

# 스마트축사 데이터 활용 대회

이종원

## 대회 주제 및 목표

한우의 발정 시기를 파악하여 발정 시기를 놓치지 않고 수정에 성공하도록 스마트축사 이미지 데이터 내에 발정행동중인 한우를 판별하는 세그멘테이션 모델을 개발

## 평가지표 및 유의사항

평가지표는 일반 소 1: 발정 소 3으로 스코어 가중합을 하는 Weighted mAP로 진행

본 대회의 모델은 세그멘테이션 모델로 개발되어야 하나 평가는 세그멘테이션 결과를 Bounding Box로 변환하여 Weighted mAP로 평가

## 데이터 셋 설명



본 대회의 데이터는 위와 같이 축사 내에서 촬영된 소들의 이미지로 이루어진 입력 데이터와 해당 이미지의 메타정보 및 소들의 segmentation 정보, 클래스 분류 정보로 이루어진 json 형식의 출력 데이터로 구성

입력데이터의 경우 1282x720 해상도의 이미지와 1280x600 해상도의 이미지가 혼재되어 있음

클래스는 비 발정행동중인 소: 1, 발정행동중인 소:2로 구분

정답(어노테이션) 파일은 Coco Dataset의 기본 구성에서 본 과제를 진행하는 데에 불필요한 요소를 정리하여 축약한 구조로 이루어져 있음

## 제공되는 데이터 구조

- data.zip
  - train\_images: 학습데이터 이미지 폴더. train\_0001부터 train\_4326까지의 이미지 가운데 일부 오류 데이터를 제외한 총 4,284개 이미지로 구성
  - test\_images: 평가데이터 이미지 폴더. test\_0001부터 test\_1443까지의 이미지 가운데 일부 오류 데이터를 제외한 총 1,431개 이미지로 구성
  - train\_answer.json: 학습데이터 정답(어노테이션) 파일.
  - test\_answer\_sample.json: 평가데이터 정답 양식 파일. Confidence score를 나타내는 "conf" 항목이 추가로 존재함에 유의
  - evaluation.py: 결과 평가에 사용되는 스코어 계산 스크립트

## 제출 양식 (test\_answer\_sample.json)

```
{"images":  
  [  
    {  
      "file_name": "test_0001.PNG",  
      "width": 1282,  
      "height": 720,  
      "id": 1},  
    {  
      "file_name": "test_0002.PNG",  
      "width": 1282,  
      "height": 720,  
      "id": 2},  
    ....  
  ],  
  "annotations":  
    [  
      {  
        "image_id": 1,  
        "segmentation": [000, 000, ...],  
        "category_id": 0,  
        "conf": 0.0},  
      {  
        "image_id": 2,  
        "segmentation": [111, 111, ...],  
        "category_id": 0,  
        "conf": 0.0},  
    ]  
  }
```

```
....  
]  
}
```

크게 "images"와 "annotations"의 두 list로 이루어진 dictionary로 Coco dataset label의 형식을 따름  
"images"파트의 메타정보를 평가시에 활용하기 때문에 제출할 파일에서 누락하지 않고 원래 양식을 준수해야 함  
train data의 annotation 파일과 다르게 confidence score를 나타내는 "conf"항목이 존재하며 비우면 안 됨에 유의

## 코드

\*\*\*\*\*소\_발정기\_판별\_with\_detectron2.ipynb 따로 첨부\*\*\*\*\*

주요 부분만 설명해보자면,  
주어진 학습 데이터 정답(어노테이션) 파일과 학습 모델(detectron2)에 입력해야 하는 데이터의 형식이 달라 이를  
전처리 과정을 통해 알맞게 변환시켜 파일로 저장하는 코드가 12번째 셀이다.  
반대로 출력도 모델에서 출력되는 형식은 binary\_mask 형태이고 원하는 형식은 segmentation의 polygon스타일  
의 형태 테두리 좌표였기 때문에 이를 바꿔주는 함수를 따로 만들어 바꾸어 주는데 이 함수가 24번째 셀에 있다.  
24번째 셀에서 만든 함수를 이용하여 모델에서 예측한 값을 제출 양식에 필요한 값만 뽑아서 형식에 맞게 .json  
파일로 만들어 저장하는 코드가 31번째 셀이다.

## 입력 데이터

총 4284장의 이미지



오른쪽 사진을 보면 빨강게 마킹한 소들을 보면 발정 행동을 하는 것을 볼 수 있다. 이와 같은 소들을 판별해 내는 것이다.



## 학습 데이터 정답(어노테이션) 파일

```
1 {
2   "images": [
3     {
4       "file_name": "train_0001.PNG",
5       "width": 1282,
6       "height": 720,
7       "id": 1
8     },
9     {
10      "file_name": "train_0002.PNG",
11      "width": 1282,
12      "height": 720,
13      "id": 2
14    },
15    {
16      "file_name": "train_0003.PNG",
17      "width": 1282,
18      "height": 720,
19      "id": 3
20    },
21    {
22      "file_name": "train_0004.PNG",
23      "width": 1282,
24      "height": 720,
25      "id": 4
26    }
27  ],
```

```
25700  },
25701  {
25702    "file_name": "train_4326.PNG",
25703    "width": 1282,
25704    "height": 720,
25705    "id": 4326
25706  }
25707 ],
25708 "annotations": [
25709   {
25710     "image_id": 1,
25711     "segmentation": [
25712       292.11,
25713       510.56,
25714       291.81,
25715       534.68,
25716       296.81,
25717       553.44,
25718       302.97,
25719       574.33,
25720       299.74,
25721       584.31,
25722       305.62,
25723       581.96,
25724       322.06,
25725       589.01,
25726       339.69,
25727       628.07,
25728       334.69,
25729       640.09,
25730       343.5,
25731       660.68,
25732       360.83,
25733       683.0,
25734       352.61,
25735       705.91,
25736       359.95,
25737       719.71,
25738       404.3,
25739       718.54,
25740       411.64,
25741       683.88
```

```
25990  855.02,
25987  308.17
25988  ],
25989  "category_id": 1
25990  },
25991  {
25992    "image_id": 1,
25993    "segmentation": [
25994      15.49,
25995      171.98,
25996      11.38,
```

입력 이미지 데이터들의 각각의 소들의 segmentation들과 이미지의 크기, 이미지 파일명, 소의 발정/비발정 상태 등의 정보가 담겨있는 정답 파일로 학습에 사용한다.

## 예측 결과







위의 사진들에서 각 segmentations된 소들이 labeling이 되어있는 것을 확인할 수 있는데, cow1으로 labeling된 segmentation들이 발정 상태의 소이고, cow라고 labeling된 segmentation들이 비발정 상태의 소이다.

## 결과 도출

MAX\_ITER와 THRESH, BATCH\_SIZE, BASE\_LR 등을 설정해주면서 제일 높은 점수가 나오도록 여러 번 시도해 보았다.

결과적으로 10000번의 학습과  $BASE\_LR = 0.00025$ , THRESH값 0.8을 넣었을 때 0.89515점으로 가장 높은 점수를 기록하고 마무리하였다.



## 공개순위

회차	순위	닉네임	제출횟수	점수	제출일
1/1	1	anerimi	42회	0.98627점	22.01.22 14:06:00
1/1	2	하찌	29회	0.97434점	22.01.20 11:35:12
1/1	3	PIG_DUCK	62회	0.9739점	22.01.24 09:32:32
1/1	4	KF94	43회	0.97372점	22.01.25 16:55:22
1/1	5	스티브킴	61회	0.97162점	22.01.20 17:42:00
1/1	6	소고기 좋아하는 군이	18회	0.96984점	22.01.24 15:09:19
1/1	7	스마트카우	32회	0.95444점	22.01.25 13:46:03
1/1	8	January	8회	0.95037점	22.01.19 13:39:28
1/1	9	공공재	2회	0.90679점	22.01.25 16:56:14
1/1	10	이금강	12회	0.89952점	22.01.13 02:01:45
1/1	11	하루여섯끼	19회	0.89515점	22.01.22 22:42:30
1/1	12	조맹팔이	11회	0.84991점	22.01.24 23:31:09
1/1	13	테쌍가즈아	64회	0.81083점	22.01.24 22:59:53

위와 같이 285명이 참가한 이 대회에서 11등을 기록했다.

## 느낀점

이전까지는 딥러닝 모델을 활용해본 경험이 없었고 이 대회가 딥러닝을 처음 접한 대회라 많이 미흡했다. 각종 지식들을 하나하나 다 공부했고, 완전 생소하다 보니 이것을 위해선 무엇을 공부하는지조차 몰라 이것저것 한참을 공부했다. 0.89515점에서 더 점수가 오르지 못했는데 다른 팀들의 0.9 후반대의 점수를 보면 궁금함이 들기도 했다. 어떻게 해야 저런 점수가 나올까 모델을 불러오는게 아니라 직접 설계했어야 했나라는 생각도 들었다. 그래서 모델을 직접 만들지 못하고 결국 이미 만들어진 모델을 불러와 세부적인 수치만 변경해보는 걸로 했던게 아쉬웠다. 만약 모델 구조도 mask r-cnn 기반으로 이 대회의 목적에 맞게 소 사진만을 위한 모델을 만들었다면 더욱 성능이 올라갔을 수도 있지 않았을까?.. 하지만 이 대회가 진행되는 18일이라는 기간안에 그것까지 진행하기엔 쉽지 않았다. 공부할수록 어려웠다. 또한, 유튜브나 구글 등 인터넷에 흔히 나온 지식들로 한계가 있었다.