**Project Overview**

We propose a mid-infrared plasmonic sensor combined with an s-CFW-enhanced convolutional neural network (CNN) model for the quantitative prediction of protein secondary structures in aqueous solutions. This project provides the related code and test files for implementing this process, offering researchers resources for reference and use.

**System Requirements**

* Windows 11 Professional
* Python 3.11.9
* PyCharm 2023
* PyTorch

**Codebase**

**s-CFW**

The **s-CFW** folder contains code and test data related to the s-CFW-enhanced infrared spectra. The specific contents are as follows:

* **CFW.py**: Returns the enhanced spectral wavenumber and peak intensity based on compensation parameters.
* **test.py**: Testing script that calls CFW.py for enhancement processing.
* **CFW test data.xlsx**: Infrared spectral data provided for testing.
* **test\_output.xlsx**: The enhanced spectral data output.

**Pre-training Parameters**

The **Pre-training parameter** folder contains parameters generated during the model pre-training process. The specific contents are as follows:

* **CFW**: Contains the weights and normalization parameters obtained from training with s-CFW-enhanced infrared spectra.
* **NO\_CFW**: Contains the weights and normalization parameters obtained from training with raw infrared spectra.

**Fine-tuning Parameters**

The **Fine\_tuning parameter** folder contains the parameters generated during the model fine-tuning process.

Due to the variation in baseline spectra (labeled as 0 0 0) across different sensors during testing, we replaced the baseline spectra for each sensor individually in the fine-tuning dataset. The specific contents are as follows:

* **CFW**: Contains the weights and normalization parameters obtained from training with s-CFW-enhanced infrared spectra, corresponding to **FIg. 1d** in the manuscript.
* **CFW\_2**: Contains the weights and normalization parameters obtained from training with s-CFW-enhanced infrared spectra, corresponding to **FIg. 4f** in the manuscript.
* **NO\_CFW**: Contains the weights and normalization parameters obtained from training with raw infrared spectra, corresponding to **FIg. 3d** and **FIg. 4e** in the manuscript.

**Test Set**

The **Test set** folder contains some of the raw data from the manuscript, which can be used for model testing. The specific contents are as follows:

* **FIg.1d\_data.csv**: Raw infrared spectral data of the protein corresponding to **FIg. 1d** (s-CFW enhanced).
* **FIg.3d\_data.csv**: Raw infrared spectral data of the protein corresponding to **FIg. 3d** (not s-CFW enhanced).
* **FIg.4e\_data.csv**: Raw infrared spectral data of the protein corresponding to **FIg. 4e** (not s-CFW enhanced).
* **FIg.4f\_data.csv**: Raw infrared spectral data of the protein corresponding to **FIg. 4f** (s-CFW enhanced).
* **Fig. S20**: Raw infrared spectral data of the protein corresponding to **Fig. S20** (s-CFW enhanced).

**Pre\_train**

The **Pre\_train** folder contains the code related to model pre-training. The specific contents are as follows:

* **data\_pre.py**: Data preprocessing module, responsible for data normalization and the division of the dataset into training, validation, and test sets.
* **model\_stru.py**: Defines the network structure, including all layers and parameter settings of the model.
* **model\_train.py**: Implements the model training process.
* **main.py**: Integrates and calls the above three parts, responsible for executing the entire pre-training process.

**Fine\_tune**

The **Fine\_tune** folder contains the code related to model fine-tuning. The specific contents are as follows:

* **data\_pre.py**: Data preprocessing module, responsible for data normalization and the division of the dataset into training, validation, and test sets.
* **model\_stru.py**: Defines the network structure, including all layers and parameter settings of the model.
* **fine\_tune.py**: Freezes certain convolutional layers and fine-tunes the model to adapt to specific tasks and datasets.

**Prediction**

The **Prediction** folder contains the code related to model testing. Using the **Fine\_tuning parameter** and **Test\_set** files, it allows reproduction of the results in **FIg. 1d**, **FIg. 3d**, **FIg. 4e**, and **FIg. 4f** from the manuscript. The specific contents are as follows:

* **model\_stru.py**: Defines the network structure, including all layers and parameter settings of the model.
* **Test\_data\_prediction.py**: Model testing code that accepts s-CFW enhanced infrared spectra as input and outputs the proportions of the three protein secondary structures.
* **prediction\_FIg.1d.xlsx**: Reproduction results for **FIg. 1d**.
* **prediction\_FIg.3d.xlsx**: Reproduction results for **FIg. 3d**.
* **prediction\_FIg.4e.xlsx**: Reproduction results for **FIg. 4e**.
* **prediction\_FIg.4f.xlsx**: Reproduction results for **FIg. 4f**.
* **prediction\_Fig. S20.xlsx**: Reproduction results for **Fig. S20**.