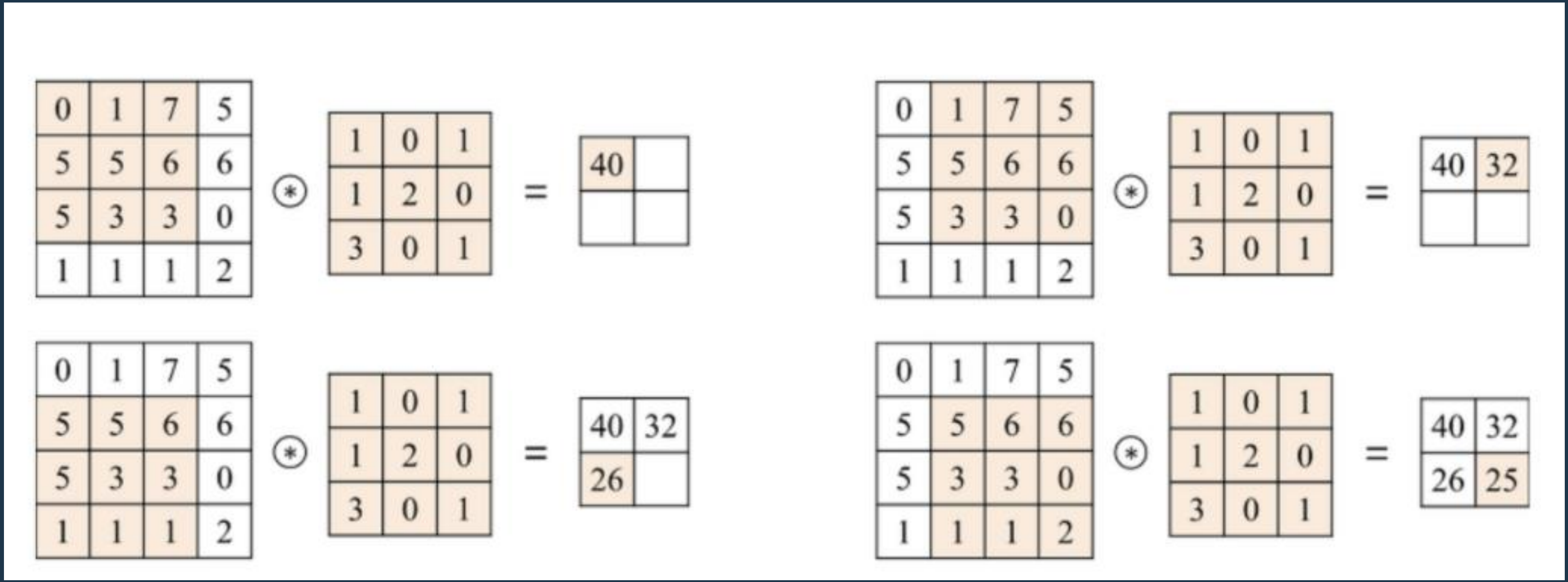
3) 다른 센서의 값을 칼만 상수를 통해 반영하여 보정

4) 보정하고 나서 줄어든 추정 오차의 분포 계산

# 1. 이론적 배경 - Convolutional Neural Network



# 1. 이론적 배경 - Convolutional Neural Network



# 1. 이론적 배경 - Convolutional Neural Network

## 1) Stride

$$\begin{bmatrix} 0 & 1 & 7 & 5 \\ 5 & 5 & 6 & 6 \\ 5 & 3 & 3 & 0 \\ 1 & 1 & 1 & 2 \end{bmatrix} \circledast \begin{bmatrix} 1 & 0 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 15 & 18 & 25 \\ 16 & 14 & 9 \\ 8 & 6 & 8 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 1 & 7 & 5 \\ 5 & 5 & 6 & 6 \\ 5 & 3 & 3 & 0 \\ 1 & 1 & 1 & 2 \end{bmatrix} \circledast \begin{bmatrix} 1 & 0 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 15 & 25 \\ 8 & 8 \end{bmatrix}$$

## 2) Padding

$$\begin{bmatrix} 0 & 1 & 7 & 5 \\ 5 & 5 & 6 & 6 \\ 5 & 3 & 3 & 0 \\ 1 & 1 & 1 & 2 \end{bmatrix} \circledast \begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & 1 \\ 1 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 41 & 33 \\ 25 & 23 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 7 & 5 & 0 \\ 0 & 5 & 5 & 6 & 6 & 0 \\ 0 & 5 & 3 & 3 & 0 & 0 \\ 0 & 1 & 1 & 1 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \circledast \begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & 1 \\ 1 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 26 & 42 & 55 & 35 \\ 34 & 41 & 33 & 28 \\ 18 & 25 & 23 & 14 \\ 3 & 9 & 8 & 8 \end{bmatrix}$$

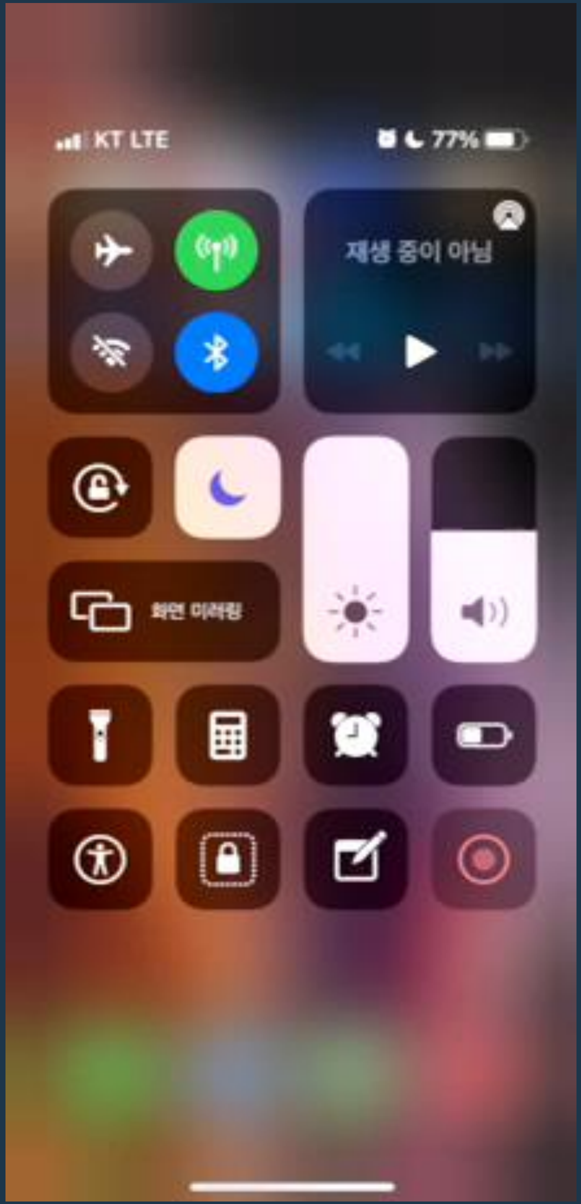
# 2. 연구 내용 - RSSI 측정


측정된 RSSI값

| H | I | J | K | L | M | N | O | P | Q | R | S | T |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| major | 100 | minor | 3 | proximity | near | rssi | -70 | uuid | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |
| major | 100 | minor | 3 | proximity | near | rssi | -73 | uuid | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |
| major | 100 | minor | 4 | proximity | near | rssi | -61 | uuid | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |
| major | 100 | minor | 2 | proximity | near | rssi | -64 | uuid | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |
| major | 100 | minor | 3 | proximity | near | rssi | -66 | uuid | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |
| major | 100 | minor | 3 | proximity | near | rssi | -67 | uuid | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |
| major | 100 | minor | 2 | proximity | near | rssi | -62 | uuid | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |
| major | 100 | minor | 2 | proximity | near | rssi | -61 | uuid | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |
| major | 100 | minor | 2 | proximity | near | rssi | -60 | uuid | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |
| major | 100 | minor | 3 | proximity | near | rssi | -66 | uuid | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |
| major | 100 | minor | 3 | proximity | near | rssi | -67 | uuid | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |
| major | 100 | minor | 3 | proximity | near | rssi | -67 | uuid | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |
| major | 100 | minor | 2 | proximity | near | rssi | -61 | uuid | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |
| major | 100 | minor | 3 | proximity | near | rssi | -65 | uuid | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |
| major | 100 | minor | 2 | proximity | near | rssi | -64 | uuid | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |
| major | 100 | minor | 3 | proximity | near | rssi | -70 | uuid | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |
| major | 100 | minor | 3 | proximity | near | rssi | -70 | uuid | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |
| major | 100 | minor | 3 | proximity | near | rssi | -73 | uuid | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |
| major | 100 | minor | 4 | proximity | near | rssi | -61 | uuid | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |
| major | 100 | minor | 3 | proximity | near | rssi | -67 | uuid | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |
| | | | | | | | | id | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |
| | | | | | | | | id | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |
| | | | | | | | | id | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |
| | | | | | | | | id | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |
| | | | | | | | | id | e2c56db5-dffb-48d2-b060-d0f5a71096e0] | | | |

안드로이드 스마트폰이 받은 RSSI 데이터를 정리하기 위해 만들어진 어플리케이션임.

# 2. 연구 내용 - RSSI 측정

# 2. 연구 내용 - RSSI 측정

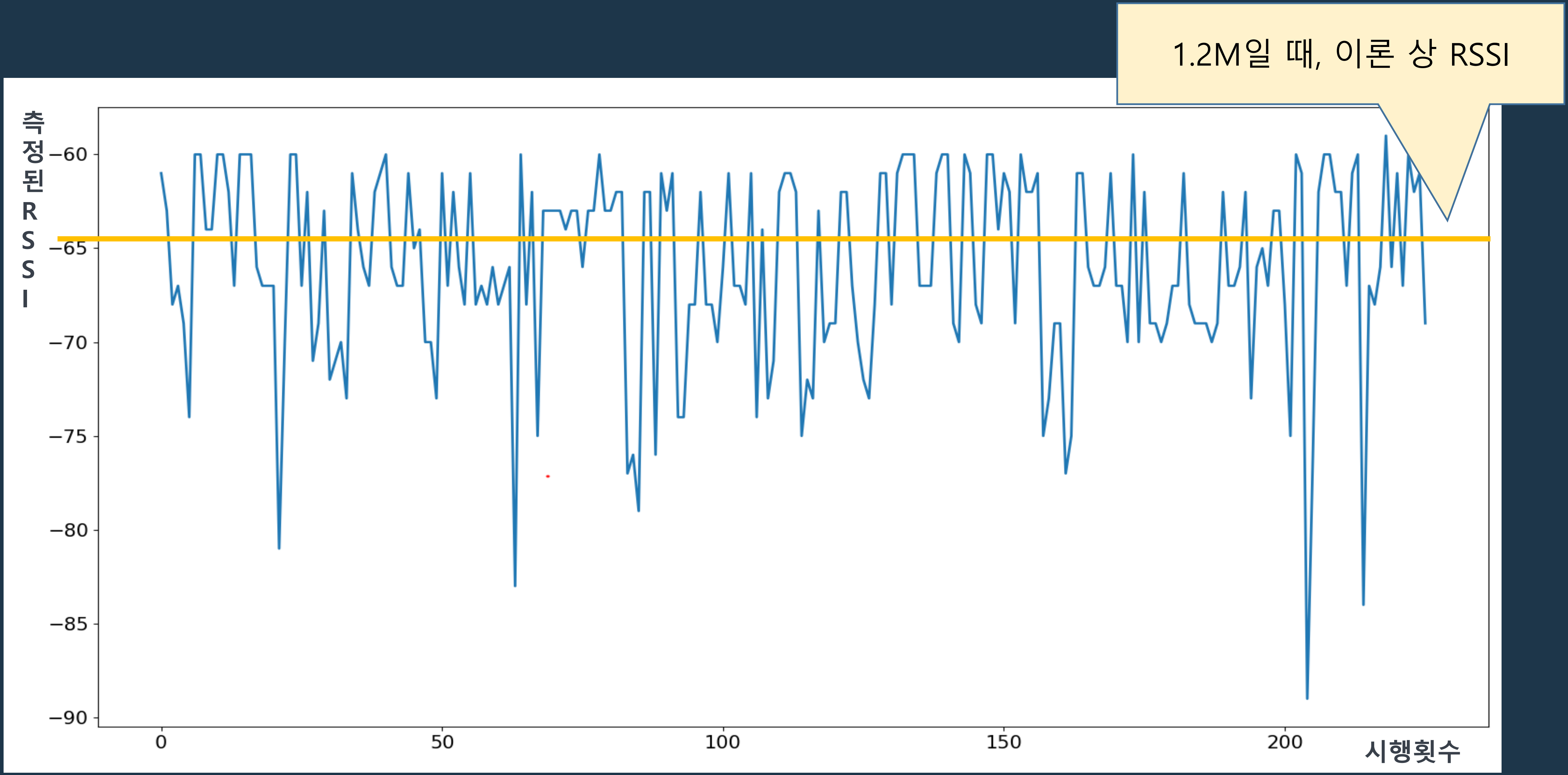
```python
class Beacon:
    x, y, N, M_Power = 0, 0, 0, 0
    mu_RSS, sigma_RSS = 0, 0
    dataset = []

    def __init__(self, x, y, N, M_Power):
        self.x = x
        self.y = y
        self.N = N
        self.M_Power = M_Power

    def RSSI_to_distance(self, RSSI):
        return math.pow(10, ((self.M_Power - RSSI) / (10 * self.N)))

    def Dist_to_RSSI(self, Dist):
        return ((-10) * self.N * math.log(Dist, 10) + self.M_Power)

    def set_mu_sigma_RSS(self, a, b, c):
        temp = []
        temp.append(a)
        temp.append(b)
        temp.append(c)
        self.mu_RSS, self.sigma_RSS = np.mean(temp), np.std(temp)

    def add_RSS(self, RSS):
        self.dataset.append(RSS)
        self.mu_RSS = np.mean(self.dataset)
        self.sigma_RSS = np.std(self.dataset)
```

1. RSSI를 통해서 거리를 구하는 함수
2. 거리를 이용해서 RSSI를 구하는 함수
3. 평균, 표준편차

# 2. 연구 내용 - RSSI 측정

| 실제 거리 | RSSI로 측정한 거리 | 오차율 |
| --- | --- | --- |
| 0.5m | 0.73 | 46% |
| 0.8m | 0.91 | 14% |
| 1.2m | 1.25 | 4% |
| 1.6m | 1.15 | 28% |
| 2.0m | 1.23 | 39% |
| 2.4m | 2.06 | 14% |
| 2.8m | 1.36 | 51% |

# 2. 연구 내용 - RSSI 측정



# 2. 연구 내용 - RSSI 측정

측정된 RSSI

N개의 RSSI 입력 받음

↓

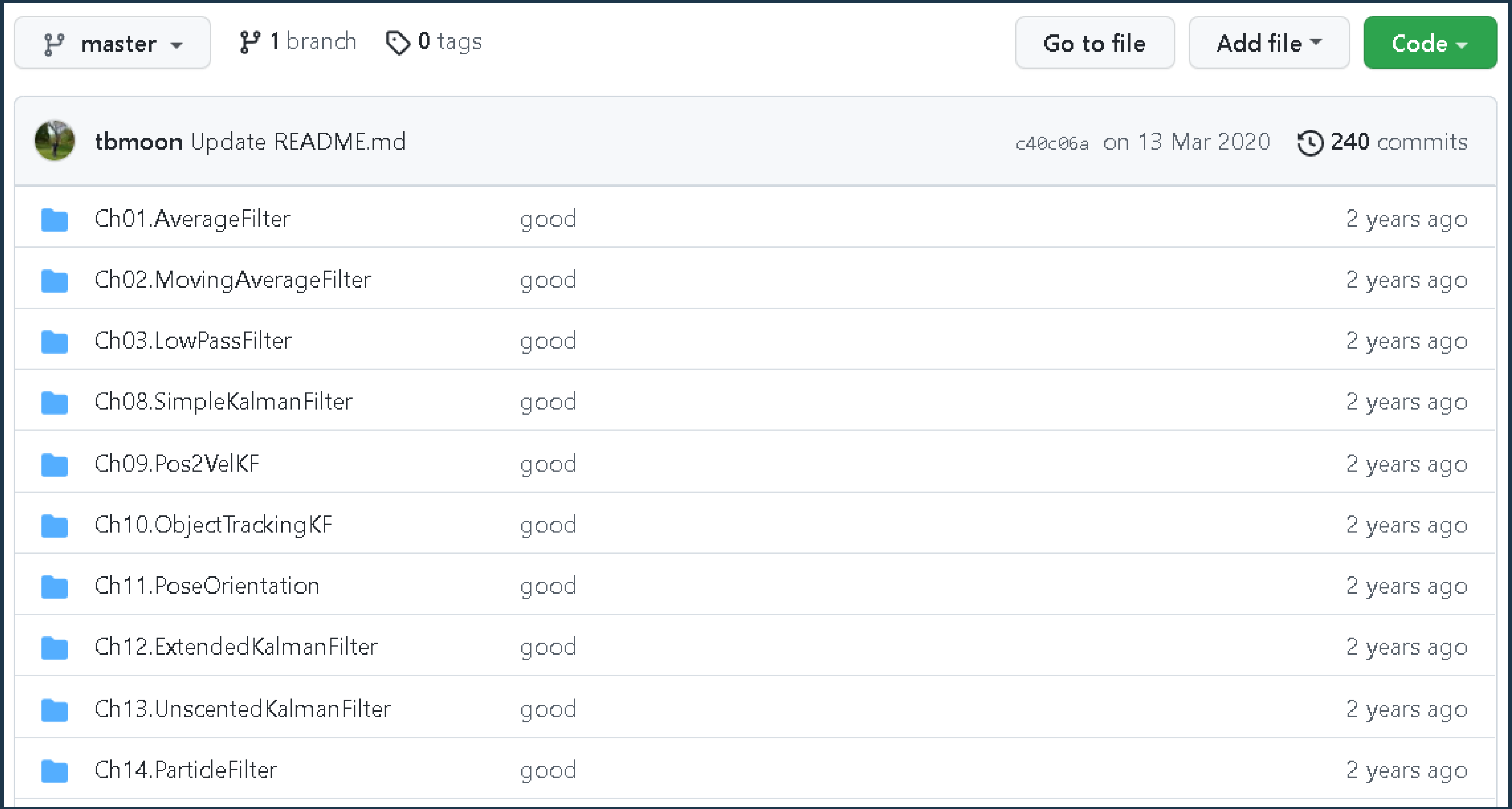
입력 받은 RSSI 값을 Kalman Filter로 1차 보정

↓

보정된 Kalman Filter 값을 CNN에 적용

시행횟수

# 2. 연구 내용 - Kalman Filter




(출처: https://github.com/tbmoon/kalman_filter)

시행횟수

# 2. 연구 내용 - Kalman Filter

TestSimpleKalman2.m | Kalman_RSS.m

```matlab
function rss = Kalman_RSS(arr)
%
%
persistent A H Q R
persistent y P
persistent flagRun


if isempty(flagRun)
  A = 1;    % 칼만 필터에서 필요한 parameter를 정의함.(A, H, Q, R)
  H = 1;

  Q = 1;
  R = 1000;

  length = size(arr,2)  % arr 함수의 열의 갯수를 반환
  cut_ratio = 0.10
  num_of_cut = fix(length * cut_ratio) % 전체 열 갯수 중에서 cut_ratio 비율 만큼 잘라냄.

  for a=1:num_of_cut % for문을 돌면서 최대, 최솟값을 없앰
      min_num = min(arr)
```

명령 창

```
ans =

  -66.8940
```

```python
def kalman_filter(z_meas, x_esti, P):
    """Kalman Filter Algorithm for One Variable."""
    # (1) Prediction.
    x_pred = A * x_esti
    P_pred = A * P * A + Q

    # (2) Kalman Gain.
    K = P_pred * H / (H * P_pred * H + R)

    # (3) Estimation.
    x_esti = x_pred + K * (z_meas - H * x_pred)

    # (4) Error Covariance.
    P = P_pred - K * H * P_pred

    return x_esti, P

# Input parameters.
time_end = 10
dt = 0.2

# Initialization for system model.
A = 1
H = 1
Q = 0
R = 4
# Initialization for estimation.
x_0 = 12  # 14 for book.
P_0 = 6
```

# 2. 연구 내용 - Kalman Filter

```
function rss = Kalman_RSS(arr)
%
%
persistent A H Q R
persistent y P
persistent flagRun


if isempty(flagRun)
  A = 1;     % 칼만 필터에서 필요한
  H = 1;

  Q = 1;
  R = 1000;
```

| 이론상 RSSI | -65.58 | |
|---|---|---|
| (Q,R) | 측정 RSSI 평균 | 표준편차 |
| (0,50) | -66.95174 | 1.293 |
| (0, 100) | -66.9471 | 1.33 |
| **(0, 1000)** | **-66.81694** | **0.915** |
| (1,50) | -67.07816 | 1.117 |
| (1, 100) | -67.02378 | 1.157 |
| (1, 1000) | -66.84224 | 1.711 |

# 2. 연구 내용 - Convolutional Neural Network

```python
def Conv1D_Fit_and_PlotWeights(model, X, y, epochs, n_weights, freq=20):
    w, loss, mae = [], [], []
    for r in range(epochs):
        history = model.fit(X, y, verbose=0)
        if r%freq==0:
            w.append(np.sort(model.layers[0].get_weights()[0].reshape(n_weights)))
            loss.append(history.history['loss'][0])

    w = np.array(w)
    fig, ax = plt.subplots(figsize=(8,4))
    epoch = np.arange(0,len(w))*20
    for n in range(n_weights):
        label = "w_{} -> {}".format(n, n+1)
        ax.axhline(n+1, c='gray', linestyle='--')

    return w
```

# 3. 결과

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 c1d (Conv1D)                (None, 206, 1)            5

=================================================================
Total params: 5
Trainable params: 5
Non-trainable params: 0
_________________________________________________________________
Input RSS: -60  -61 -59 -58 -58
47.528821885585785
```

# 3. 결과

| 실제 거리(m) | 추정 RSSI로 측정한 거리 | 오차율 |
|---|---|---|
| 0.5 | 0.475 | 5% |
| 0.8 | 0.858 | 7.25% |
| 1.2 | 1.245 | 3.75% |
| 1.6 | 1.732 | 8.25% |
| 2.0 | 2.14 | 7% |
| 2.4 | 2.7 | 12.5% |

# 3. 결과(한계점)

1) 제한된 Sample Data의 크기

2) 측정된 RSSI값 자체의 불안정성

3) 고정된 위치에서의 RSSI

# 4. 참고문헌


[1]김우찬,이청길,곽호영,“위치측정을위한비콘의RSSI 안정화”, 제주대학교, 2019

[2]박종형,박두익,염철민,강진수,원유재,“비콘의세기를이용한실내위치추적정확도개선에관한연구”, 충남대학교,2018

[3]김지성,김용갑,“RSSI 측정결과필터링을이용한거리계산보정알고리즘에관한연구”, IBC, 2017

[4]김지성,김용갑,황근창,“비콘의RSSI 특성을이용한실내위치추적시스템에관한연구”, IIBC, 2017

[5]김정진,조성욱,지영민,“CNN을이용한이미지분류”, 한국정보기술학회,2017

[6]최학영,“CNN기반결함검출”, 서경대학교대학원,2017

[7]이현수,“이미지컨텐츠검색을위한CNN의구조”, 중앙대학교대학원,2018

[8]김성필,2019, “칼만필터는어렵지않아:with MATLAB Examples”, 한빛아카데미