# vt2-webseg

## Project Structure

```
.
├── config
│   ├── main.yaml                 # Main configuration file
│   ├── model                     # Configurations for training model
│   │   ├── model1.yaml           # First variation of parameters to
train model
│   │   └── model2.yaml           # Second variation of parameters to
train model
│   └── process                   # Configurations for processing data
│       ├── process1.yaml         # First variation of parameters to
process data
│       └── process2.yaml         # Second variation of parameters to
process data
├── data
│   ├── final                     # data after training the model
│   ├── processed                 # data after processing
│   └── raw                       # raw data
├── docs                          # documentation for your project
├── .gitignore                    # ignore files that cannot commit to
Git
├── Makefile                      # store useful commands to set up
the environment
├── models                        # store models
├── notebooks                     # store notebooks
├── pyproject.toml                # Configure black

├── README.md                     # describe your project
├── src                           # store source code
│   ├── __init__.py               # make src a Python module
│   ├── process.py                # process data before training model
│   ├── train_model.py            # train model
│   └── utils.py                  # store helper functions
└── tests                         # store tests
    ├── __init__.py               # make tests a Python module
    ├── test_process.py           # test functions for process.py
    └── test_train_model.py       # test functions for train_model.py
```

## Set up the environment for training

1. For YOLO-WS with YOLOv5 training

```
python3 -m venv yolov5-WS
source yolov5-WS/bin/activate
```

```
pip install -r yolov5-requirements.txt
```

Then in the console:

```
python src/yolov5/run_yolowsv5.py --data
data/yolo_dataset_processed_full/dataset.yaml --cfg
src/yolov5/models/yolov5sWS.yaml --hyp src/hyp_yolows.yaml --imgs 512 --
batch-size 32 --epochs 300
```

2. For YOLO-WS with Ultralytics YOLOv11 training

```
python3 -m venv ultralytics
source ultralytics/bin/activate
pip install ultralytics
```

Then in the console (for standard parameters):

```
yolo detect train data=data/yolo_dataset_processed_full/dataset.yaml
model=yolo11s.pt imgsz=512 batch=32 epochs=300
```

1. For WEB-SAM training

```
python3 -m venv websam
source websam/bin/activate
pip install -r websam-requirements.txt
```

PROF

TODO: make this into a script

3. Install dependencies:

- To install all dependencies, run:

```
pip install -r requirements-dev.txt
```

- To install only production dependencies, run:

```
pip install -r requirements.txt
```

- To install a new package, run:

```
pip install <package-name>
```

## YOLOV5 TRAINING

```
python src/yolov5/run_yolowsv5.py --data
data/yolo_dataset_processed_small/dataset.yaml --cfg
src/yolov5/models/yolov5sWS.yaml --hyp src/hyp_yolows.yaml --imgs 1024 -
-rect --batch-size 16 --epochs 33
```

## YOLOV5 INFERENCE

```
python src/yolov5/detect.py --source
data/yolo_dataset_processed_small/images/val --weights
src/yolov5/runs/train/exp/weight
s/best.pt --conf 0.2
```

## View and alter configurations

To view the configurations associated with a Pythons script, run the following command:

```
python src/process.py --help
```

Output:

PROF

```
process is powered by Hydra.

== Configuration groups ==
Compose your configuration from those groups (group=option)

model: model1, model2
process: process1, process2


== Config ==
Override anything in the config (foo.bar=value)

process:
  use_columns:
  - col1
  - col2
model:
```

```
  name: model1
data:
  raw: data/raw/sample.csv
  processed: data/processed/processed.csv
  final: data/final/final.csv
```

To alter the configurations associated with a Python script from the command line, run the following:

```
python src/process.py data.raw=sample2.csv
```

## Auto-generate API documentation

To auto-generate API document for your project, run:

```
make docs
```