### **Code Instruction Document (Taking VOC07+12 Dataset as an Example)**

### **1. Dataset Preparation**

Use the VOC format for training. Download the VOC07+12 dataset in advance, unzip it, and place it in the root directory.

1. **Dataset Processing**

Modify annotation\_mode=2 in voc\_annotation.py.

Run voc\_annotation.py to generate 2007\_train.txt and 2007\_val.txt in the root directory.

**3. Network Training**

The default parameters in train.py are for training the VOC dataset. Directly run train.py to start training.

**4. Training Result Prediction**

Two files are required for prediction: yolo.py and predict.py.

First, modify model\_path and classes\_path in yolo.py. These two parameters **must be adjusted**.

model\_path points to the trained weight file in the logs folder.

classes\_path points to the .txt file corresponding to the detection categories.

After completing the modifications, run predict.py for detection. Enter the image path after execution to start detection.

**Training with Your Own Dataset**

**1.Dataset Preparation**

Use the VOC format for training. Prepare your dataset in advance:

Place label files in VOCdevkit/VOC2007/Annotations/.

Place image files in VOCdevkit/VOC2007/JPEGImages/.

**2. Dataset Processing**

After organizing the dataset, use voc\_annotation.py to generate 2007\_train.txt and 2007\_val.txt.

Modify parameters in voc\_annotation.py. For the first training, you can initially only modify classes\_path, which points to the .txt file of detection categories.

Create a cls\_classes.txt file to list the categories you need to detect. Example content for model\_data/cls\_classes.txt:

cat

dog

…

Modify classes\_path in voc\_annotation.py to match cls\_classes.txt, then run voc\_annotation.py.

**3.Network Training**

Multiple training parameters are configured in train.py (refer to the comments after downloading the repository). The most critical parameter is still classes\_path in train.py.

classes\_path must point to the same .txt file as in voc\_annotation.py! This **must be modified** when training with your own dataset.

After modifying classes\_path, run train.py to start training. After multiple epochs, weights will be generated in the logs folder.

#### **4. Training Result Prediction**

The prediction process is the same as above:

Modify model\_path and classes\_path in yolo.py.

model\_path points to the weight file in the logs folder.

classes\_path points to the category .txt file.

Run predict.py and enter the image path to perform detection.

#### **5. The method proposed in this paper is mainly implemented in darknet and YOLO.**