데이터 증강 라벨링 이미지 처리 #01

전진희

실습01. 이미지 폴더에 있는 파일 가져오기(os.listdir)

<기본>

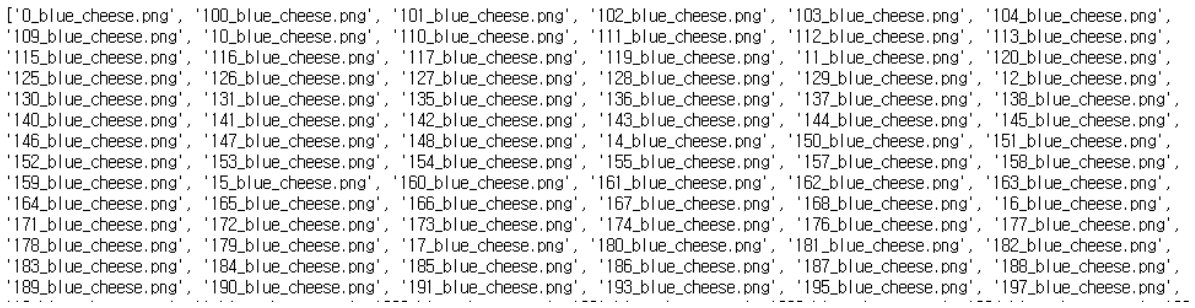
```
import os

# 이미지가 저장되어 있는 폴더 경로
img_dir = "./blue_cheese_img_download/"

# 디렉토리 내 모든 파일 목록 가져오기
file_list = os.listdir(img_dir)
print(file_list)

# 정렬이 되지 않는다는 단점이 있다
```

<이미지 정렬 시도>

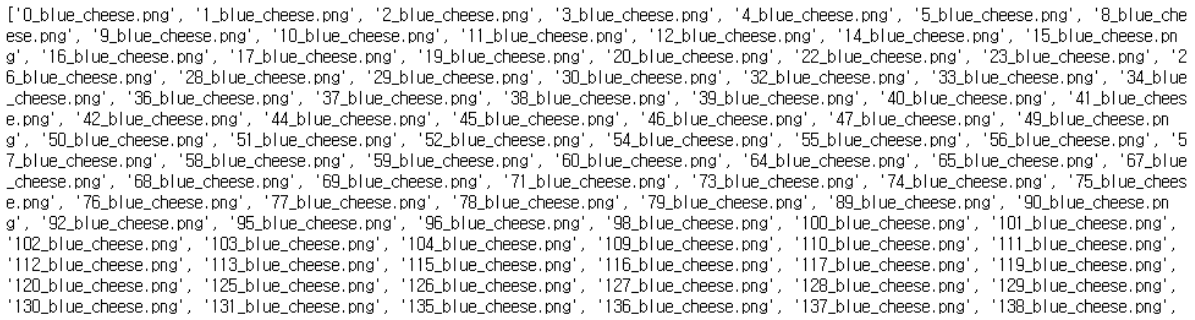
```
import os

img_dir = "./blue_cheese_img_download/"

# 디렉토리 내 모든 파일 목록 가져오기
file_list_temp = sorted(os.listdir(img_dir))
print(file_list_temp)

# 텍스트와 숫자가 합쳐져서 숫자정렬이 이상하게 되었다
```

<natsort 라이브러리를 이용한 이미지 정렬>

```
!pip install natsort
# 텍스트로 된 숫자 정렬에 쓰이는 라이브러리 설치

from natsort import natsort
img_dir = "./blue_cheese_img_download/"
file_list_temp01 = natsort.natsorted(os.listdir(img_dir))
print(file_list_temp01)
```



Natsort 쓰기 전



Natsort 사용 후

<Glob 사용>

```
import glob
import os

"""
폴더구조
train
   image
      apple
            - aaa.png
      bon
            - bbb.png

에서 image 아래 폴더 모든 파일 가져오고 싶다면
아래와 같이 작성

(확장자가 여러개 일 경우 if 문 써서 파일 가져오게 하면된다 'if ext_list...')
"./train/image/*/*.png"

file_list_temp
['./train/image/apple/aaa.png' ,'./train/image/bon/bbb.png']



"""
image_ path = "./blue_cheese_img_download/*.png"
file_list_temp = glob.glob(image_path)

# file_list_temp = glob.glob(os.path.join("./blue_cheese_img_download/", "*.png")) 위에 두줄을 이 한줄로 대체 가능
print(file_list_temp)
```

```
['./blue_cheese_img_download₩₩0_blue_cheese.png', './blue_cheese_img_download₩₩100_blue_cheese.png', './blue_cheese_img_download₩₩101_blue_cheese.png', './blue_cheese_img_download₩₩102_blue_cheese.png', './blue_cheese_img_download₩₩103_blue_cheese.png', './blue_cheese_img_download₩₩104_blue_cheese.png', './blue_cheese_img_download₩₩109_blue_cheese.png', './blue_cheese_img_download₩₩10_blue_cheese.png', './blue_cheese_img_download₩₩110_blue_cheese.png', './blue_cheese_img_download₩₩111_blue_cheese.png', './blue_cheese_img_download₩₩112_blue_cheese.png', './blue_cheese_img_download₩₩113_blue_cheese.png', './blue_cheese_img_download₩₩115_blue_cheese.png', './blue_cheese_img_download₩₩116_blue_cheese.png', './blue_cheese_img_download₩₩117_blue_cheese.png', './blue_cheese_img_download₩₩119_blue_cheese.png', './blue_cheese_img_download₩₩11_blue_cheese.png', './blue_cheese_img_download₩₩120_blue_cheese.png', './blue_cheese_img_download₩₩125_blue_cheese.png', './blue_cheese_img_download₩₩126_blue_cheese.png', './blue_cheese_img_download₩₩127_blue_cheese.png', './blue_cheese_img_download₩₩128_blue_cheese.png', './blue_cheese_img_download₩₩129_blue_cheese.png', './blue_cheese_img_download₩₩12_blue_cheese.png', './blue_cheese_img_download₩₩130_blue_cheese.png', './blue_cheese_img_download₩₩131_blue_cheese.png', './blue_cheese_img_download₩₩135_blue_cheese.png', './
```

os.walk

```
import os

def get_img_path(root_path):
    file_paths = []  # 파일 경로를 저장할 리스트

    # root_path를 시작으로 디렉토리 트리를 탐색
    for (path, dir, files) in os.walk(root_path):
        print("path:", path)  # 현재 디렉토리 경로 출력
        print("dir:", dir)  # 현재 디렉토리에 있는 하위 디렉토리들 출력
        print("files:", files)  # 현재 디렉토리에 있는 파일들 출력

        # 현재 디렉토리에 있는 각 파일에 대해 반복
        for file in files:
            ext = os.path.splitext(file)[-1].lower()  # 파일 확장자를 소문자로
            print("ext:", ext)



            formats_list = [".bmp", ".jpg", ".png", ".tif", ".gif", ".dng", ".tiff"]

            if ext in formats_list:
                file_path = os.path.join(path, file)  # 전체 파일 경로
                print("file_path:", file_path)  # 파일 경로 출력
                file_paths.append(file_path)  # 파일 경로를 리스트에 추가

    return file_paths


file_paths_temp = get_img_path("./blue_cheese_img_download/")
print(file_paths_temp)
```

```
g', '56_blue_cheese.png', '57_blue_cheese.png', '58_blue_cheese.png', '59_blue_cheese.png', '5_blue_cheese.png', '60_blue_cheese.png',
'64_blue_cheese.png', '65_blue_cheese.png', '67_blue_cheese.png', '68_blue_cheese.png', '69_blue_cheese.png', '71_blue_cheese.png', '73
_blue_cheese.png', '74_blue_cheese.png', '75_blue_cheese.png', '76_blue_cheese.png', '77_blue_cheese.png', '78_blue_cheese.png', '79_bl
ue_cheese.png', '89_blue_cheese.png', '8_blue_cheese.png', '90_blue_cheese.png', '92_blue_cheese.png', '95_blue_cheese.png', '96_blue_c
heese.png', '98_blue_cheese.png', '9_blue_cheese.png']
ext: .png
file_path: ./blue_cheese_img_download/0_blue_cheese.png
ext: .png
file_path: ./blue_cheese_img_download/100_blue_cheese.png
ext: .png
file_path: ./blue_cheese_img_download/101_blue_cheese.png
ext: .png
```

실습02. 이미지 정제 : 정사각형 데이터 만들기

딥러닝이나 영상 처리를 하다 보면 입력 데이터를 정사각형으로 만들어줘야 한다.

그 때 정사각형이 아닌 데이터는 정사각형이 아닌 데이터 위 아래에 여유 공간을 더 추가해준다.

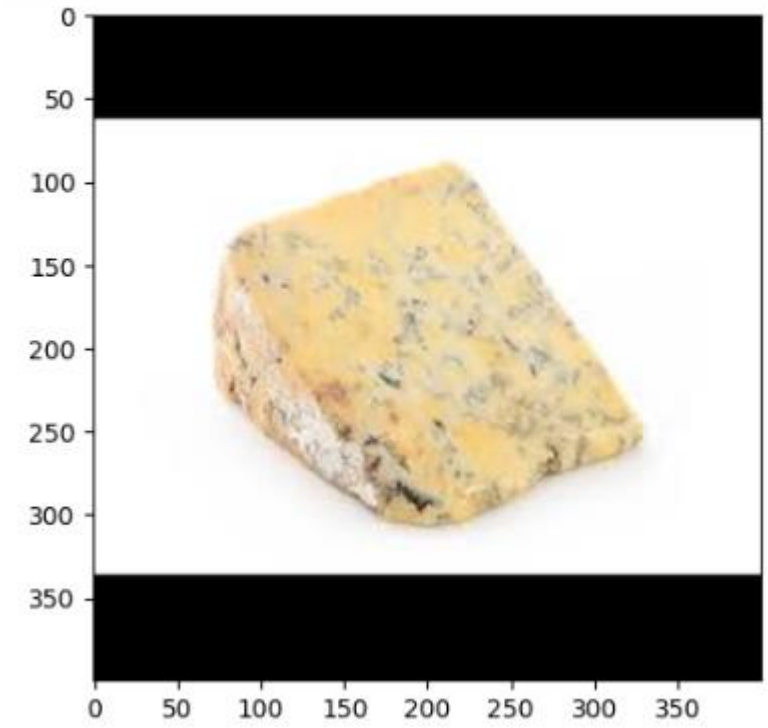
```
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt
```

```
from PIL import Image

def expend2square(pil_img, background_color):
    width, height = pil_img.size
    #print(width, height)

    # 정사각형이면 수정할 필요 없음
    if width == height:
        return pil_img

    # 너비가 높이보다 큰 경우 처리하는 코드
    elif width > height:
        # 너비와 같은 크기의 빈 이미지(result)를 생성
        result = Image.new(pil_img.mode, (width, width), background_color)
        # 원본 이미지를 생성한 빈 이미지의 왼쪽 상단에 붙이기
        result.paste(pil_img, (0, (width - height) // 2))
        return result

    # 높이가 너비보다 큰 경우 처리하는 코드
    else:
        result = Image.new(pil_img.mode, (height, height), background_color)
        # 원본 이미지를 생성한 빈 이미지의 상단 중앙에 붙이기
        result.paste(pil_img, ((height - width) // 2, 0))
        return result

        """
    .paste() 메서드는 result 이미지 객체에 pil_img 이미지 객체를 (0, (width - height) // 2) 위치에 붙여넣습니다.
        따라서, pil_img 이미지는 result 이미지의 왼쪽 위 모서리에서부터 (0, (width - height) // 2) 좌표로 이동하여 붙여지게 됩니다.
```

```
        """

def resize_with_padding(pil_img, new_size, background_color) :
    """
    입력된 이미지를 새 크기로 크기 조정하는 함수

    변수:
        pil_img (PIL.Image): 크기 조정할 이미지 객체 (PIL 이미지 객체)
        new_size (tuple): 새로운 크기 (너비, 높이)를 나타내는 튜플
        background_color (tuple): 확장된 영역의 배경 색상 (R, G, B)를 나타내는 튜플

    Returns:
        PIL.Image: 크기가 조정된 이미지 객체 (PIL 이미지 객체)
    """
    img = expend2square(pil_img, background_color)
    img = img.resize((new_size[0], new_size[1]), Image.ANTIALIAS)

    return img

img = Image.open("./blue_cheese_img_download/100_blue_cheese.png")

img_new = resize_with_padding(img, (400, 400), (0, 0, 0))
print(img_new.size)

plt.imshow(img_new)
plt.show()
```

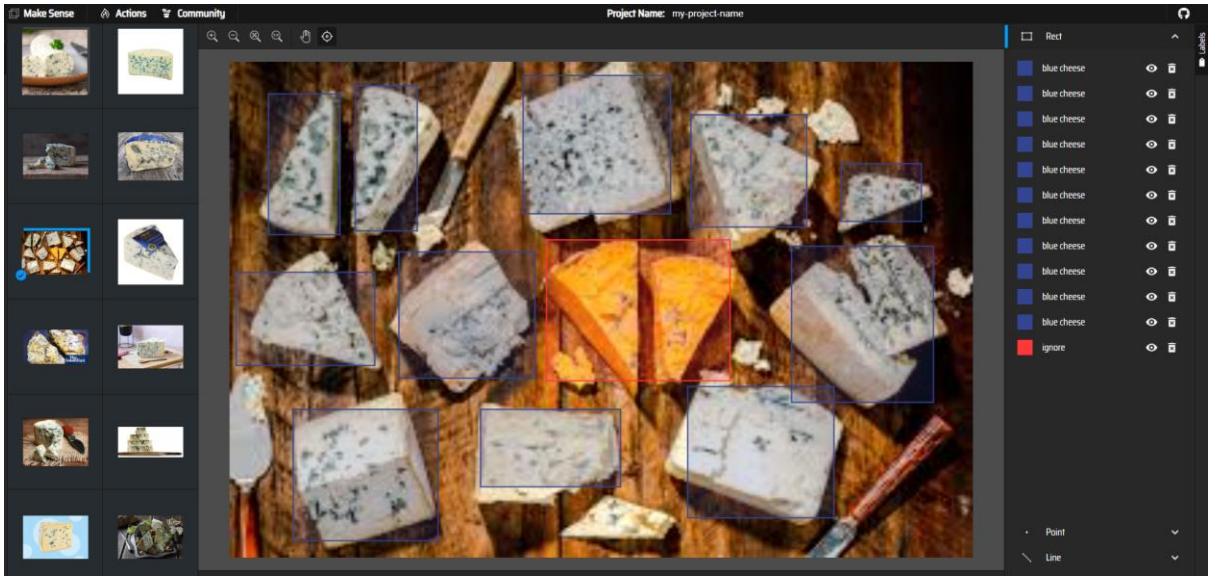
```
(400, 400)
```

```
C:#Users#Public#Documents#ESTsoft#CreatorTemp#ipykernel_16
and will be removed in Pillow 10 (2023-07-01). Use LANCZOS
  img = img.resize((new_size[0], new_size[1]), Image.ANTIA
```

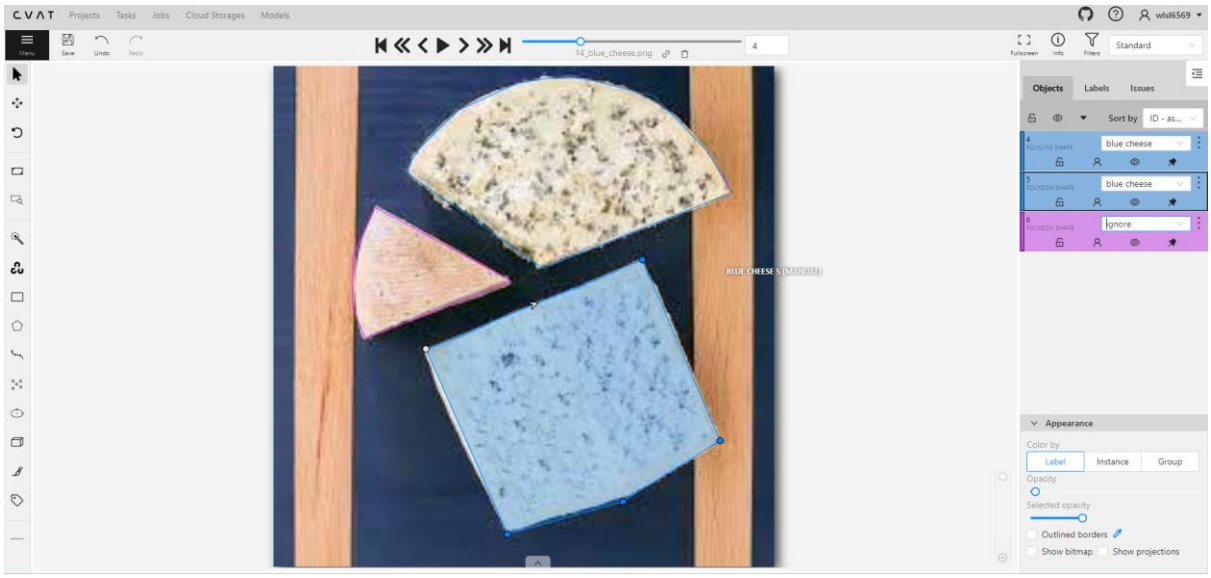


실습03. makesense에서 라벨링 툴 다뤄보기

makesense



Roboflow



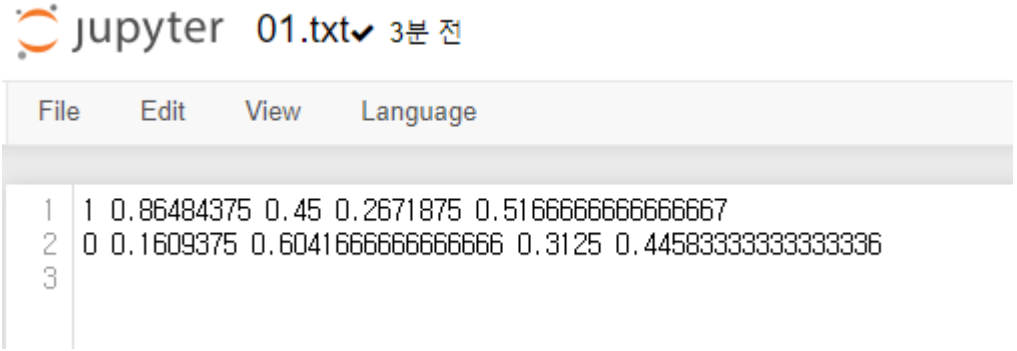
실습04. Json 파일 다루기 -> yolo 포맷 변경

```
import json
import os
import cv2
import matplotlib.pyplot as plt

# 1. JSON 경로 선언
json_path = "./instances_default.json"

# 2. JSON 읽기

with open(json_path, 'r', encoding='utf-8') as j : # 경로에 있는 파일을 'r' 모드(읽기 모드)로 열고 인코딩은 UTF-8로 지정
    json_data = json.load(j)

    # print(json_data)

# 3. catrgories, images, annotations -> COCO dataset 기준
categories_info = json_data['categories']
images_info = json_data['images']
annotations_info = json_data['annotations']
  #print(categories_info, images_info, annotations_info)
```

```
label_dict = {1 : 0, 2 : 1 }

# 4.image_info 정보 가져오기 name, width, height, image_id
for image_json in images_info :
    # print(images_json)
     image_id = image_json['id']
     image_name = image_json['file_name']
     image_width = image_json['width']
     image_height = image_json['height']

     image_path = os.path.join("./", image_name)
    # print(image_path)

     # image 읽기
     image = cv2.imread(image_path)
     image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# 5. bbox info 가져오기
     for anno_json in annotations_info :
         # print("anno_json >> ", anno_json)
          if image_id == anno_json['image_id'] :
               bbox = anno_json['bbox']
              # print("bbox >> ", bbox)

               # float -> int
               x = int(bbox[0])
               y = int(bbox[1])
               w = int(bbox[2])
               h = int(bbox[3])

               # 6. 라벨 정보 가져오기
               category_id = anno_json['category_id']
               label_number = label_dict[category_id]
              # print(label_number)

               # 7. xywh -> centerX, centerY, w, h 변환하기
               center_x = ((2 * x + w)/(2 * image_width))
               center_y = ((2 * y + h)/(2 * image_height))
               yolo_w = w/image_width
```

```
            yolo_h = h/image_height

            print(label_number, center_x, center_y, yolo_w, yolo_h)

            # yolo 라벨 이름은 이미지 이름과 동일해야한다
            # ex) aaa.png -> aaa.txt
            # 8. 텍스 파일 쓰기
            file_name_temp = image_name.replace(".jpg", "")
            print(file_name_temp)

            with open(f'{file_name_temp}.txt','a') as f:
                f.write(f"{label_number} {center_x} {center_y} {yolo_w} {yolo_h} ₩n")




            # 박스 그리기
#               cv2.rectangle(image, (x,y), (x+w, y+h), (0,255,0), 2)
#     plt.imshow(image)
#     plt.show()
# 제대로 됐으면 ok!
```

```
1 0.86484375 0.45 0.2671875 0.5166666666666667
01
0 0.1609375 0.6041666666666666 0.3125 0.44583333333333336
01
```



실습05. xml 파일 다루기

```
import os
import cv2
import matplotlib.pyplot as plt
from xml.etree.ElementTree import parse
```

```
def xml_read(xml_path) :
    root = parse(xml_path).getroot()

    image_info = root.findall("image") # findall로 image 태그 들어간 모든 정보 찾기

    for image in image_info :
        bbox = image.findall('box')

        # image width height
        image_width = image.attrib['width']
        image_height = image.attrib['height']
        print("이미지 크기 정보 >> ",image_width, image_height)

        # image name
        image_name = image.attrib['name']
        print("이미지 이름 : " , image_name)
        image_path = os.path.join("./", image_name)
        print("이미지 경로 >>", image_path)

        # image read
        image = cv2.imread(image_path)
        # BGR - RGB
        image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

        for box_info in bbox :
            label = box_info.attrib['label']
            print(label)

            xtl = box_info.attrib['xtl']
            ytl = box_info.attrib['ytl']
            xbr = box_info.attrib['xbr']
            ybr = box_info.attrib['ybr']
```

```
            # str 형태라서 바꿔 줘야함
            # string -> float
            xtl_f = float(xtl)
            ytl_f = float(ytl)
            xbr_f = float(xbr)
            ybr_f = float(ybr)

            # float -> int
            xtl_i = int(xtl_f)
            ytl_i = int(ytl_f)
            xbr_i = int(xbr_f)
            ybr_i = int(ybr_f)

# xtl: 바운딩 박스의 좌측 상단 모서리의 x 좌표
# ytl: 바운딩 박스의 좌측 상단 모서리의 y 좌표
# xbr: 바운딩 박스의 우측 하단 모서리의 x 좌표
# ybr: 바운딩 박스의 우측 하단 모서리의 y 좌표

            # 그리기
            image = cv2.rectangle(image, (xtl_i, ytl_i), (xbr_i, ybr_i), (0,255,0), 2)

            # 라벨 추가
            image = cv2.putText(image, label, (xtl_i, ytl_i-10),
            cv2.FONT_HERSHEY_COMPLEX_SMALL, 1, (0,255,255),2, cv2.LINE_AA)


    plt.imshow(image)
    plt.show()


xml_read("./annotations.xml")
```

```
이미지 크기 정보 >>  640 480
이미지 이름 :  01.jpg
이미지 경로 >> ./01.jpg
dog
cat
```