

Getting Started

In this project, we are going to have a simulation about face recognition. The whole structure is divided into 4 parts:

- [The general process of working with Yolo version 8](#)
- [Working online](#)
- [Working offline and preparing the dataset from scratch](#)
- [Results](#)

General process

Installation

YOLOv8 released a package named "ultralytics", that you can install with the mentioned command below.

```
pip install ultralytics
```

Preparation

By cloning the data, we can have the overall structure, we can prepare this item with the following command: (of course, we will tell another method for preparation in the last part)

```
git clone https://github.com/SMSajadi99/Custom-Data-YOLOv8-Face-Detection.git
```

Start Training

You can start training YOLOv8 on custom data by using mentioned command below in the terminal/(command prompt).

```
yolo task=detect mode=train model=yolov8n.pt data=custom.yaml epochs=3  
imgsz=640
```

- `task = detect` (It can be segment or classify)
- `mode = train` (It can be predict or val)
- `model = yolov8n.pt` (It can be yolov8s/yolov8l/yolov8x)
- `epochs = 3` (It can be any number)

- `imgsz = 640` (It can be 320, 416, etc, but make sure it needs to be a multiple of 32)

Hint: In the project that I implemented, I implemented it with yolov8s weight. I doubt you can change this weight according to the value of your GPU. Wait for training to complete, and then do inference with newly created weights. Custom-trained weights will be saved in the folder path mentioned below. `runs/train/exp/weights/best.pt`

Start Test

Once your model is trained, you can use it to make predictions on new data. Use the mentioned command below for detection with custom weights.

```
yolo task=detect mode=predict model="runs/train/exp/weights/best.pt"
source="test.png"
```

or

```
yolo task=detect mode=predict model="runs/train/exp/weights/best.pt"
source="test.mp4"
```

Working online

To work online, open this [code](#) and execute it based on the first part.

Working offline

To prepare the data, you must download the data from the following [site](#) and place it in a folder like the following structure:

```
└─ Dataset_Original
   └─ wider_face_split.zip
   └─ WIDER_test.zip
   └─ WIDER_train.zip
   └─ WIDER_val.zip
```

Now `unzip` them and with this [code](#) that you have at your disposal, you can have the structure to start training. Run the following code:

```
python Prapertion.py
```

Folder structure:

After running the following code, the folder structure should be as follows: (It is clear that 3 folders train, valid and test are important.)

```

└─ Dataset_Original
    ├── test
    │   ├── images
    │   └── labels
    ├── train
    │   ├── images
    │   └── labels
    └── valid
        ├── images
        └── labels
    .
    .
    .

```

Now create a folder called `yolov8` and make the previous folders in the following format:

```

└─ images
    ├── test
    ├── train
    └── valid
└─ labels
    ├── train
    └── valid

```

In the `yolov8` folder, create a file named `custom.yaml` and set the following values in it: (Make sure to set the path according to your folder)

```

path:  /<PATH-TO>/yolov8/
train: images/train
test:  images/test
val:   images/valid

```

```

#Classes
names:
  0: face

```

Now all the items are ready and you can train and test it based on the [General process](#) section.

Hint: In the `ckpts` folder, I put two sample yolov8 weights based on `yolov8s.pth` and 25 trained epochs numbers that you can use as an evaluation.

Result