OptiSVR Spectral Refractive Index Prediction System

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

0. Update Instructions

- The graphical framework has been upgraded to the PyQT5 architecture, better suited for large-scale project requirements.
- Opening web page files no longer relies on the Edge browser; instead, the system default browser is used.
- Added a page refresh function.
- Further beautified the graphical interface for a more modern look.

1. Software Overview

This software is an AI-based spectrometer refractive index prediction system. It adopts a transformer architecture and uses a Support Vector Regression (SVR) algorithm and a clustering regression model. Through multiple training sessions and hyperparameter adjustments, the best weights are obtained, and the optimal pre-trained model is saved to achieve the goal of predicting the spectrometer's refractive index.

The prediction approach involves interpolating the measured data to convert it into an incident angle-deviation angle image, and then predicting the image curve. This method can accurately predict the refractive index characteristics of the prism.

2. System Requirements

- Operating System: Windows 10/11
- Recommended Python Version: 3.9.6
- Installation of necessary Python libraries (see Installation Guide)
- Recommended Hardware Configuration: 16-core CPU, 16GB RAM

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 a detailed list of dependency 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 approximately 5-10 seconds.

5. Main Functions

5.1 Generate Theoretical Data

- **Function**: Generates theoretical refractive index images based on the physical characteristics of the prism.
- **Operation**:
    1. Click the "Generate Theoretical Data" button.
    2. The system will generate theoretical images in the `./template` directory.
    3. Experimental data images will be generated in the `./actual_data` directory.

5.2 Train Model

- **Function**: Uses theoretical data to train the prediction model.
- **Operation**:
    1. Ensure theoretical data has been generated (images exist in the `./template` directory).
    2. Click the "Train Model" button.
    3. The training process will run in the background (progress can be viewed in the output window).
    4. After training is completed, the model will be saved in the `saved_models` directory.

5.3 Load Model

- **Function**: Loads a pre-trained model.
- **Operation**:
    1. Click the "Load Model" button.
    2. Select the directory of the previously trained model.
    3. Upon successful loading, the status bar will display "Loaded".

5.4 Import Experimental Data

The system provides two import methods:

5.4.1 Import Raw Data

- **Function**: Imports raw data from experimental measurements and generates images.
- **Operation**:
    1. Click the "Import Data 1 (Raw Data)" button.
    2. Select a text file (.txt format) containing incident angles and deviation angles.
    3. The system will generate an interpolated curve and display it in the result area.

5.4.2 Import Data and Predict up to 80 Degrees

- **Function**: Imports data and predicts up to an incident angle of 80 degrees.
- **Operation**:
    1. Click the "Import Data 2 (Plot up to 80 Degrees)" button.
    2. Select a text file containing incident angles and deviation angles.
    3. The system will generate a curve extended to 80 degrees and display it.

5.5 Predict Refractive Index

- **Function**: Uses the loaded model to predict the prism's refractive index.
- **Operation**:
    1. Ensure the model is loaded and experimental data is imported.
    2. Click the "Predict Refractive Index" button.
    3. The system will display the prediction result (refractive index value and confidence level).

5.6 View Optimization History

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

5.7 View Visualization Results

- **Function**: Views visualization charts generated during the training process.
- **Operation**:
    1. Ensure a model is loaded.
    2. Click the "View Visualization Results" button.
    3. The system will display charts such as feature visualizations and cluster distributions in the result area.

6. File Structure Description

```
Project Directory
├── template/          # Theoretical data images
├── actual_data/       # Experimental data images
├── saved_models/      # Saved trained models
|   └── run_20250101_123456/  # Model directory named by timestamp
|       ├── models/           # Model files
|       └── results/          # Training result charts
├── resnet50/          # Pre-trained model
|   └── resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5/ # ResNet50 model file
├── tables/             # Experimental data tables (CSV format)
```

```
├── logs/               # System logs
├── img/                # System icons and welcome image
├── Spectral_Refractive_Index_Prediction_System.pyw/  # Main program
├── requirements.bat/   # Environment setup script
└── requirements.txt/   # Environment list
├── Incident_Angle_and_Deviation_Angle_Data.txt/ # Example data
└── readme.pdf/         # User manual file
```

7. Notes

1. Theoretical data must be generated before training the model.
2. A model must be loaded and experimental data imported before prediction.
3. The system records detailed logs in the `logs/` directory.
4. The training process may take a relatively long time (2-5 minutes); please wait patiently.

8. Common Issue Handling

Issue 1: "Insufficient valid samples" prompt 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 (modify the refractive index range in the code).
    2. Increase the number of training samples.
    3. Retrain the model.

9. Technical Support

- Development Team: Wu Xun, Xu Yitian
- Development Team Affiliation: Zhejiang University of Technology
- Version: 1.2 (2025)

10. Exiting the System

Click "File" → "Exit" in the menu bar or simply close the window to exit the system.

> Note: This system uses advanced artificial intelligence technology. Prediction results are for reference only. Please combine with physical experiments for verification in practical applications.
>
> The folder already contains some pre-existing training weights.

中文简介

OptiSVR分光计折射率预测系统

0. 更新须知

- 图形化框架升级为 PyQT5 架构，更匹配大型项目需求
- 打开网页文件不再依赖Edge浏览器，而是使用系统默认浏览器
- 新增刷新页面功能
- 进一步美化图形界面，更加现代化

1. 软件概述

本软件是一个基于人工智能的分光计折射率预测系统，采用transformer架构，使用支持向量回归(SVR)算法和分簇回归模型，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，以达到预测分光计折射率的目的。

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，能够准确预测棱镜的折射率特性。

2. 系统要求

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

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5-10秒后进入主界面。

5. 主要功能

5.1 生成理论数据

- **功能**：根据棱镜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
    2. 系统将在`./template`目录下生成理论图像
    3. 在`./actual_data`目录下生成实验数据图像

5.2 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
    3. 训练过程将在后台进行（可在输出窗口查看进度）
    4. 训练完成后，模型将保存在`saved_models`目录下

5.3 加载模型

- **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
    3. 加载成功后，状态栏将显示"已加载"

5.4 导入实验数据

系统提供两种导入方式：

5.4.1 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

5.4.2 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

5.5 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
    3. 系统将显示预测结果（折射率值和置信度）

5.6 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. 系统将在默认浏览器中显示优化历史图表

5.7 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

6. 文件结构说明

```
项目目录
├── template/           # 理论数据图像
├── actual_data/        # 实验数据图像
├── saved_models/       # 保存的训练模型
|   └── 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
├── resnet50/           # 预训练模型
|   └── resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5/ # resnet50模型文件
├── tables/              # 实验数据表（CSV格式）
├── logs/                # 系统日志
├── img/                 # 系统图标和欢迎图片
├── Spectral_Refractive_Index_Prediction_System.pyw/  # 主程序
├── requirements.bat/    # 环境安装程序
└── requirements.txt/    # 环境列表
├── 入射角和偏向角数据.txt/# 示例数据
└── readme.pdf/          # 使用说明文件
```

7. 注意事项

1. 训练模型前需先生成理论数据
2. 预测前需先加载模型并导入实验数据
3. 系统会记录详细日志在`logs/`目录下
4. 训练过程可能较长时间（2-5分钟），请耐心等待

8. 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（修改代码中的折射率范围）
    2. 增加训练样本数量
    3. 重新训练模型

9. 技术支持

- 开发团队名单：吴迅　徐一田
- 开发团队单位：浙江工业大学
- 版本：1.2 (2025)

10. 退出系统

点击菜单栏"文件"→"退出"或直接关闭窗口即可退出系统。

> 提示：本系统使用先进的人工智能技术，预测结果仅供参考，实际应用时请结合物理实验验证。
>
> 文件夹中已经包含了一部分原有的训练权重。