

대학 대항전 : 퍼즐 이미지 AI 경진대회

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

CONTENTS

01

Competition & Data

- Competition info

02

Model Selection

- Choosing Model

03

Strategies

- Data Augmentation
- Ensemble

퍼즐 이미지 재구성 AI 모델 개발

예를 들어, 회색 숫자는 본 대회에서 지정한 퍼즐 조각의 위치이고 파란색 숫자는 해당 퍼즐의 조각이 원래 있어야 할 위치입니다.

즉 예시와 같이 [9, 12, 11, 14, 6 ... 1, 4, 5, 2, 7]의 각 위치에 해당되는 이미지 조각의 원래 순서가 제공됩니다. 즉, 뒤섞인 퍼즐 조각의 1부터 16까지의 위치에 해당되는 이미지 조각이 몇 번째 순서인지 맞추는 AI 모델을 만드는 것이 이번 대회의 과제입니다.

퍼즐 이미지를 재구성하는 AI 모델을 개발해야 합니다. 구체적으로는 4x4의 격자 형태의 16개의 조각으로 구성된 순서가 뒤바뀐 퍼즐 이미지를 대상으로 원래 순서를 맞추는 것입니다.



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

Model Selection

모델은 [파이썬초보만](#)님의 모델을 변형하여 사용하였습니다.
코드 공유해주신 [파이썬초보만](#)님 감사드립니다.

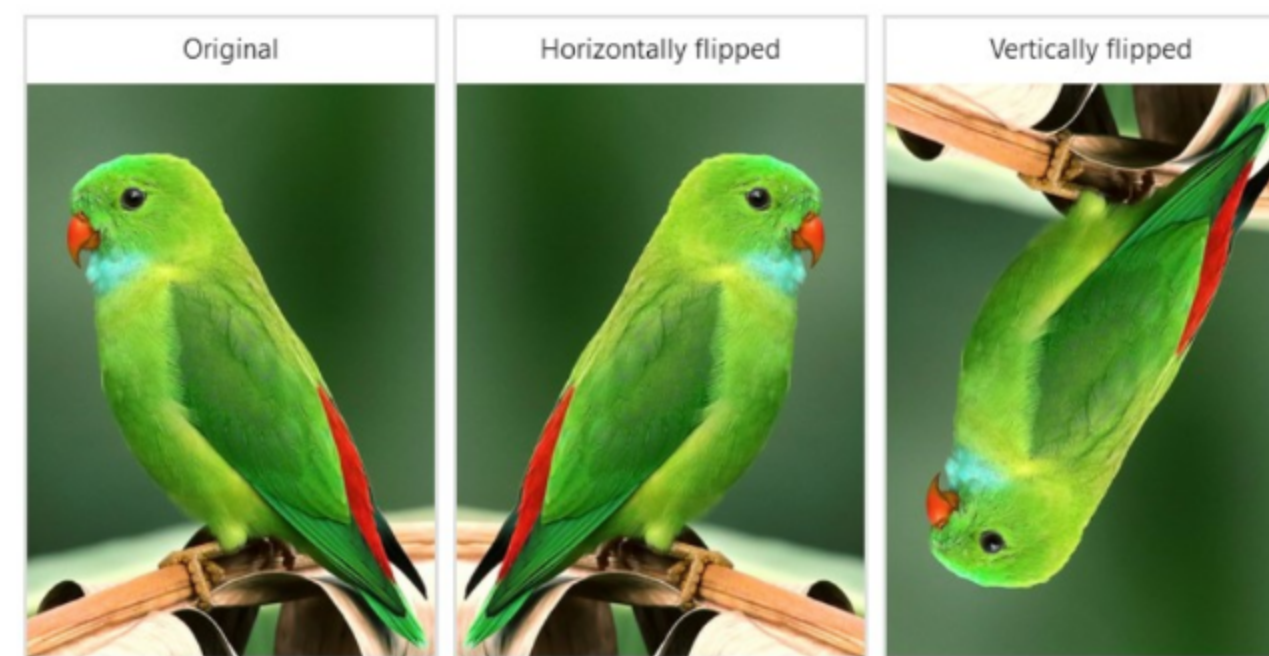
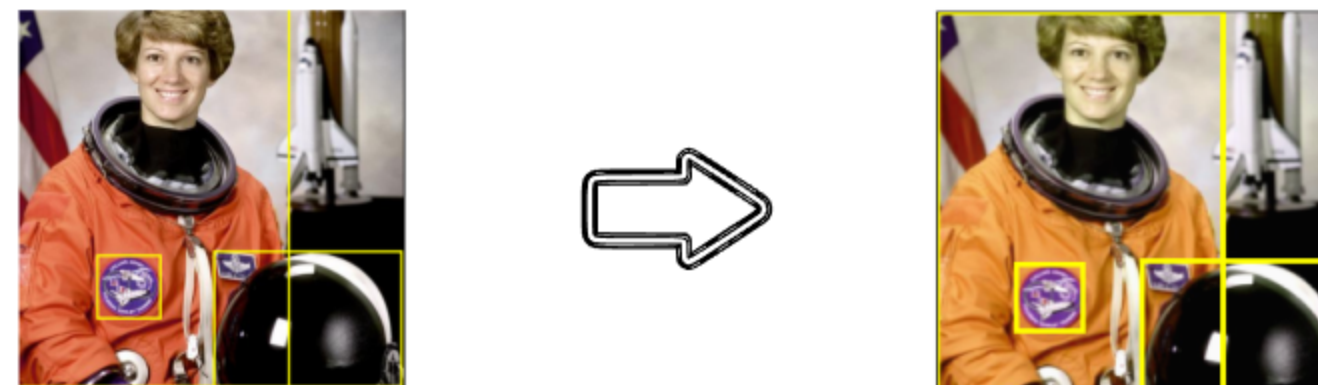
- 각 패치를 중심으로 인접한 모양(┐,└,┘,┌,+,┑, 등)을 사용하여 인접한 패치들의 연결관계를 사용하여 학습에 도움을 주었습니다.
- 16 X 16 의 패치들로 나누어 최대한 퍼즐과 비슷한 형태에서 특징을 추출하려 했습니다.

기존 Optimizer 변경

Adam -> AdamW + CosineAnnealingLR (+0.006)

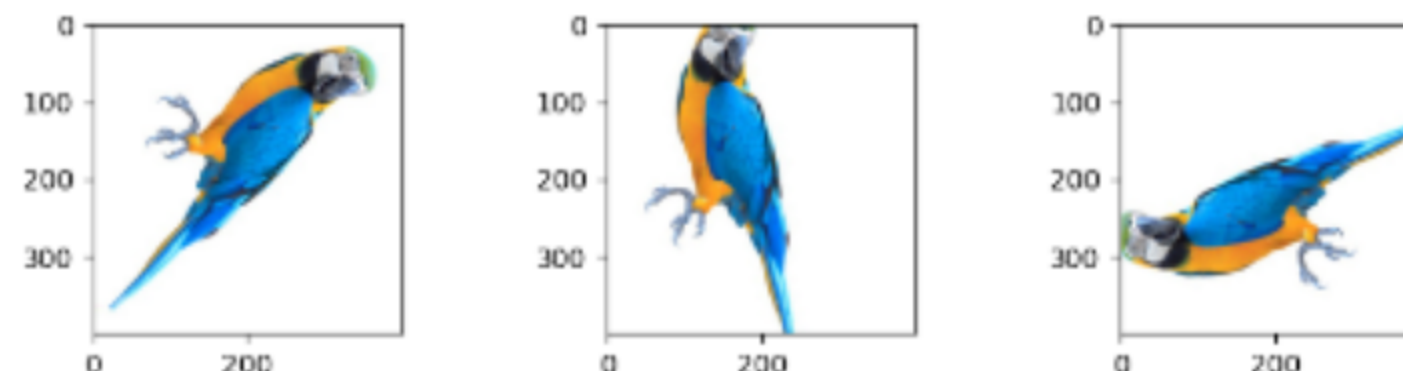
Strategies

- Random Photometric Distort
- Random Erasing (+0.001)
- Random Vertical Flip (+0.002)
- Random Horizontal Flip (+0.002)

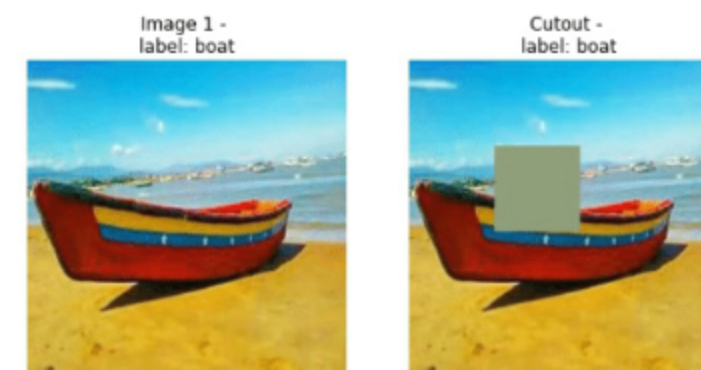


Strategies

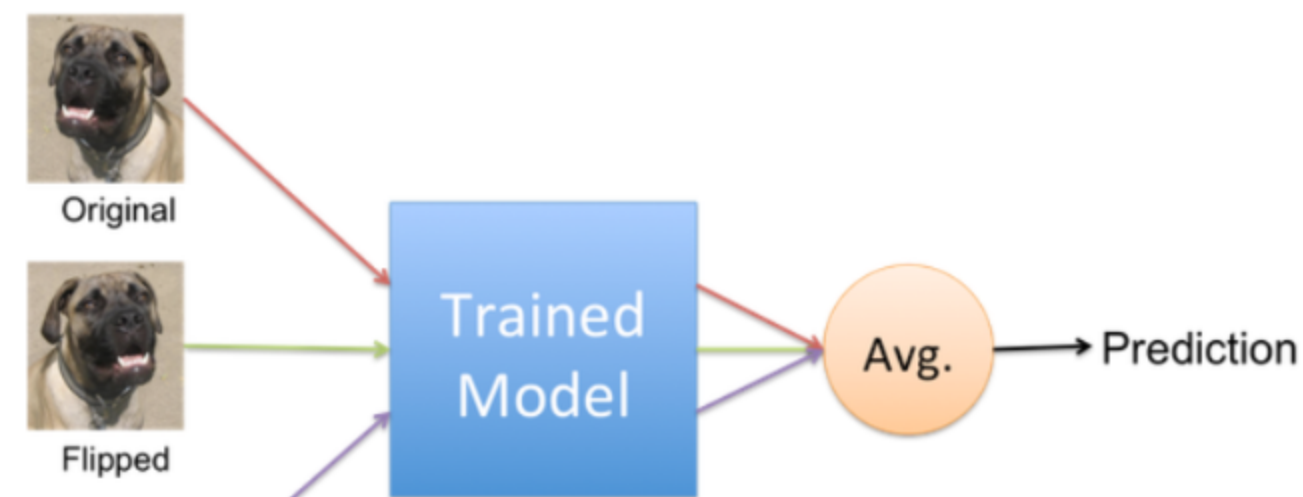
- Random Rotate 90, 180, 270 (+0.004)



- Random Cutout (Before Shuffle) (-0.003)



- Test Time Augment (Flip) (-0.006)



Strategies

```
pred1 = pd.read_csv('./CSVs/0.977.csv').to_numpy() ## Basemodel(Jigsaw_Electra) ALL 19 Epoch
pred2 = pd.read_csv('./CSVs/0.975.csv').to_numpy() ## AdamW + CosineALR Train/Valid 80/20
pred3 = pd.read_csv('./CSVs/0.983.csv').to_numpy() ## AdamW + CosineALR ALL 43 Epoch
pred4 = pd.read_csv('./CSVs/0.981.csv').to_numpy() ## RHF 0.3, RVF 0.3, AdamW + CosineALR Train/Valid 80/20
pred5 = pd.read_csv('./CSVs/0.985.csv').to_numpy() ## RHF 0.3, RVF 0.3, AdamW + CosineALR ALL 59 Epoch
pred6 = pd.read_csv('./CSVs/0.988.csv').to_numpy() ## RHF 0.3, RVF 0.3, ROTATE 0.3, AdamW + CosineALR ALL 91 Epoch
pred7 = pd.read_csv('./CSVs/0.982.csv').to_numpy() ## RVF_TTA model 0.985
pred8 = pd.read_csv('./CSVs/0.987.csv').to_numpy() ## RVF_TTA_ver2 model 0.988
pred9 = pd.read_csv('./CSVs/0.988_2.csv').to_numpy() ## RHF_TTA
pred10 = pd.read_csv('./CSVs/0.987_2.csv').to_numpy() ## RHF_RVF_TTA
pred11 = pd.read_csv('./CSVs/0.985_2.csv').to_numpy() ## RHF 0.3, RVF 0.3, ROTATE 0.3, CUTOUT 0.5, AdamW + CosineALR ALL 61 Epoch
pred12 = pd.read_csv('./CSVs/0.986.csv').to_numpy() ## RHF 0.3, RVF 0.3, ROTATE 0.3, ColorJitter, AdamW + CosineALR ALL 65 Epoch
pred13 = pd.read_csv('./CSVs/0.983_2.csv').to_numpy() ## RHF 0.3, RVF 0.3, ROTATE 0.3, AdamW + CosineALR (1e-7)
pred14 = pd.read_csv('./CSVs/0.988_3.csv').to_numpy() ## RHF 0.5, RVF 0.5, ROTATE 0.5, Erase 0.5, AdamW + CosineALR ALL 90 Epoch
```

앞에서 다룬 Augmentation들을 적용한 Model로 Inference를 진행한후 CSV들을 Ensemble하여 Robust하게 만듦 -> 성능은 떨어지는 CSV도 Ensemble에 도움이 됨