OptiSVR Spectral Refractive Index Prediction System

A spectrometer refractive index prediction system based on artificial intelligence algorithm.

0. Update Notes

This version has largely resolved the bug that may cause system freezing or crashing during training, though the possibility cannot be entirely ruled out. Additionally, the progress bar display has been improved to make it more simple and straightforward.

1. Software Overview

This system is a spectrometer refractive index prediction system based on artificial intelligence algorithms. It employs various unsupervised learning clustering methods (K-Means & SOM) and a Support Vector Regression (SVR) algorithm, combined with the deep learning ResNet50 network model for feature extraction from incidence angle-deviation angle images. Through multiple training sessions and hyperparameter adjustments, the best weights are obtained, and the optimal pre-trained model is saved.

The prediction approach involves interpolating measured data, converting it into incidence angle-deviation angle images, and then predicting the image curves, enabling accurate prediction of the spectrometer's refractive index characteristics.

2. System Requirements

- Operating System: Windows 10/11
- Recommended Python 3.9.6, with necessary Python libraries installed (see Installation Guide)
- Recommended Hardware: CPU 16+ cores, RAM 16GB+

3. Installation Guide

1. Install Python 3.9.6 (Download from Python official website)
2. Install dependency libraries: Run the `requirements.bat` file. The `requirements.txt` file contains the detailed list of dependent libraries.

4. Software Startup

Run the main program file `Spectral_Refractive_Index_Prediction_System.py`. The system will display a welcome screen and enter the main interface after an estimated 5-8 seconds. Upon first run, the system will automatically initialize the relevant directories and configuration files.

5. Main Features

5.1 User System

- **Function**: A new user system handles various user operations such as user registration, login, password reset, password change, guest login, profile editing, permission description, permission management, user management, logout, etc.
- **Operations**:
    1. Upon starting the software, the user login window will pop up first.
    2. Users can click the avatar in the upper right corner to reveal a floating box for editing profiles, viewing permission descriptions, requesting permission upgrades, or logging out. The default administrator account is `admin` with the password `admin123`.
    3. Only administrators can operate on other users' accounts and permissions.

5.2 Generate Theoretical Data

- **Function**: Generates theoretical refractive index images based on spectrometer physical characteristics.
- **Operations**:
    1. Click the "Generate Theoretical Data" button. To customize the generation range, click the "Custom Generate Theoretical Data" button to modify the range and step size in the popup window.
    2. The system will generate theoretical images in the `./template` directory.
    3. Prediction data images will be generated in the `./actual_data` directory.
    4. During generation, progress will be displayed in the progress bar window or the system output box.
    5. Both the progress bar window and the function button area have a "Stop Generation" button to pause theoretical data generation at any time.

5.3 Train Model

- **Function**: Uses theoretical data to train the prediction model.
- **Operations**:
    1. Ensure theoretical data has been generated (images exist in the `./template` directory).
    2. Click the "Train Model" button. A selection window will pop up to choose the desired optimization method and clustering algorithm.
    3. The training process will run in the background; view progress in the progress bar window.
    4. After training completes, the model will be saved in the `./saved_models` directory, and training visualization data will be automatically displayed.
    5. Both the progress bar window and the function button area have a "Stop Training" button to pause the model training process at any time.

5.4 Load & Export Model

- **Function**: Load a pre-trained model.
- **Operations**:
    1. Click the "Load Model" button.
    2. Select the previously trained model directory `./saved_models`.
    3. Upon successful loading, the status bar will display "Loaded".
    4. Model files can be exported in Joblib and Pickle formats.

5.5 Model Comparison

- **Function**: Compare all existing models.
- **Operations**:
    1. Click the "Model Comparison" button.
    2. If comparison data already exists, the system will automatically display the comparison results and visualization images.
    3. If new models have been added, the system will automatically re-evaluate each model's performance and recommend the best model.

5.6 Import Experimental Data

The system provides two import methods:

5.6.1 Import Raw Data

- **Function**: Import raw experimentally measured data and generate images.
- **Operations**:
    1. Click the "Import Data 1 (Raw Data)" button.
    2. Select a text file (.txt format) containing incidence angles and deviation angles.
    3. The system will generate an interpolated curve and display it in the results area.

5.6.2 Import Data and Predict to 80 Degrees

- **Function**: Import data and predict/extend the curve up to an incidence angle of 80 degrees.
- **Operations**:
    1. Click the "Import Data 2 (Plot to 80°)" button.
    2. Select a text file containing incidence angles and deviation angles.
    3. The system will generate the curve extended to 80 degrees and display it.

5.7 Predict Refractive Index

- **Function**: Use the loaded model to predict the prism refractive index.
- **Operations**:
    1. Ensure a model is loaded and experimental data is imported.

2. Click the "Predict Refractive Index" button for a single data prediction.
3. Click the "Batch Predict" button to predict data from all files within a selected folder.
4. The system will display the prediction results (refractive index value and confidence level).
5. The results display area provides a function to save the prediction results to a specified location.

5.8 View Optimization History

- **Function**: View the hyperparameter optimization process during model training.
- **Operations**:
    1. Ensure a model is loaded.
    2. Click the "View Optimization History" button.
    3. The system will display the optimization history chart in the system's default web browser.

5.9 View Visualization Results

- **Function**: View visualization charts from the training process.
- **Operations**:
    1. Ensure a model is loaded.
    2. Click the "View Visualization Results" button.
    3. The system will display charts such as feature visualization and cluster distribution in the results area.
    4. If the clustering algorithm is SOM, click the button under the SOM tab to display SOM training-related visualization results in the system's default web browser.

5.10 View Prediction History

- **Function**: View historical prediction data.
- **Operations**:
    1. Click the "Prediction History" button or use the shortcut Ctrl+H.
    2. Historical prediction data will be displayed in a popup window.

5.11 View System Monitoring History

- **Function**: View historical system monitoring data.
- **Operations**:
    1. Click the menu bar item "View System Monitoring History".
    2. Historical system monitoring data will be displayed in a popup window.

5.12 System Monitoring

- **Function**: Monitor the current computer's hardware usage.
- **Operations**:
    1. Click the "System Monitoring" button.

2. The current computer hardware usage will be displayed under the "System Monitoring" tab in the system output box, updated every second.

5.13 Customize Shortcuts

- **Function**: Customize shortcuts for each function.
- **Operations**:
    1. Click the menu bar item "About - Customize Shortcuts".
    2. Modify function shortcuts in the popup window.

6. File Structure Description

```
Project Directory
├── template/ # Theoretical data images
├── actual_data/ # Prediction data images
├── All_Users/ # User data
├── saved_models/ # Saved trained models
├── history/ # Historical data
├── monitoring_logs/ # System monitoring logs
| └── run_20250101_123456/ # Model directory named by timestamp
|   ├── models/ # Model files
|   └── results/ # Training result charts
├── LibreHardwareMonitor/
| └── LibreHardwareMonitorLib.dll
├── resnet50/ # Pre-trained model
| └── resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5/
├── logs/ # System logs
├── prediction_results/ # Prediction result output
├── settings/ # Configuration folder
├── Sample_Dataset/ # Sample dataset folder
├── img/ # Images
| ├── icon.ico/ # System icon
| └── welcome.jpg/ # Welcome image
├── core/ # Core code package
| ├── gui_components/  # GUI components package
|   ├── __init__.py
|   ├── auto_updater.py
|   ├── batch_prediction.py
|   ├── data_import.py
|   ├── left_panel.py
|   ├── login_dialog.py
|   ├── menu.py
|   ├── model_comparison.py
|   ├── model_manager.py
|   ├── prediction_history.py
|   ├── progress_dialogs.py
|   ├── shortcut_manager.py
|   ├── system_monitor.py
|   ├── system_support.py
|   ├── training.py
|   └── welcome_screen.py
```

```
| ├── __init__.py
| ├── cluster_regressor.py
| ├── config.py
| ├── data_pipeline.py
| ├── feature_extractor.py
| ├── gui.py
| ├── model_trainer.py
| ├── predictor.py
| ├── prism_simulator.py
| ├── som.py
| ├── start_screen.py
| ├── user_manager.py
| ├── utils.py
| └── visualizer.py
├── Spectral_Refractive_Index_Prediction_System.py/ # Main program
├── requirements.bat/ # Python environment installer
├── requirements.txt/ # Python dependency list
├── numpy-1.25.1+mkl-cp39-cp39-win_amd64.whl/ # NumPy library offline installer
├── Sample_Incidence_Deviation_Angle_Data.txt/ # Sample data
└── README.pdf/ # User manual
```

7. Notes

1. The system automatically creates the necessary directory structure on first run; ensure the software has write permissions.
2. Theoretical data must be generated before training the model.
3. A model must be loaded and data imported before prediction.
4. Imported data must conform to a specific format; refer to the sample data.
5. The training process may take significant time (25 seconds - 5 minutes); please be patient.
6. Training and prediction processes will consume considerable CPU, GPU, and memory resources.
7. Do not install the software in paths containing Chinese characters or special characters.
8. Only administrators have access to all functions; other users have permission restrictions.

8. Shortcut List

Data Processing

Generate Theoretical Data: Ctrl+G
Custom Generate Theoretical Data: Ctrl+Shift+G
Stop Generation: Ctrl+Z
Data Augmentation: Ctrl+U

File Operations

Save Current System Output Content: Ctrl+S
Import Data 1 (Raw Data): Ctrl+O
Import Data 2 (Plot to 80°): Ctrl+Shift+O
Exit: Ctrl+Q

Model Operations

Train Model: Ctrl+T
Stop Training: Ctrl+X
Load Model: Ctrl+L
Export Model: Ctrl+E
Model Management: Ctrl+M

Prediction Analysis

Predict Refractive Index: Ctrl+P
Batch Predict: Ctrl+B

View Analysis

View Optimization History: Ctrl+O (Note: Conflict with Import Data 1 shortcut might exist based on description)
View Visualization Results: Ctrl+V
Model Comparison: Ctrl+Shift+M

History

View Prediction History: Ctrl+H
View Monitoring Logs: Ctrl+N

System Tools

System Monitoring: Ctrl+Y
Refresh Interface: F5
Clear Charts: Ctrl+D
Clear Output: Ctrl+Shift+D
User Management: Ctrl+Shift+P

Help

Toggle Theme: Alt+D
User Guide: F1
Shortcut List: Ctrl+K
Customize Shortcuts: Ctrl+Shift+K
About: Ctrl+A

9. Common Issue Resolution

Issue 1: "Insufficient valid samples" during training

- **Cause**: Not enough images in the `./template` directory.
- **Solution**:
    1. Click the "Generate Theoretical Data" button.
    2. Ensure there are enough images in the `./template` directory (at least 10).

Issue 2: Inaccurate prediction results

- **Cause**: Insufficient training data for the model.
- **Solution**:
    1. Generate more theoretical data (customize the refractive index range if needed).
    2. Increase the number of training samples by clicking the "Data Augmentation" button.
    3. Retrain the model.

Issue 3: Training process lags or crashes

- **Solution**:
    1. Close other resource-intensive programs.
    2. Reduce training parameters (decrease the number of trials).
    3. Check if system memory is sufficient.

Issue 4: Unable to save results

- **Solution**:
    1. Check if disk space is sufficient.
    2. Confirm the software has write permission for the directory.
    3. Check if antivirus software is blocking file writes.

10. Technical Support

If you encounter unresolved issues, please contact technical support:

- Email: 3298700189@qq.com
- Development Team Members: Wu Xun, Xu Yitian, Zhao Zihan
- Development Team Institution: Zhejiang University of Technology
- Version: 3.1.0 (2025)

11. Exiting the System

Normal Exit

1. Click the close button in the upper right corner of the window or use the shortcut Ctrl+Q.
2. The system will automatically save the current state and exit safely.

Force Quit

If the program becomes unresponsive, end the process via Task Manager.

> Tips:
>
> 1. Regularly back up important models and prediction results.
> 2. The system automatically logs operations for troubleshooting and analysis.

> 3. Read the function descriptions in detail before use to fully utilize the software's capabilities.
> 4. For processing large amounts of data, it is recommended to run on a computer with better performance.
> 5. This software is for learning and research purposes only. Do not use it for illegal purposes.
> 6. The default administrator account is `admin` with the password `admin123`.

Version: V3.1.0
Release Date: 2025

中文简介

OptiSVR分光计折射率预测系统

0. 更新说明

该版本基本解决了训练时可能会出现系统假死闪退的Bug，但并不排除还会出现的可能性；同时改进了进度条的显示，使之更加朴素简明。

1. 软件概述

本系统是一个基于人工智能算法的分光计折射率预测系统，采用多种无监督学习聚类(K-Means & SOM)和支持向量回归(SVR)算法，同时结合深度学习ResNet50网络模型对入射角-偏向角图像进行特征提取，通过多次训练调整超参数得到最佳权重保存最佳预训练模型。

预测思路是将测定的数据插值后转为入射角-偏向角图像，对图片曲线进行预测，能够准确预测分光计的折射率特性。

2. 系统要求

- 操作系统：Windows 10/11
- 推荐使用 Python 3.9.6，并安装必要的Python库（见安装指南）
- 推荐硬件配置：CPU 16核+，内存16GB+

3. 安装指南

1. 安装Python 3.9.6（从Python官网下载）
2. 安装依赖库：运行 requirements. bat 文件， requirements. txt 文件里有详细的依赖库列表

4. 软件启动

运行主程序文件Spectral_Refractive_Index_Prediction_System.py，系统将显示欢迎界面，预计5-8秒后进入主界面。首次运行时，系统会自动初始化相关目录和配置文件。

5. 主要功能

5.1 用户系统

- **功能**：新增的用户系统，能处理用户的各种操作，如用户注册、登录、重置密码、修改密码、访客登录、编辑资料、权限说明、权限管理、用户管理、退出登录等。
- **操作**：
    1. 启动软件后，会先弹出用户登录窗口
    2. 用户可在右上角点击头像，出现悬浮框即可进行编辑资料、查看权限说明、提升权限、退出登录操作。其中默认管理员账号是admin，密码是admin123。
    3. 只有管理员才能对其他用户的账号、权限进行操作

5.2 生成理论数据

- **功能**：根据分光计物理特性生成理论折射率图像
- **操作**：
    1. 点击"生成理论数据"按钮。如果想自定义理论数据生成范围，可以点击"自定义生成理论数据"按钮，将在弹窗里修改生成范围及步长
    2. 系统将在`./template`目录下生成理论图像
    3. 在`./actual_data`目录下生成预测数据图像
    4. 在生成过程中，将会在进度条窗口或者系统输出框显示进度
    5. 进度条窗口和功能按钮区都有“停止生成”按钮，可随时暂停生成理论数据

5.3 训练模型

- **功能**：使用理论数据训练预测模型
- **操作**：
    1. 确保已生成理论数据（`./template`目录中有图像）
    2. 点击"训练模型"按钮，将会有选择窗口弹出，可选择想要优化的方式和聚类的算法
    3. 训练过程将在后台进行，在进度条窗口查看进度
    4. 训练完成后，模型将保存在`./saved_models`目录下，并自动显示训练可视化的数据
    5. 进度条窗口和功能按钮区都有“停止训练”按钮，可随时暂停模型训练过程

5.4 加载与导出模型

- **功能**：加载预训练的模型
- **操作**：
    1. 点击"加载模型"按钮
    2. 选择之前训练好的模型目录`./saved_models`
    3. 加载成功后，状态栏将显示"已加载"
    4. 可导出 Joblib 和 Pickle 类型的模型文件

5.5 模型比较

- **功能**：比较所有存在的模型
- **操作**：
    1. 点击"模型比较"按钮
    2. 如果已有比较数据，系统将自动显示比较结果及可视化图片
    3. 如果有新增的模型，系统将自动重新评估每个模型的性能，并给出最佳推荐模型

5.6 导入实验数据

系统提供两种导入方式：

5.6.1 导入原始数据

- **功能**：导入实验测量的原始数据并生成图像
- **操作**：
    1. 点击"导入数据1(原始数据)"按钮
    2. 选择包含入射角和偏向角的文本文件（.txt格式）
    3. 系统将生成插值曲线并显示在结果区域

5.6.2 导入数据并预测至80度

- **功能**：导入数据并预测至入射角80度
- **操作**：
    1. 点击"导入数据2(绘图到80度)"按钮
    2. 选择包含入射角和偏向角的文本文件
    3. 系统将生成扩展至80度的曲线并显示

5.7 预测折射率

- **功能**：使用加载的模型预测棱镜折射率
- **操作**：
    1. 确保已加载模型并导入实验数据
    2. 点击"预测折射率"按钮，将会进行单次数据的预测
    3. 点击“批量预测”按钮，将会对一个文件夹里的所有数据文件里的数据进行预测
    4. 系统将显示预测结果（折射率值和置信度）
    5. 结果展示区提供保存预测结果功能，可将单次预测的结果保存到指定位置

5.8 查看优化历史

- **功能**：查看模型训练时的超参数优化过程
- **操作**：
    1. 确保已加载模型
    2. 点击"查看优化历史"按钮

3. 系统将在系统默认浏览器中显示优化历史图表

5.9 查看可视化结果

- **功能**：查看训练过程中的可视化图表
- **操作**：
    1. 确保已加载模型
    2. 点击"查看可视化结果"按钮
    3. 系统将在结果区域显示特征可视化、聚类分布等图表
    4. 如果聚类算法是SOM算法，可以点击SOM标签页下的按钮，在系统默认浏览器中显示SOM训练的相关可视化结果

5.10 查看预测历史

- **功能**：查看历史预测数据
- **操作**：
    1. 点击"预测历史"按钮或者快捷键Ctrl+H
    2. 将在弹窗里显示历史预测数据

5.11 查看系统监控历史

- **功能**：查看历史系统监控数据
- **操作**：
    1. 点击菜单栏“查看系统监控历史”
    2. 将在弹窗里显示系统监控历史数据

5.12 系统监控

- **功能**：对当前电脑硬件使用情况进行监控
- **操作**：
    1. 点击"系统监控"按钮
    2. 将在系统输出框的“系统监控”标签页下显示当前电脑硬件使用情况，每隔秒更新一次数据

5.13 自定义快捷键

- **功能**：自定义每个功能的快捷键
- **操作**：
    1. 点击菜单栏"关于-自定义快捷键"项
    2. 可在弹窗里修改功能的快捷键

6. 文件结构说明

```
项目目录
├── template/ # 理论数据图像
├── actual_data/ # 预测数据图像
```

```
├── All_Users/ # 用户数据
├── saved_models/ # 保存的训练模型
├── history/ # 历史数据
├── monitoring_logs/ # 系统监控日志
| └── run_20250101_123456/ # 按时间戳命名的模型目录
|   ├── models/ # 模型文件
|   └── results/ # 训练结果图表
├── LibreHardwareMonitor/
| └── LibreHardwareMonitorLib.dll
├── resnet50/ # 预训练模型
| └── resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5/
├── logs/ # 系统日志
├── prediction_results/ # 预测结果输出
├── settings/ # 配置文件夹
├── 数据集示例/ # 数据集示例文件夹
├── img/ # 图片
| ├── icon.ico/ # 系统图标
| └── welcome.jpg/ # 欢迎图片
├── core/ # 核心代码包
| ├── gui_components/  # GUI组件包
|   ├── __init__.py
|   ├── auto_updater.py
|   ├── batch_prediction.py
|   ├── data_import.py
|   ├── left_panel.py
|   ├── login_dialog.py
|   ├── menu.py
|   ├── model_comparison.py
|   ├── model_manager.py
|   ├── prediction_history.py
|   ├── progress_dialogs.py
|   ├── shortcut_manager.py
|   ├── system_monitor.py
|   ├── system_support.py
|   ├── training.py
|   └── welcome_screen.py
| ├── __init__.py
| ├── cluster_regressor.py
| ├── config.py
| ├── data_pipeline.py
| ├── feature_extractor.py
| ├── gui.py
| ├── model_trainer.py
| ├── predictor.py
| ├── prism_simulator.py
| ├── som.py
| ├── start_screen.py
| ├── user_manager.py
| ├── utils.py
| └── visualizer.py
├── Spectral_Refractive_Index_Prediction_System.py/ # 主程序
├── requirements.bat/ # Python 环境安装程序
├── requirements.txt/ # Python 依赖库列表
```

```
├── numpy-1.25.1+mkl-cp39-cp39-win_amd64.whl/ # numpy库离线安装包
├── 入射角和偏向角数据示例.txt/ # 示例数据
└── README.pdf/ # 使用说明文件
```

7. 注意事项

1. 首次运行时系统会自动创建必要的目录结构，请确保软件有写入权限
2. 训练模型前需先生成理论数据
3. 预测前需先加载模型并导入数据
4. 导入的数据需要符合特定格式，建议参考示例数据
5. 训练过程可能较长时间（25秒-5分钟），请耐心等待
6. 训练和预测过程会占用较多CPU、GPU和内存资源
7. 请勿将软件安装在包含中文或特殊字符的路径中
8. 只有管理员才能使用所有功能，其他用户都有权限限制

8. 快捷键列表

数据处理

生成理论数据	Ctrl+G
自定义生成理论数据	Ctrl+Shift+G
停止生成	Ctrl+Z
数据增强	Ctrl+U

文件操作

保存当前系统输出内容	Ctrl+S
导入数据1(原始数据)	Ctrl+O
导入数据2(绘图到80度)	Ctrl+Shift+O
退出	Ctrl+Q

模型操作

训练模型	Ctrl+T
停止训练	Ctrl+X
加载模型	Ctrl+L
导出模型	Ctrl+E
模型管理	Ctrl+M

预测分析

预测折射率	Ctrl+P
批量预测	Ctrl+B

查看分析

查看优化历史	Ctrl+O
查看可视化结果	Ctrl+V
模型比较	Ctrl+Shift+M

历史记录

查看预测历史	Ctrl+H
查看监控日志	Ctrl+N

系统工具

系统监控	Ctrl+Y
刷新界面	F5
清空图表	Ctrl+D
清空输出	Ctrl+Shift+D
用户管理	Ctrl+Shift+P

帮助

切换主题	Alt+D
使用指南	F1
快捷键列表	Ctrl+K
自定义快捷键	Ctrl+Shift+K
关于	Ctrl+A

9. 常见问题处理

问题1：训练时提示"有效样本不足"

- **原因**：`./template`目录中没有足够图像
- **解决**：
    1. 点击"生成理论数据"按钮
    2. 确保`./template`目录中有足够图像（至少10张）

问题2：预测结果不准确

- **原因**：模型训练数据不足
- **解决**：
    1. 生成更多理论数据（可自定义折射率范围）
    2. 增加训练样本数量，点击“数据增强”按钮
    3. 重新训练模型

问题3. 训练过程卡顿或崩溃

- **解决**：
    1. 关闭其他占用资源的程序
    2. 降低训练参数（减少试验次数）
    3. 检查系统内存是否充足

问题4. 无法保存结果

- **解决**：
    1. 检查磁盘空间是否充足
    2. 确认软件具有目录写入权限
    3. 检查防病毒软件是否阻止文件写入

10. 技术支持

如遇到无法解决的问题，请联系技术支持：

- 邮箱：3298700189@qq.com
- 开发团队成员：吴迅　徐一田　赵子涵
- 开发团队单位：浙江工业大学

- 版本：3.1.0 (2025)

11. 退出系统

正常退出

1. 点击窗口右上角关闭按钮或使用快捷键Ctrl+Q
2. 系统会自动保存当前状态并安全退出

强制退出

如遇程序无响应，可通过任务管理器结束进程

> 温馨提示：
>
> 1. 建议定期备份重要的模型和预测结果
> 2. 系统会自动记录操作日志，便于问题追踪和分析
> 3. 使用前建议详细阅读各功能说明，以便充分发挥软件效能
> 4. 如需处理大量数据，建议在性能较好的计算机上运行
> 5. 本软件仅供学习和研究使用，请勿用于非法用途
> 6. 默认管理员账号是admin，密码是admin123。

版本：V3.1.0
发布日期：2025年