



버드스트라이크 및 태풍을 고려한 최적의 풍력발전소 위치 선정

LOWG
1103 김석현
1114 이동윤

Ocean ICT: > LOWG > 1. 탐구동기

지구온난화가 사회 문제로 대두됨에 따라 해상풍력발전에 대한 관심이 증가하고 있다. 하지만 기존에는 버드스트라이크 현상으로 인한 생태계 파괴와 태풍, 긴 송전거리의 영향으로 인한 경제적 손실이 크다는 문제점들이 있었다. 따라서 이번 연구에서는 철새와 태풍의 경로, 지면으로의 거리를 분석하여 이러한 문제점을 사전에 예방할 수 있는 최적의 풍력발전소 위치를 탐구해볼 것이다.

Ocean ICT: > LOWG > 2. 융합 분야

- 1 지구과학
- 2 태풍의 발생 원리에 대해 알아보고 태풍의 이동경로를 파악
- 3 철새가 이동하는 이유와 철새의 종류 및 철새가 이동하는 방법 탐구
- 4 철새의 이동경로 파악
- 5 수학
- 6 지면과의 거리를 측정할 때 피타고라스 정리 이용
- 7 가중치들을 합하여 최적의 풍력발전소 위치 계산

Ocean ICT: > LOWG > 3. 코드 설계

1 지도에서 바다 추출

```
import numpy as np
from matplotlib import pyplot as plt
import os

map_hgr = cv2.imread('map_hgr.jpg', cv2.IMREAD_COLOR)
map_rgb = cv2.cvtColor(map_hgr, cv2.COLOR_BGR2RGB)
hgr_low = np.array([200, 170, 110])
hgr_high = np.array([220, 220, 100])
map_mask = cv2.inRange(map_hgr, hgr_low, hgr_high)
map_rgb = cv2.bitwise_and(map_rgb, map_rgb, mask = map_mask)
map_rgb = map_rgb[1:227, 1:227]
```

6 풍량 +가중치 부여

```
wind_hgr = cv2.imread('wind_hgr.jpg', cv2.IMREAD_COLOR)
wind_hgr_b = wind_hgr[1:227, 1:227]
wind_rgb = cv2.cvtColor(wind_hgr_b, cv2.COLOR_BGR2RGB)
wind_low = np.array([100, 100, 100])
wind_high = np.array([100, 100, 100])
wind_mask = cv2.inRange(wind_hgr_b, wind_low, wind_high)
wind_rgb = cv2.bitwise_and(wind_rgb, wind_rgb, mask = wind_mask)
wind_arr = np.zeros((200, 227))
for k in range(200):
    for l in range(227):
        if wind_rgb[k, l, 0] <= 100:
            wind_arr[k, l] = 100
        elif wind_rgb[k, l, 1] <= 100:
            wind_arr[k, l] = 100
        elif wind_rgb[k, l, 2] <= 100:
            wind_arr[k, l] = 100
        else:
            wind_arr[k, l] = 0
```

12 철새 이동경로 -가중치 부여

```
bird_map_hgr = cv2.imread('bird_map.jpg', cv2.IMREAD_COLOR)
bird_map_rgb = cv2.cvtColor(bird_map_hgr, cv2.COLOR_BGR2RGB)
b_hgr_low = np.array([10, 10, 10])
b_hgr_high = np.array([100, 100, 100])
bird_mask = cv2.inRange(bird_map_hgr, b_hgr_low, b_hgr_high)
bird_map_rgb = cv2.bitwise_and(bird_map_rgb, bird_map_rgb, mask = bird_mask)
bird_map = bird_map[1:227, 1:227]
```

15

Ocean ICT: > LOWG > 3. 코드 설계

16 최적의 풍력발전소 위치 계산

```
fin_arr = map_arr + wind_arr + bird_arr + dist_arr + st_arr
gen_list = list()
for a in range(269):
    for b in range(227):
        if fin_arr[a, b] == np.max(fin_arr):
            gen_list.append([a, b])
map_rgb_marked = map_rgb[1:227, 1:227]
```

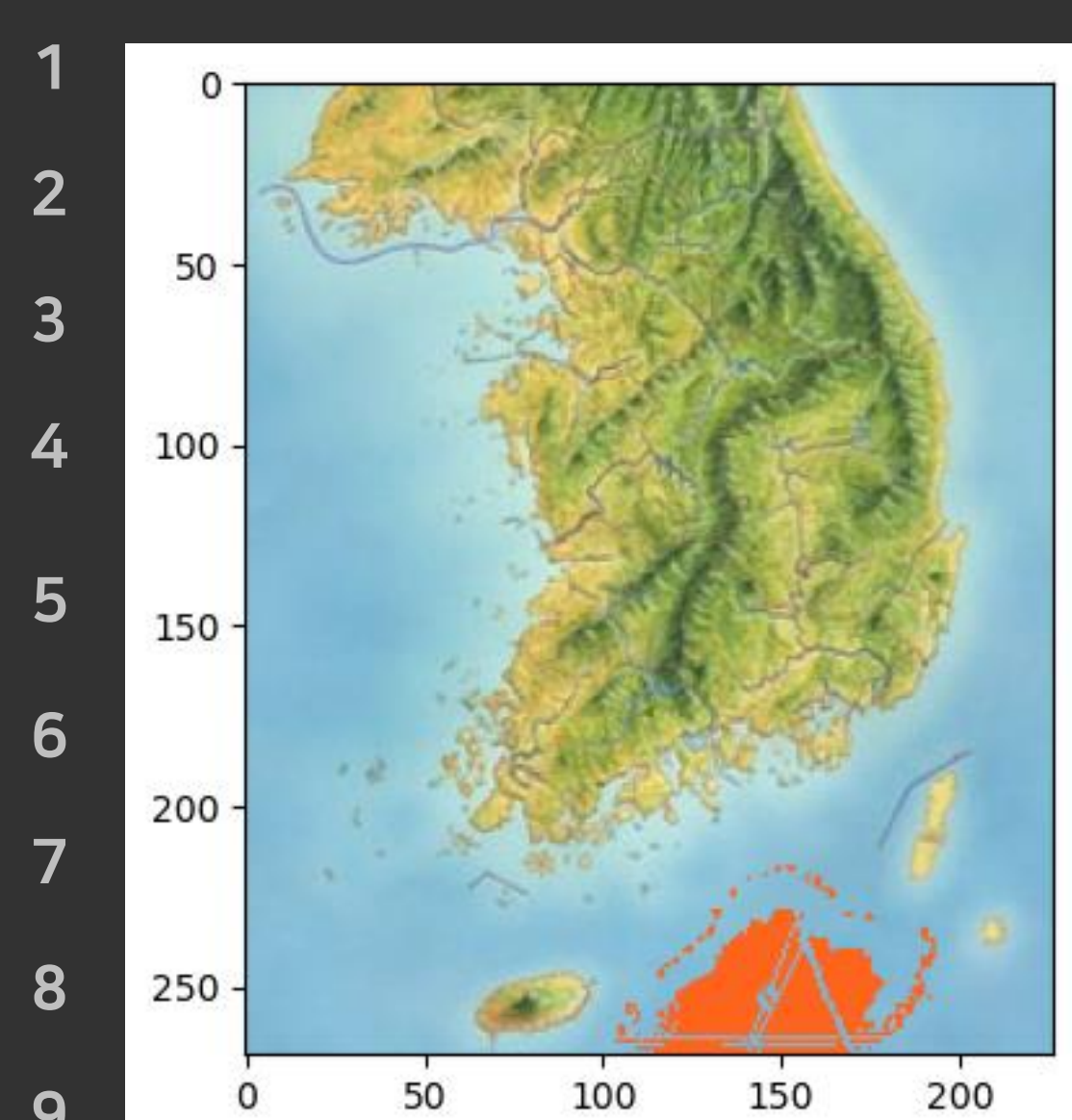
21 태풍 이동경로 -가중치 부여

```
stap_hgr = cv2.imread('stap_hgr.jpg', cv2.IMREAD_COLOR)
stap_rgb = cv2.cvtColor(stap_hgr, cv2.COLOR_BGR2RGB)
stap_rgb_low = cv2.inRange(stap_hgr, dist_low, dist_high)
stap_rgb_high = cv2.inRange(stap_hgr, dist_low, dist_high)
stap_mask = cv2.inRange(stap_rgb, stap_rgb_low, stap_rgb_high)
stap_mask = cv2.bitwise_and(stap_rgb, stap_rgb, mask = stap_mask)
st_arr = np.zeros((200, 227))
for l in range(227):
    for p in range(269):
        if st_arr[p, l] <= 0:
            st_arr[p, l] = -100
        else:
            pass
```

26 지면으로부터 거리 +가중치 부여

```
dist_arr = np.zeros((200, 227))
for q in range(227):
    for v in range(227):
        dist_sq = (q-100)**2 + (v-130)**2
        if dist_sq <= 8000:
            dist_arr[q, v] = 50
        elif dist_sq <= 16000:
            dist_arr[q, v] = 40
        elif dist_sq <= 24000:
            dist_arr[q, v] = 30
        elif dist_sq <= 32000:
            dist_arr[q, v] = 20
        elif dist_sq <= 40000:
            dist_sq = 10
        else:
            dist_sq = 0
```

Ocean ICT: > LOWG > 4. 실험 결과



- 10 각 요소들에 +, - 가중치를 부여
- 11 가중치 합 계산 및 지도에 마스킹
- 12 남해안 부근이 가장 가중치 값이 컸음
- 13 일부 태풍 및 철새의 경로가 제외됨

Ocean ICT: > LOWG > 5. 느낀점

- 1 1103 김석현
- 2 본 탐구를 통해 파이썬 모듈들에 대해 알게되어 좋았음
- 3 그리고 이를 이미지 데이터 처리, 연산에 적용시켜
- 4 원하는 결과물을 제작할 수 있게되어 보람있음
- 5 1114 이동윤
- 6 다음에는 해양 생태계와 같은 더 많은 변수를 고려해
- 7 진정한 친환경 에너지를 실천할 수 있는 풍력발전소를
- 8 설계해보고 싶다는 생각이 들

