WiDFS 3.0 Reproduction Instructions

1. Folder Structure

(1) Delay_Doppler_Validation/

- Contains scripts for extracting Delay, Doppler, and Micro-Doppler features from single-target human motion datasets.
- These results correspond to *Section VIII.A (Delay–Doppler Feature Extraction Quality)* in the paper.

(2) Activity_Recognition_Validation/

- Contains scripts for validating the recognition performance of our WiDFS 3.0 scheme on the Widar 3.0 gesture recognition dataset.
- These results correspond to *Section VIII.B (Single-Target Sensing Performance*) in the paper.

2. Python Environment Setup

(1) Install Python 3.10 – 3.12

(2) Navigate to the project root directory and install dependencies:

pip install -r requirements.txt

(3) Core feature extraction algorithms are provided as precompiled Python wheels for Linux, Windows, and macOS. (Source code can be released later.)

3. Installing Core Interfaces

(1) Delay–Doppler Validation

Go to Delay_Doppler_Validation/ and install the WiDFS 3.0 core algorithm interface. Example for Linux + Python 3.10:

cd Delay_Doppler_Validation/wheels-linux-x86_64

pip install srcc-0.1.0-cp310-cp310-manylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl

(2) Activity Recognition Validation

Go to Activity_Recognition_Validation/ and install the feature generation core algorithm interface (different wheel). Example for Linux + Python 3.10:

cd Activity_Recognition_Validation/wheels-linux-x86_64

pip install mdproc-0.1.0-cp310-cp310-manylinux1_x86_64.manylinux_2_28_x86_64.manylinux_2_5_x86_64.whl

(For other systems and Python versions, install the corresponding wheel provided in the folder.)

4. Running the Programs

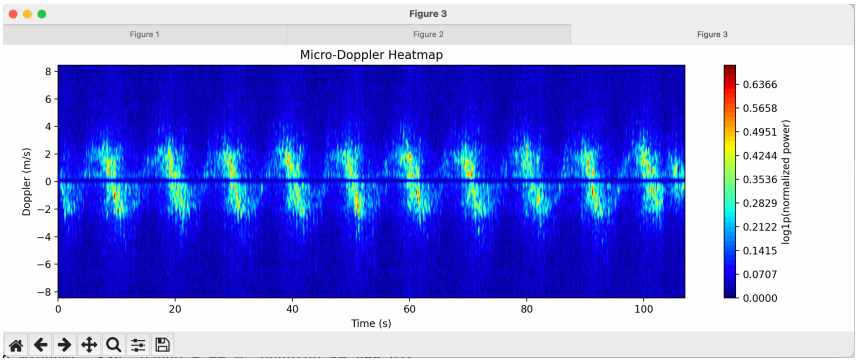
(1) Delay–Doppler Validation

Run: **python WiDFS3_Feature_Extractor.py**

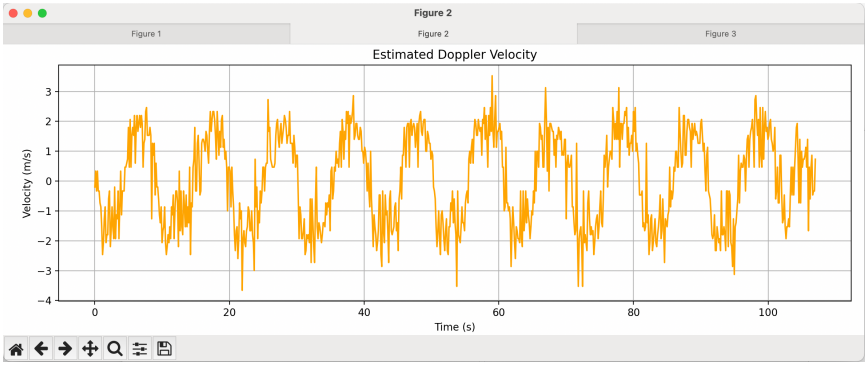
Output:

- Extracted Micro-Doppler spectrograms
- Delay and Doppler features

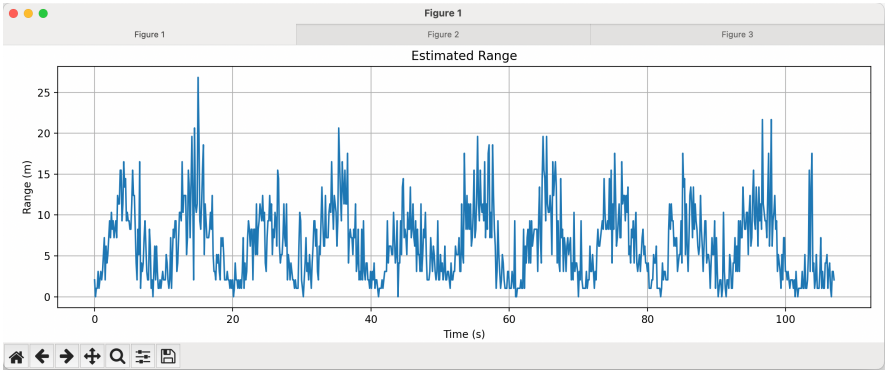
Micro-Doppler



Doppler Velocity

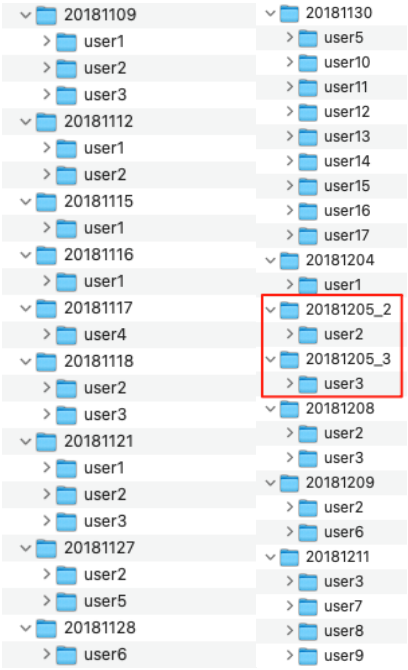


Range



4.2 Activity_Recognition_Validation

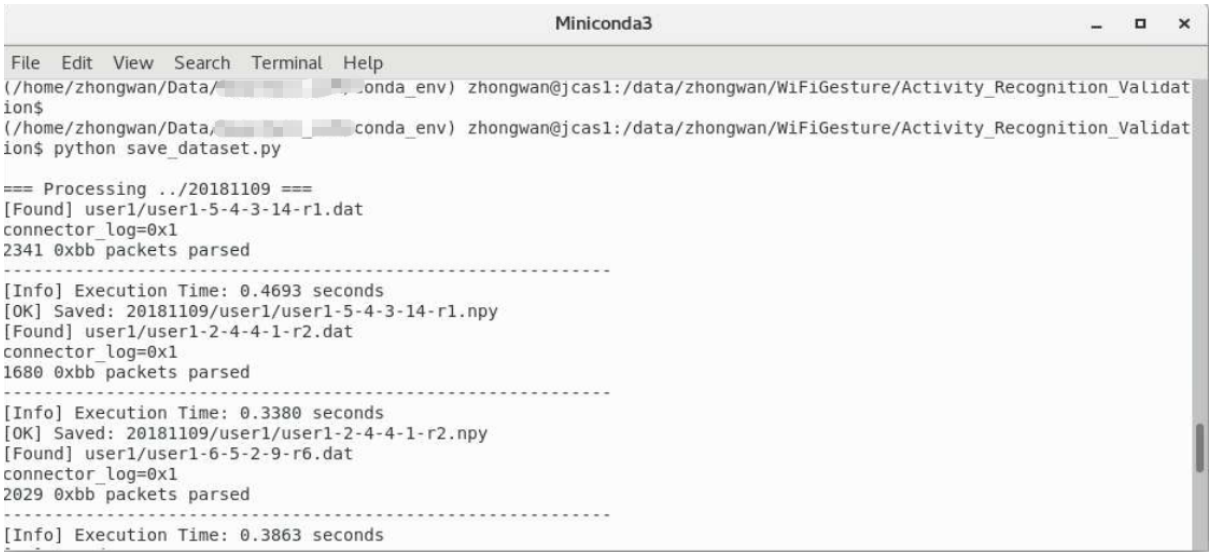
(1) Feature Generation (~42.3 GB size)



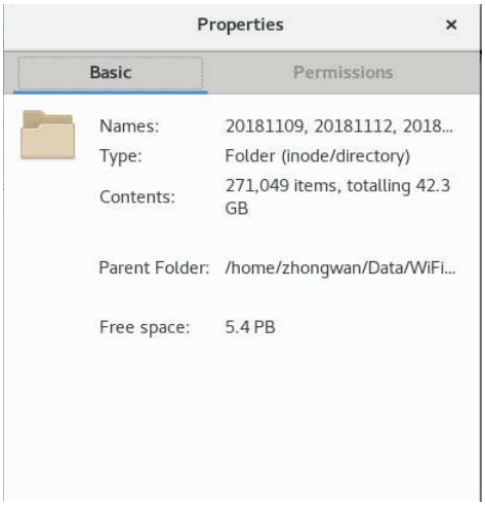
- Split the original folder 20181205 in Widar3.0 into two subfolders:
  - 20181205_2 containing only user2
  - 20181205_3 containing only user3
- In the same directory as the Widar3.0 dataset, place the Activity_Recognition_Validation folder. Then navigate into this folder and run:

python save_dataset.py

On our server, the feature extraction time for each .dat file (with 3 Rx antennas) is about 0.4 seconds.

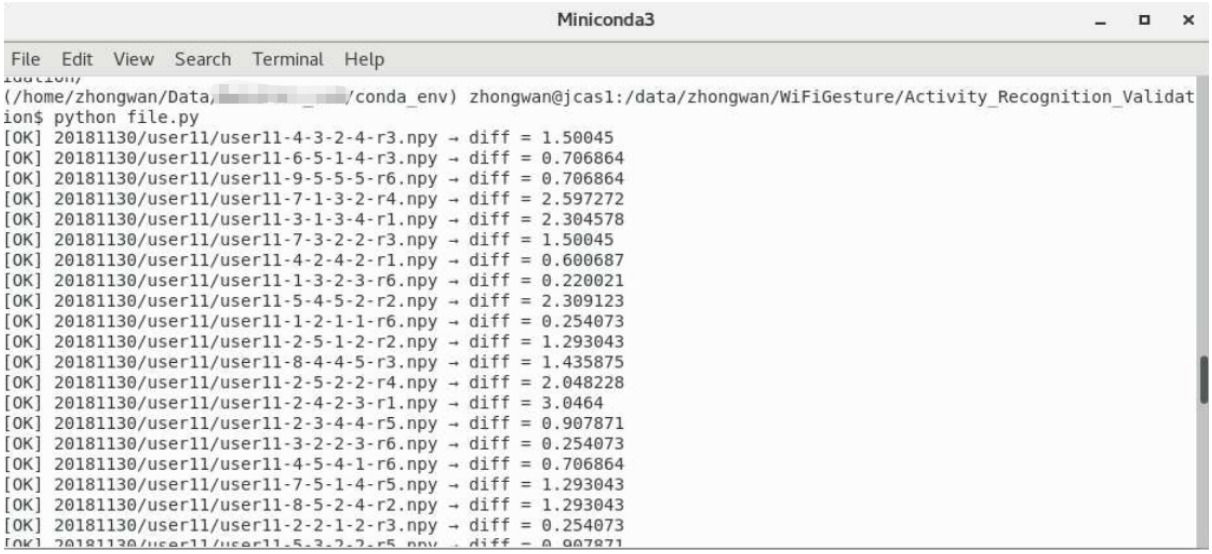


- After several hours, features will be generated. The feature directory structure will mirror that of the original Widar 3.0 CSI dataset.



(2) CSV Dataset Generation

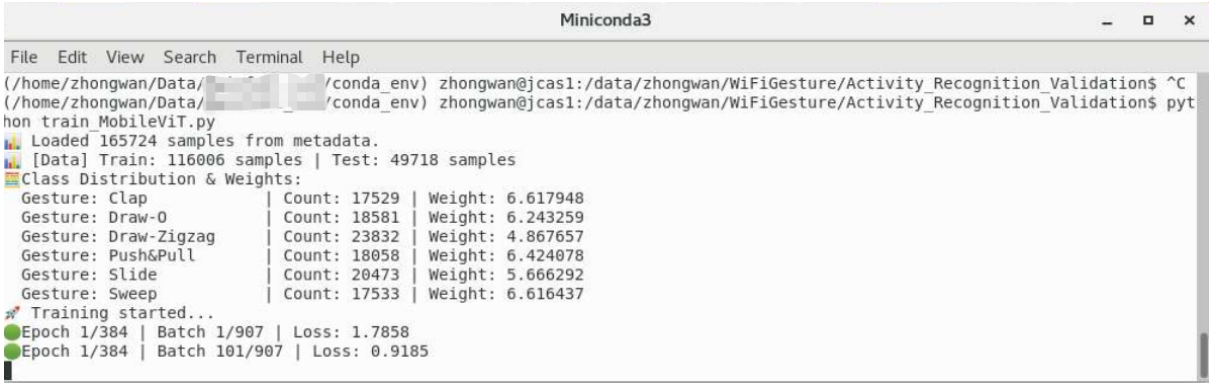
Run: **python file.py**



This will produce the dataset file gesture_metadata.csv.

(3) Model Training and Testing (70%–30% split)

Run: **python train_MobileVit.py**



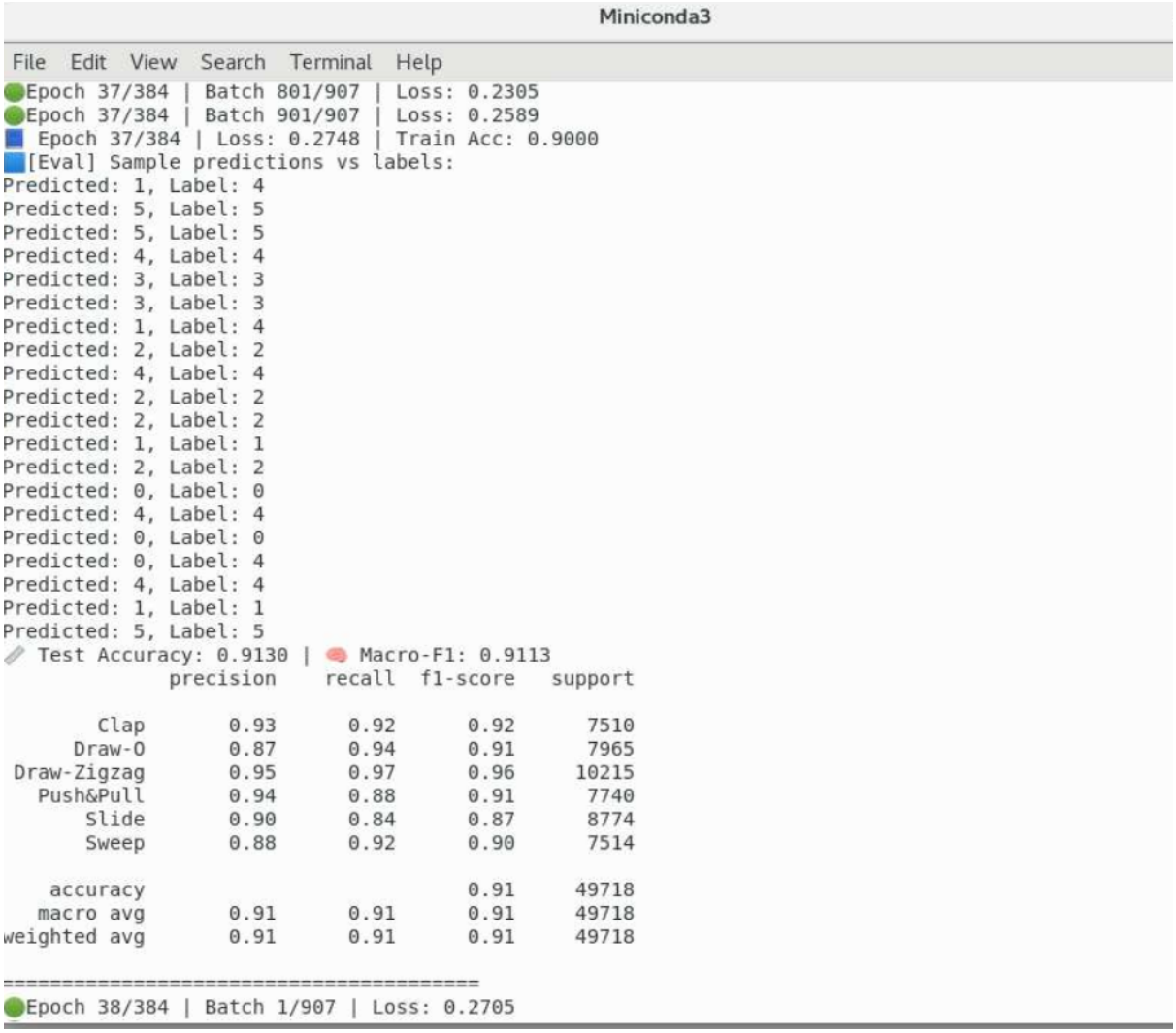
Miniconda3

File Edit View Search Terminal Help

```
🟢Epoch 20/384 | Batch 601/907 | Loss: 0.2517
🟢Epoch 20/384 | Batch 701/907 | Loss: 0.2996
🟢Epoch 20/384 | Batch 801/907 | Loss: 0.3697
🟢Epoch 20/384 | Batch 901/907 | Loss: 0.3885
🟦 Epoch 20/384 | Loss: 0.3272 | Train Acc: 0.8802
🟦[Eval] Sample predictions vs labels:
Predicted: 0, Label: 4
Predicted: 0, Label: 0
Predicted: 5, Label: 5
Predicted: 2, Label: 2
Predicted: 3, Label: 3
Predicted: 0, Label: 0
Predicted: 2, Label: 2
Predicted: 4, Label: 4
Predicted: 5, Label: 5
Predicted: 1, Label: 1
Predicted: 0, Label: 5
Predicted: 0, Label: 0
Predicted: 3, Label: 3
Predicted: 4, Label: 4
Predicted: 2, Label: 2
Predicted: 5, Label: 5
Predicted: 5, Label: 5
Predicted: 3, Label: 3
Predicted: 0, Label: 0
Predicted: 5, Label: 5
🩹 Test Accuracy: 0.8983 | 🧠 Macro-F1: 0.8963
              precision    recall  f1-score   support

        Clap       0.90      0.93      0.91      7510
      Draw-O       0.83      0.95      0.88      7965
 Draw-Zigzag       0.96      0.95      0.96     10215
   Push&Pull       0.93      0.87      0.90      7740
       Slide       0.88      0.81      0.85      8774
       Sweep       0.88      0.88      0.88      7514

    accuracy                           0.90     49718
   macro avg       0.90      0.90      0.90     49718
weighted avg       0.90      0.90      0.90     49718
```

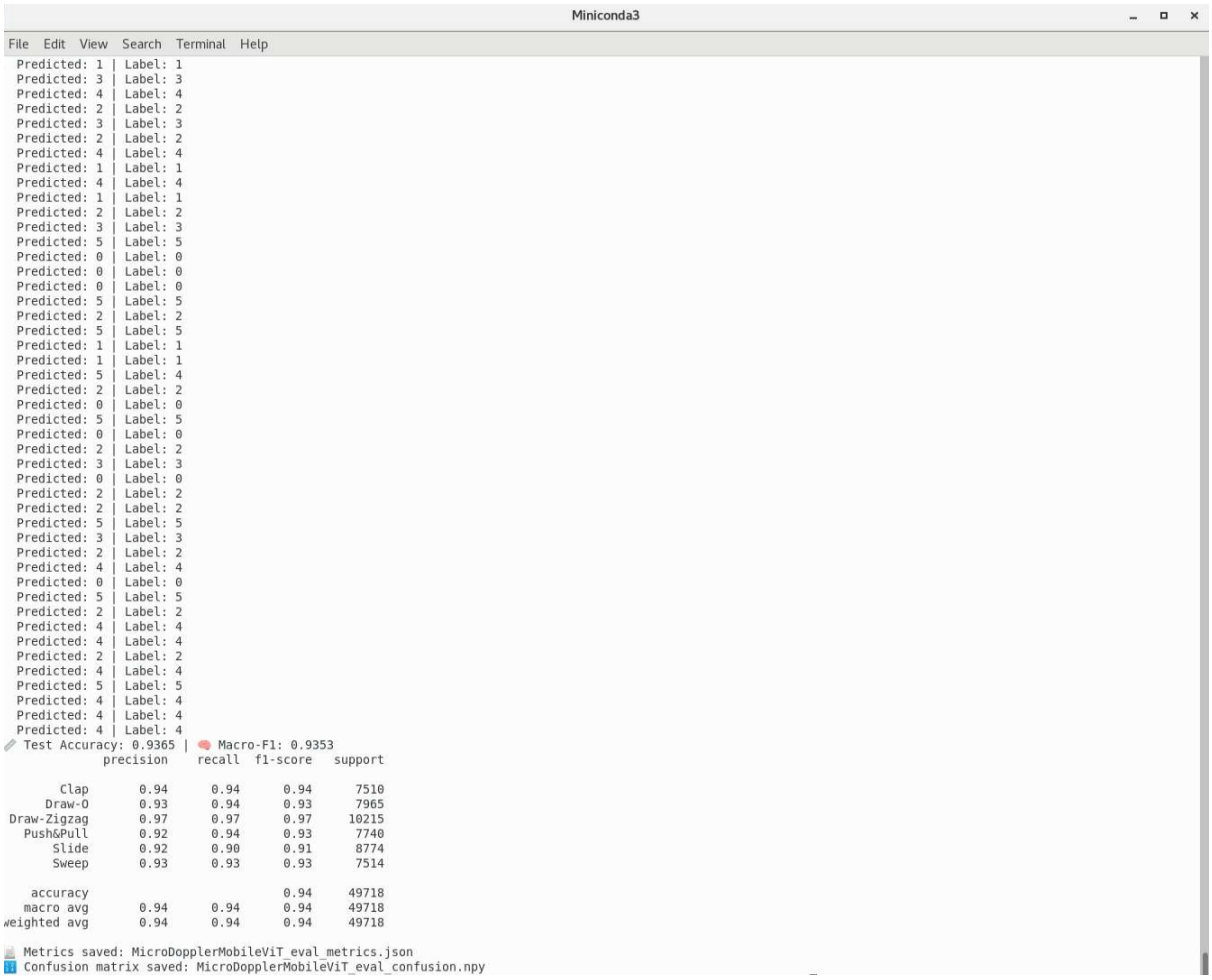


The training and testing process will start automatically. A new folder saved_models_gesture will be created, containing trained models and result logs.

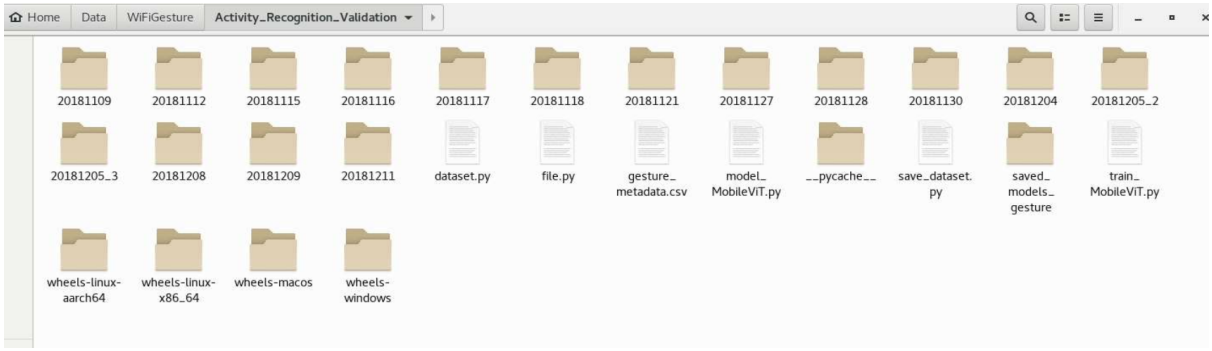
(4) Model Testing Only (70%–30% split)

Using the trained model, you can directly evaluate performance.

Run: **python evaluate_only.py**



In addition, the main folder contains both the extracted features and the source code.



The submitted folder saved_models_gesture contains our trained MobileViT models, along with the corresponding confusion matrices and the metrics.json file.

5. Contact

If you encounter any problems during reproduction, please contact:

zhongqin.wang@uts.edu.au, wzqhost@gmail.com