

cat\_dog\_person.yaml - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
path: /content/datasets/cat_dog_person # dataset root dir
train: images/train # train images (relative to 'path')
val: images/train # val images (relative to 'path')
test: # test images (optional)
```

# Classes

nc: 3 # number of classes

names: ['cat', 'dog', 'person'] # class names

# Download script/URL (optional)

download: [https://github.com/DrKAI/CV/raw/main/cat\\_dog\\_person.zip](https://github.com/DrKAI/CV/raw/main/cat_dog_person.zip)

## Custom yaml 구조

cat\_dog\_person.yaml - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
path: /content/datasets/cat_dog_person # dataset root dir
train: images/train # train images (relative to 'path')
val: images/train # val images (relative to 'path')
test: # test images (optional)
```

# Classes

nc: 3 # number of classes

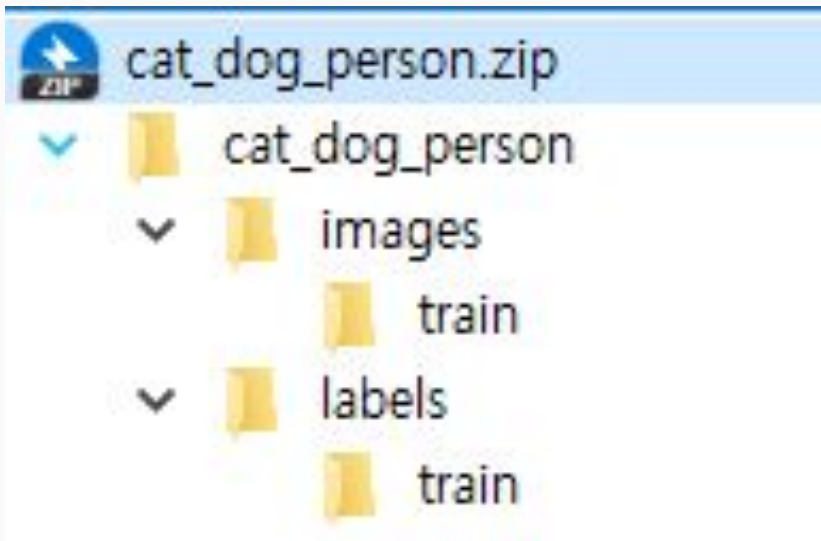
names: ['cat', 'dog', 'person'] # class names

# Download script/URL (optional)

download: [https://github.com/DrKAI/CV/raw/main/cat\\_dog\\_person.zip](https://github.com/DrKAI/CV/raw/main/cat_dog_person.zip)

## Custom yaml 구조

- 데이터셋 경로 설정
- 클래스 수 / 클래스 명
- 데이터셋 다운로드 경로



### Custom yaml 구조

다운받은 데이터셋의 구조가  
yaml 설정과 같아야 한다

# UltraLytics YOLO : Custom Data

**File Explorer Structure:**

- Myproject
  - images
    - train
    - val
  - labels
  - train
  - val

**\*ypjt.yaml - Windows 메모장**

```
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
path: /content/Myproject #<-- 내가 데이터로 사용하는 경로.
train: images/train # 내가 라벨링에 사용한 이미지들을 넣을 폴더
val: images/val # 혹시나 validation셋을 따로 구성한다면 val로, 아니면 train으로

# Classes
nc: 4 # 내가 사용할 클래스 종류가 몇개인지
names: ['pica', 'ggobook', 'eve', 'mu']
```

**메모장으로 yaml만들기**  
메모장 좌상단> 파일>  
다른이름으로 저장>  
파일형식을 모든파일>파일이름 뒤 yaml

**Annotation Information:**  
각 이미지에 해당하는 annotation정보(txt파일) 들을 넣을 폴더 (yaml 파일에는 명시 안함) (val은 따로 관리할 경우에만 val폴더 사용)

**File Save Dialog:**  
파일 이름(N): mypjt.yaml  
파일 형식(T): 모든 파일 (\*.\*)