

대학 대항전: 퍼즐 이미지 AI 경진대회

알고리즘 | 월간 데이콘 | 대학 대항전 | 비전 | 자기지도학습 | 기술 혁신 | 분류

IT융합대학
인공지능공학과

백찬형

2024년 02월 06일



Contents

● **I. Introduction**

● **II. EDA**

● **III. Strategies**

● **IV. Conclusion**

●

●

I. Introduction

1. 대회 목표

- 4x4의 격자 형태의 16개의 조각으로 구성된 순서가 뒤바뀐 퍼즐 이미지를 대상으로 원래 순서를 맞추는 것

2. 평가 산식

- 퍼즐 재구성 정확도 : $(1 \times 1 \text{ 퍼즐 재구성 정확도} + 2 \times 2 \text{ 퍼즐 재구성 정확도} + 3 \times 3 \text{ 퍼즐 재구성 정확도} + 4 \times 4 \text{ 퍼즐 재구성 정확도}) / 4$
- $0.3 * (\text{Seen 이미지 퍼즐 재구성 정확도}) + 0.7 * (\text{Unseen 이미지 퍼즐 재구성 정확도})$



| | | | |
|---------|---------|---------|---------|
| 1 9 | 2 12 | 3 11 | 4 14 |
| 5 6 | 6 16 | 7 10 | 8 13 |
| 9 15 | 10 8 | 11 3 | 12 1 |
| 13 4 | 14 5 | 15 2 | 16 7 |

II. EDA

1. 정렬된 이미지

- 실사 이미지와 생성 이미지로 구성되어 있으며 생성 이미지가 더 많은 경향을 보임.
- 단색이 대부분을 차지하는 이미지나 복잡한 패턴을 보이는 데이터도 있음.



실사 이미지



생성 이미지

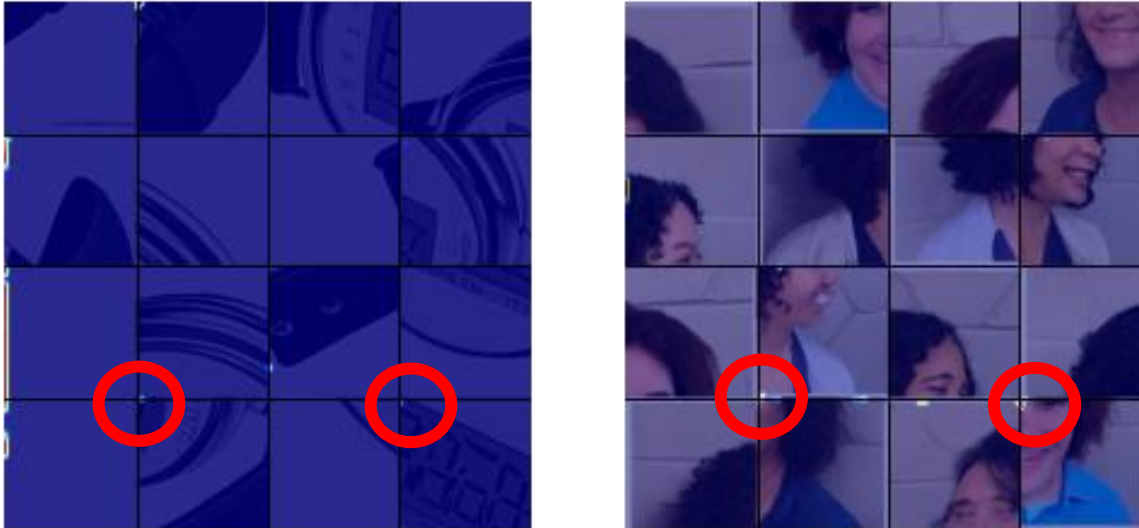


대부분 단색인 이미지

II. EDA

2. gradCAM

- class 예측 시 특정 부분(Edge, Vertex)을 집중적으로 보는 경향이 있음.
- 사람처럼 edge 부분을 보면서 판단하는 경향이 있음.
- Vertex 부분에 속단하는 경향이 있음.



gradCAM 시각화

III. Strategies

1. Model Selection

- **Model: SegFormer**
- (batch, 16, 4, 4)를 output으로 출력하도록 실험했을 때, 퍼즐 재구성 정확도가 높은 모델을 선정함.

| Model | Public Score |
|-----------------|--------------|
| U-Net(baseline) | 0.08603 |
| ResNet50 | 0.11836 |
| DeepLabV3 | 0.20338 |
| DTrC | 0.41533 |
| SegFormer | 0.85638 |

III. Strategies

1. Model Selection

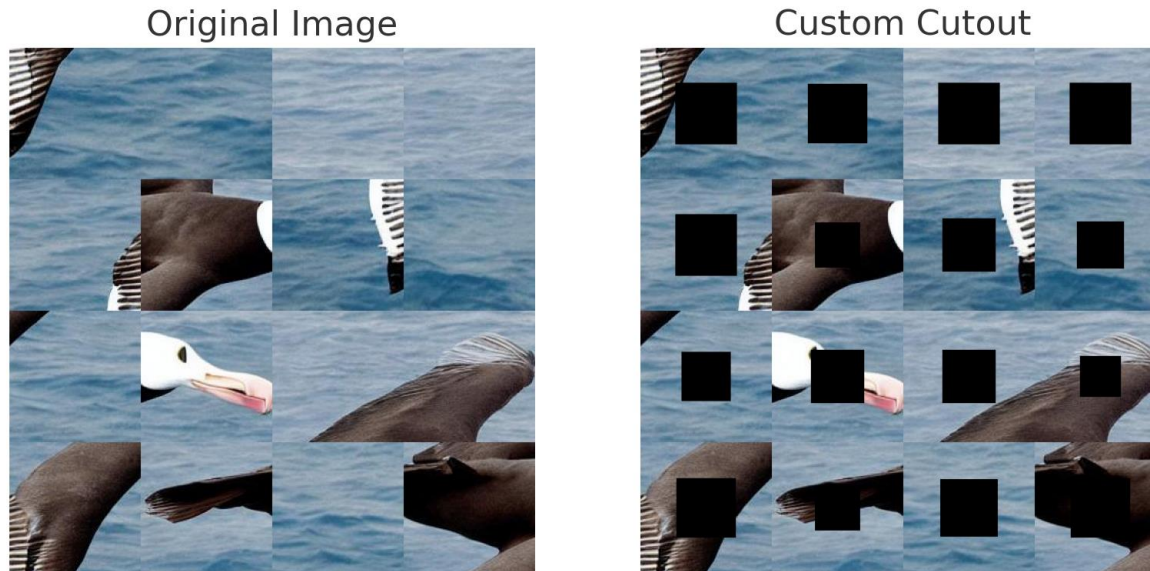
- Output shape를 맞추기 위해 Down Sampling 실험함.
- Baseline과 같이 Conv를 통해 shape를 맞춘다면 성능이 떨어지는 경향을 보임.
- AdaptiveAvgPool을 사용함으로써 Conv를 사용하는 경우보다 모델의 복잡도를 줄이도록 유도함.

| Model | Public Score |
|-------------------------------|--------------|
| SegFormer + Conv | 0.42717 |
| SegFormer + AdaptiveAvgPool2d | 0.85638 |

III. Strategies

2. Augmentation

- 퍼즐의 Edge 근처에서 feature를 얻도록 Custom Cutout을 설계함.
- 인간이 퍼즐을 푸는 것처럼 Edge를 우선적으로 보고 판단하도록 유도함.

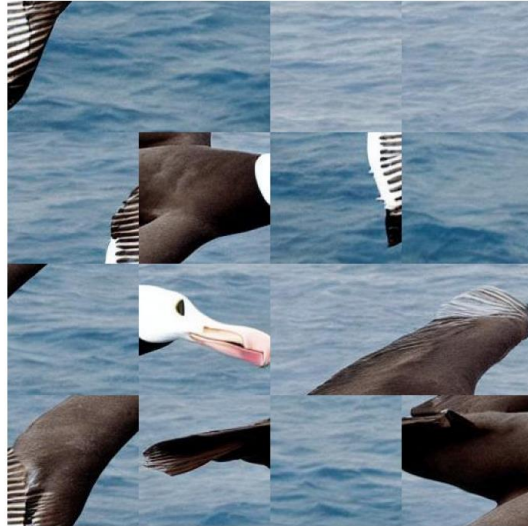


III. Strategies

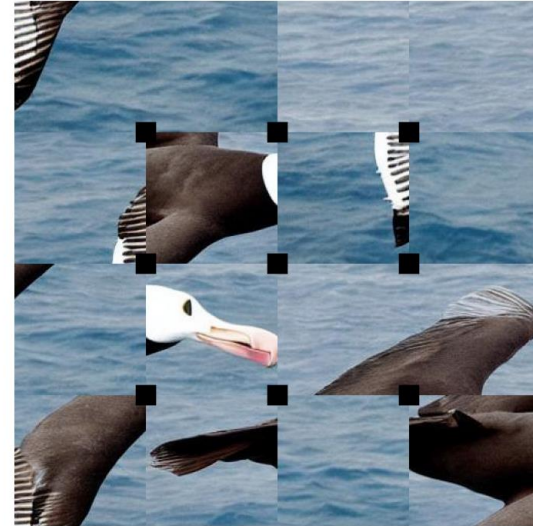
2. Augmentation

- 퍼즐의 Vertex 부분에 속단하는 경향을 방지하기 위해 Custom Cutout을 설계함.

Original Image



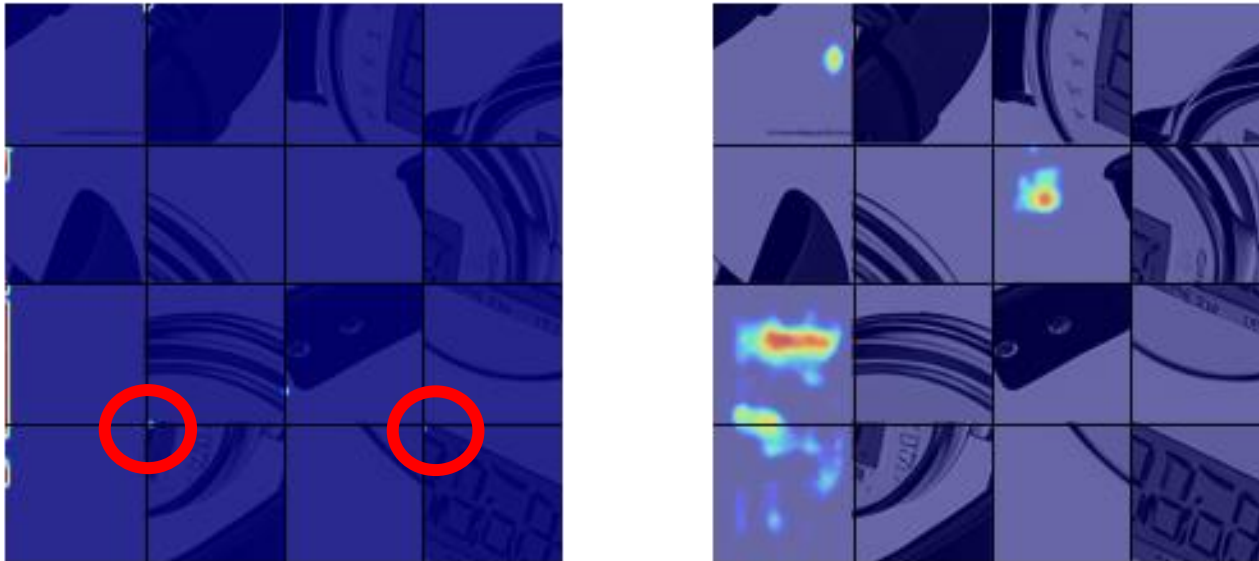
Vertex Cutout



III. Strategies

2. Augmentation

- 퍼즐의 Edge 근처에서 다양한 부위를 보고 클래스를 예측함
- 퍼즐의 Vertex 부분을 덜 보며 클래스를 예측함



Custom Cutout 적용 전/후

III. Strategies

2. Augmentation

- 단색이 많은 이미지에 대해 쉽게 풀 수 있도록 밝기, 채도 등의 변화 시도
- 최대한 비슷한 퍼즐이 발생하지 않도록 유도함.

Original Image



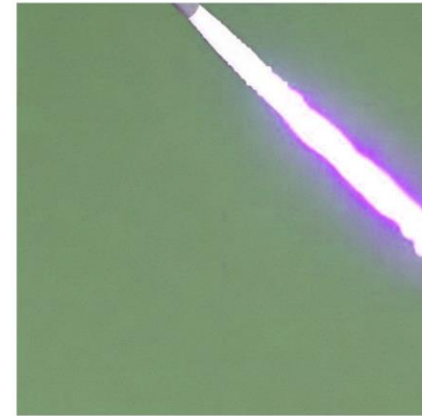
RandomBrightnessContrast



HueSaturationValue



ChannelShuffle



III. Strategies

3. 중복 제거

- Inference 과정에서 중복이 발생함.
- 이를 방지하기 위해 한번 선택된 logit은 소거하는 방식으로 추론

2x2 퍼즐에서 중복 제거 전략 예시

| | 1 | 2 | 3 | 4 |
|---|-----|-----|-----|------|
| 1 | 0.1 | 0.7 | 0.1 | 0.8 |
| 2 | 0.6 | 0.1 | 0.1 | 0.1 |
| 3 | 0.2 | 0.1 | 0.2 | 0.05 |
| 4 | 0.1 | 0.1 | 0.6 | 0.05 |

기존 argmax 방식: [2, 1, 4, 1]

| | 1 | 2 | 3 | 4 |
|---|-----|-----|-----|------|
| 1 | 0.1 | 0.7 | 0.1 | 0.8 |
| 2 | 0.6 | 0.1 | 0.1 | 0.1 |
| 3 | 0.2 | 0.1 | 0.2 | 0.05 |
| 4 | 0.1 | 0.1 | 0.6 | 0.05 |

새로운 argmax 방식

- 1열에서 label을 정하고 해당 행을 소거: [2,] (blue)
 - 2열에서 label을 정하고 해당 행을 소거: [2, 1] (red)
 - 3열에서 label을 정하고 해당 행을 소거: [2, 1, 4] (green)
 - 4열에서 label을 정하고 해당 행을 소거: [2, 1, 4, 3]
- (자세한 방법은 코드 참고)

III. Strategies

4. Ensemble

- Train data를 random split한 10가지 버전으로 학습한 10개 모델 중 성능이 가장 잘 나온 7개를 Ensemble한다.
- Hard Vote보다 Soft Vote가 성능이 나오는 경향을 보임.

| Model | Public Score |
|-------------------------------|--------------|
| Segformer (Best Model) | 0.99617 |
| Segformer * 7 | 0.99762 |
| Segformer * 10 | 0.99727 |
| SegFormer * 7 + custom argmax | 0.99764 |

IV. Conclusion

1. 마무리

- Down Sampling 방법에 따라 성능 차이가 발생함.
- Cutout, 색조, 채도 등을 적용하여 어려운 퍼즐 이미지를 풀 수 있도록 유도함.
- 성능이 가장 잘 나오는 7개 모델을 Ensemble하고 중복 제거한 결과물이 가장 높은 score를 기록함.

Thank you for listening

Q&A