

# 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

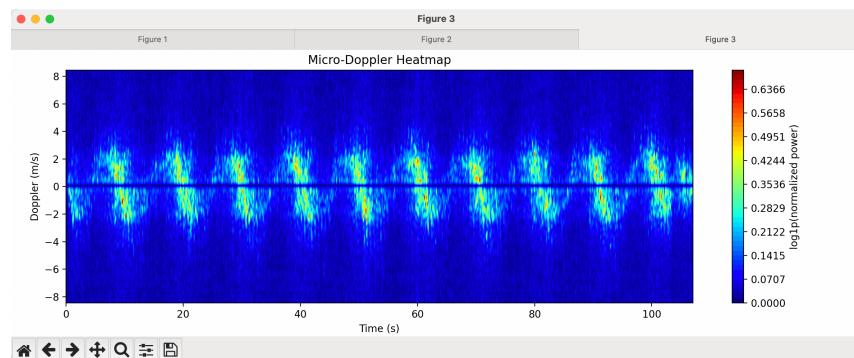
### (1) Delay–Doppler Validation

Run: `python WiDFS3_Feature_Extractor.py`

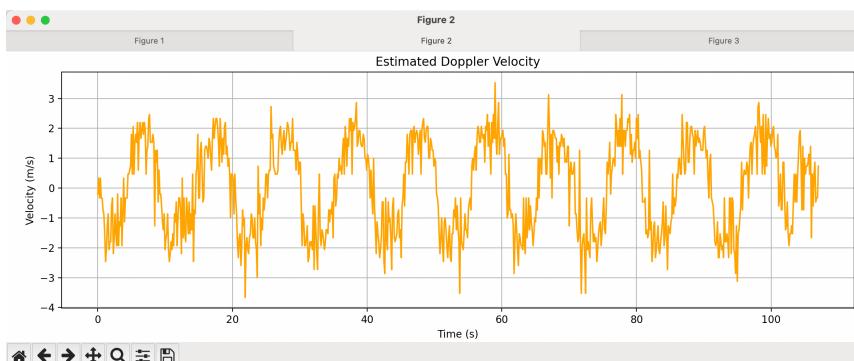
Output:

- Extracted Micro-Doppler spectrograms
- Delay and Doppler features

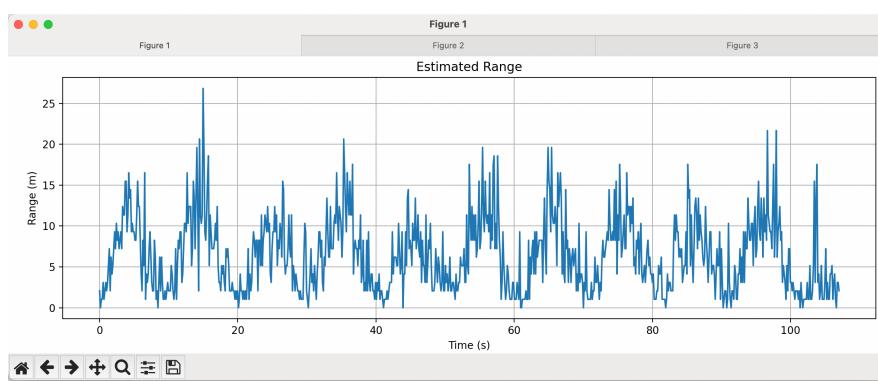
#### Micro-Doppler



#### Doppler Velocity

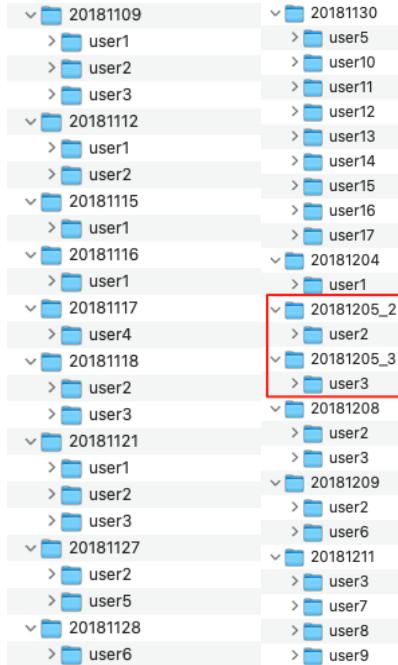


#### Range



## 4.2 Activity\_Recognition\_Validation

### (1) Feature Generation (~42.3 GB size)



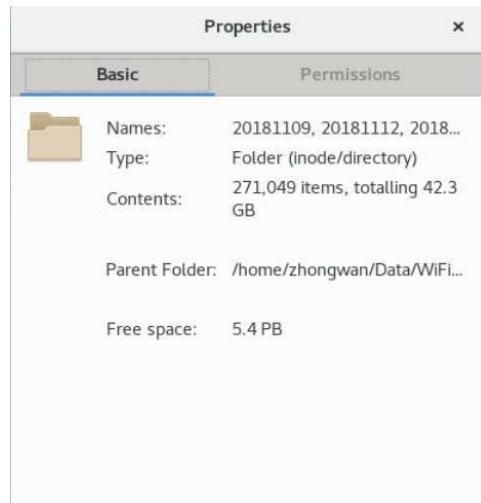
- Split the original folder 20181205 in Widar3.0 into two subfolders:
  - 20181205\_2 containing only user2
  - 20181205\_3 containing only user3
- In the same directory as the Widar3.0 dataset, place the Activity\_Recognition\_Validation folder. Then navigate into this folder and run:

```
python save_dataset.py
```

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

```
Miniconda3
File Edit View Search Terminal Help
(/home/zhongwan/Data/.../onda_env) zhongwan@jcasl:/data/zhongwan/WiFiGesture/Activity_Recognition_Validation$ (/home/zhongwan/Data/.../conda_env) zhongwan@jcasl:/data/zhongwan/WiFiGesture/Activity_Recognition_Validation$ python save_dataset.py
== Processing ../20181109 ==
[Found] user1/user1-5-4-3-14-r1.dat
connector_log=0x1
2341 0xbb packets parsed
-----
[Info] Execution Time: 0.4693 seconds
[OK] Saved: 20181109/user1/user1-5-4-3-14-r1.npy
[Found] user1/user1-2-4-4-1-r2.dat
connector_log=0x1
1680 0xbb packets parsed
-----
[Info] Execution Time: 0.3380 seconds
[OK] Saved: 20181109/user1/user1-2-4-4-1-r2.npy
[Found] user1/user1-6-5-2-9-r6.dat
connector_log=0x1
2029 0xbb packets parsed
-----
[Info] Execution Time: 0.3863 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**

```

Miniconda3
File Edit View Search Terminal Help
/home/zhongwan/Data/.../conda_env) zhongwan@jcas1:/data/zhongwan/WiFiGesture/Activity_Recognition_Validation$ python file.py
[OK] 20181130/user11/user11-4-3-2-4-r3.npy → diff = 1.50045
[OK] 20181130/user11/user11-6-5-1-4-r3.npy → diff = 0.706864
[OK] 20181130/user11/user11-9-5-5-5-r6.npy → diff = 0.706864
[OK] 20181130/user11/user11-7-1-3-2-r4.npy → diff = 2.597272
[OK] 20181130/user11/user11-3-1-3-4-r1.npy → diff = 2.304578
[OK] 20181130/user11/user11-7-3-2-2-r3.npy → diff = 1.50045
[OK] 20181130/user11/user11-4-2-4-2-r1.npy → diff = 0.600687
[OK] 20181130/user11/user11-1-3-2-3-r6.npy → diff = 0.220021
[OK] 20181130/user11/user11-5-4-5-2-r2.npy → diff = 2.309123
[OK] 20181130/user11/user11-1-2-1-1-r6.npy → diff = 0.254073
[OK] 20181130/user11/user11-2-5-1-2-r2.npy → diff = 1.293043
[OK] 20181130/user11/user11-8-4-4-5-r3.npy → diff = 1.435875
[OK] 20181130/user11/user11-2-5-2-2-r4.npy → diff = 2.048228
[OK] 20181130/user11/user11-2-4-2-3-r1.npy → diff = 3.0464
[OK] 20181130/user11/user11-2-3-4-4-r5.npy → diff = 0.907871
[OK] 20181130/user11/user11-3-2-2-3-r6.npy → diff = 0.254073
[OK] 20181130/user11/user11-4-5-4-1-r6.npy → diff = 0.706864
[OK] 20181130/user11/user11-7-5-1-4-r5.npy → diff = 1.293043
[OK] 20181130/user11/user11-8-5-2-4-r2.npy → diff = 1.293043
[OK] 20181130/user11/user11-2-2-1-2-r3.npy → diff = 0.254073
[OK] 20181130/user11/user11-5-3-2-2-r5.npy → diff = 0.907871

```

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
/home/zhongwan/Data/.../conda_env) zhongwan@jcas1:/data/zhongwan/WiFiGesture/Activity_Recognition_Validation$ ^C
/home/zhongwan/Data/.../conda_env) zhongwan@jcas1:/data/zhongwan/WiFiGesture/Activity_Recognition_Validation$ python train_MobileViT.py
Loaded 165724 samples from metadata.
[Data] Train: 116006 samples | Test: 49718 samples
Class Distribution & Weights:
Gesture: Clap | Count: 17529 | Weight: 6.617948
Gesture: Draw-0 | Count: 18581 | Weight: 6.243259
Gesture: Draw-Zigzag | Count: 23832 | Weight: 4.867657
Gesture: Push&Pull | Count: 18058 | Weight: 6.424078
Gesture: Slide | Count: 20473 | Weight: 5.666292
Gesture: Sweep | Count: 17533 | Weight: 6.616437
Training started...
Epoch 1/384 | Batch 1/907 | Loss: 1.7858
Epoch 1/384 | Batch 101/907 | Loss: 0.9185

```

### 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-0      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      0.90      0.90      49718
macro avg       0.90      0.90      0.90      49718
weighted avg    0.90      0.90      0.90      49718
```

```
Miniconda3
File Edit View Search Terminal Help
Epoch 37/384 | Batch 801/907 | Loss: 0.2305
Epoch 37/384 | Batch 901/907 | Loss: 0.2589
Epoch 37/384 | Loss: 0.2748 | Train Acc: 0.9000
[Eval] Sample predictions vs labels:
Predicted: 1, Label: 4
Predicted: 5, Label: 5
Predicted: 5, Label: 5
Predicted: 4, Label: 4
Predicted: 3, Label: 3
Predicted: 3, Label: 3
Predicted: 1, Label: 4
Predicted: 2, Label: 2
Predicted: 4, Label: 4
Predicted: 2, Label: 2
Predicted: 2, Label: 2
Predicted: 1, Label: 1
Predicted: 2, Label: 2
Predicted: 0, Label: 0
Predicted: 4, Label: 4
Predicted: 0, Label: 0
Predicted: 0, Label: 4
Predicted: 4, Label: 4
Predicted: 1, Label: 1
Predicted: 5, Label: 5
Test Accuracy: 0.9130 | Macro-F1: 0.9113
precision    recall   f1-score   support
Clap          0.93      0.92      0.92      7510
Draw-0        0.87      0.94      0.91      7965
Draw-Zigzag   0.95      0.97      0.96     10215
Push&Pull    0.94      0.88      0.91      7740
Slide         0.90      0.84      0.87      8774
Sweep         0.88      0.92      0.90      7514
accuracy      0.91      0.91      0.91      49718
macro avg     0.91      0.91      0.91      49718
weighted avg  0.91      0.91      0.91      49718
=====
Epoch 38/384 | Batch 1/907 | Loss: 0.2705
```

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.

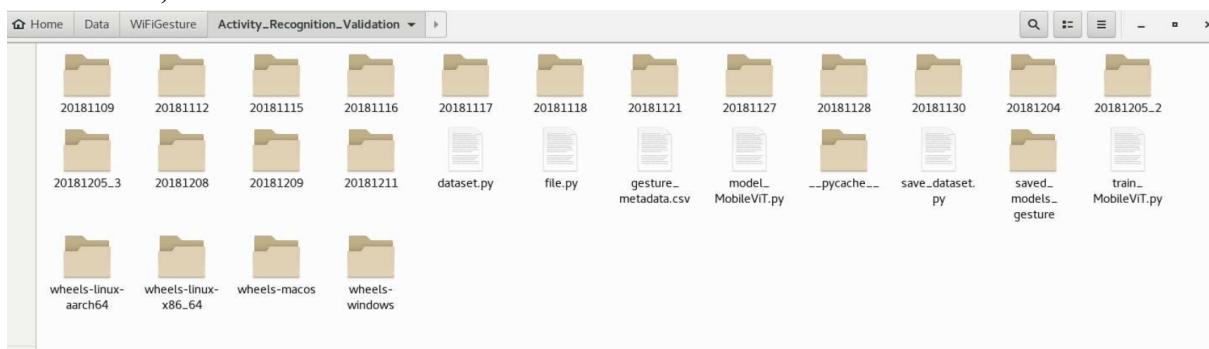
Run: **python evaluate\_only.py**

Miniconda3

File Edit View Search Terminal Help

```
Predicted: 1 | Label: 1
Predicted: 3 | Label: 3
Predicted: 4 | Label: 4
Predicted: 2 | Label: 2
Predicted: 3 | Label: 3
Predicted: 2 | Label: 2
Predicted: 4 | Label: 4
Predicted: 1 | Label: 1
Predicted: 4 | Label: 4
Predicted: 1 | Label: 1
Predicted: 2 | Label: 2
Predicted: 3 | Label: 3
Predicted: 5 | Label: 5
Predicted: 0 | Label: 0
Predicted: 0 | Label: 0
Predicted: 5 | Label: 5
Predicted: 2 | Label: 2
Predicted: 5 | Label: 5
Predicted: 1 | Label: 1
Predicted: 1 | Label: 1
Predicted: 5 | Label: 4
Predicted: 2 | Label: 2
Predicted: 0 | Label: 0
Predicted: 5 | Label: 5
Predicted: 0 | Label: 0
Predicted: 2 | Label: 2
Predicted: 3 | Label: 3
Predicted: 0 | Label: 0
Predicted: 2 | Label: 2
Predicted: 2 | Label: 2
Predicted: 5 | Label: 5
Predicted: 3 | Label: 3
Predicted: 2 | Label: 2
Predicted: 4 | Label: 4
Predicted: 0 | Label: 0
Predicted: 5 | Label: 5
Predicted: 2 | Label: 2
Predicted: 4 | Label: 4
Predicted: 4 | Label: 4
Predicted: 2 | Label: 2
Predicted: 4 | Label: 4
Predicted: 5 | Label: 5
Predicted: 4 | Label: 4
Predicted: 4 | Label: 4
Test Accuracy: 0.9365 | Macro-F1: 0.9353
precision recall f1-score support
Clap 0.94 0.94 0.94 7510
Draw-O 0.93 0.94 0.93 7965
Draw-Zigzag 0.97 0.97 0.97 10215
Push&Pull 0.92 0.94 0.93 7740
Slide 0.92 0.90 0.91 8774
Sweep 0.93 0.93 0.93 7514
accuracy 0.94 0.94 0.94 49718
macro avg 0.94 0.94 0.94 49718
weighted avg 0.94 0.94 0.94 49718
Metrics saved: MicroDopplerMobileViT_eval_metrics.json
Confusion matrix saved: MicroDopplerMobileViT_eval_confusion.npy
```

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`