Exoplanet Spectral Analysis using CNN

Abstract

This project implements a **machine learning system** for analyzing transmission spectra of exoplanet atmospheres to detect and measure concentrations of **water (H_2O)** and **oxygen (O_2)**. Synthetic data was generated to simulate real atmospheric spectra, and a **Convolutional Neural Network (CNN)** was trained to accurately predict gas concentrations. The system automates exoplanet atmospheric analysis — a key step toward identifying potentially habitable worlds.

Introduction

Problem Statement

Develop an automated deep learning model to detect and quantify **H_2O** and **O_2** in exoplanet atmospheres using **transmission spectroscopy** data. The model must handle spectral complexity, noise, and atmospheric effects efficiently.

Objectives

- Build a reliable tool to identify and characterize exoplanets with potential **biosignatures**
- Automate and accelerate spectroscopic data analysis
- Improve prediction accuracy of molecular concentrations

Motivation

- Identifying potentially habitable planets
- Detecting biosignature gases (H_2O + O_2)
- Reducing manual data analysis time
- Enabling faster telescope-based follow-up studies

Data Generation

Synthetic transmission spectra were generated for exoplanet atmospheres containing **H_2O, O_2, CO_2, CH_4, and N_2**.

Steps:

1. **Absorption Features** – Gaussian or Voigt profiles simulate molecular absorption bands:

- H_2O → 1.4, 1.9, 2.7 µm
- O_2 → 0.69, 0.76, 1.27 µm
- CO_2, CH_4, N_2 with unique wavelengths

2. **Noise Addition** – Gaussian noise simulates realistic observation errors
3. **Concentration Ranges:**
   - H_2O: 0–10%
   - O_2: 0–25%
4. **Dataset:**
   - ~10,000 spectra
   - 1000 wavelength points per sample (0.5–3.0 µm)
   - Labels: H2O_concentration, O2_concentration

Data Preprocessing

- **Standardization:** Normalize spectra to mean 0 and std 1
- **Noise Reduction:** Apply **Savitzky–Golay filter** to smooth spectra
- **Dimensionality Reduction (Optional):** PCA to capture 95% variance
- **Train-Test Split:** 80% training, 20% validation

Methodology

1. Model Used: Convolutional Neural Network (CNN)

- **3 Convolutional Layers** + **Max Pooling**
- **Fully Connected Layers** + **Dropout** (to prevent overfitting)
- **Output:** Continuous values for H_2O and O_2 concentrations

2. Why CNN?

- Detects **local spectral patterns** like absorption peaks
- Learns **hierarchical features** automatically
- Handles **noise and high-dimensional data** effectively

3. Training Setup

- **Optimizer:** Adam
- **Loss Function:** Mean Squared Error (MSE)
- **Batch Size:** 32
- **Epochs:** 50
- **Validation Split:** 80/20
- **Framework:** PyTorch

4. Feature Processing

- **Standard Scaling** → equal feature weightage
- **Savitzky–Golay Filter** → denoising without losing peaks
- CNN implicitly performs **feature selection and extraction**

Experimental Setup

Tools & Libraries

- **Python 3.8+**
- **PyTorch** – model building & training
- **NumPy, Pandas** – data handling
- **SciPy** – spectral simulation (Voigt profiles, filters)
- **Scikit-learn** – scaling, train/test split
- **Matplotlib** – plotting loss curves & evaluation

Environment

- CPU: Multicore (recommended 16GB+ RAM)
- GPU (optional): CUDA-enabled for faster training
- Platform: Google Colab or Local Python Environment

Evaluation

Metrics

- **Mean Squared Error (MSE):** Penalizes large errors
- **Mean Absolute Error (MAE):** Measures average deviation

Formulas:

$$MSE = \frac{1}{n}\sum(y_{true} - y_{pred})^2$$
$$MAE = \frac{1}{n}\sum \mid y_{true} - y_{pred} \mid$$

Visual Evaluation

- Scatter plots: True vs Predicted concentrations for H_2O & O_2
- Training & validation loss curves
- Optional metrics: RMSE, R^2 score

Results & Discussion

- CNN achieved **low MSE and MAE**, showing strong predictive accuracy
- Outperformed traditional regressors (Linear Regression, SVR, Random Forest)
- Errors increased for extreme absorption spectra — can be reduced via tuning or augmentation
- Visualization confirmed close alignment between predicted and actual gas concentrations

---

Error Analysis

Observed Issues

- **Overfitting:** Model memorizing training noise
- **Underfitting:** Insufficient layers for complex spectra
- **High Error Cases:** Extreme concentrations caused deviation

Improvements

- Add **dropout** & **weight decay**
- Perform **hyperparameter optimization** (learning rate, depth)
- Introduce **data augmentation** and **early stopping**

Conclusion

This project demonstrates that a **CNN-based deep learning model** can effectively analyze spectral data to predict **H_2O and O_2** concentrations in exoplanet atmospheres.
The approach automates the interpretation of complex transmission spectra, making it a valuable tool for **exoplanetary research** and **biosignature detection**.
Future work includes expanding molecular diversity, fine-tuning hyperparameters, and adapting the model for real telescope data.

How to Run

```
# Install dependencies

pip install -r requirements.txt


# Run the training script

python train_model.py


# Evaluate the model

python evaluate_model.py
```

📁 **Tech Stack**

Languages: Python
Frameworks: PyTorch, SciPy, Scikit-learn
Environment: Google Colab
Visualization: Matplotlib