Surrogate Models for CNN Interpretability - Assignment

🎯 Objective

The goal of this assignment is to build **interpretable surrogate models** that can approximate the predictions of a trained Convolutional Neural Network (CNN) for glaucoma detection using **hand-crafted morphological features** extracted from retinal images.

🤔 What is a Surrogate Model?

A **surrogate model** (also called a **proxy model** or **substitute model**) is a simpler, more interpretable model that approximates the behavior of a complex "black-box" model.

Why do we need them?

The Problem with CNNs:

- CNNs achieve high accuracy but are **black boxes**
- Hard to explain **why** a prediction was made
- Difficult to trust in clinical settings
- Cannot easily identify which features matter most

The Solution - Surrogate Models:

- Use interpretable models (Decision Trees, Linear Models, etc.)
- Train them to **mimic the CNN's predictions**
- Gain insights into what the CNN has learned
- Provide explanations to clinicians

📊 The Task

You will work with two types of data:

1. **Morphological Features** (Input)

Hand-crafted numerical features extracted from retinal fundus images, such as:

- Cup-to-disc ratio (CDR)
- Rim area measurements
- Blood vessel characteristics
- Optic disc parameters
- Neuroretinal rim features
- And others...

These features are stored in CSV files with the following structure:

```
filename,feature_1,feature_2,feature_3,...,feature_N
image001.jpg,0.45,123.4,0.89,...,45.2
image002.jpg,0.23,98.7,0.76,...,38.1
...
```

2. **CNN Predictions** (Target)

Probabilities of glaucoma predicted by a trained CNN model:

```
filename,glaucoma_probability
image001.jpg,0.87
image002.jpg,0.12
...
```

Your task: Build a model that learns to predict the CNN's `glaucoma_probability` using only the morphological features.

🎓 Key Concepts

Regression vs Classification

Important: This is a **regression problem**, not classification!

- **Input:** Morphological features (continuous values)
- **Output:** Probability (continuous value between 0 and 1)
- **Goal:** Predict the exact probability the CNN would give

You are NOT predicting the class (normal/glaucoma), but rather **how confident the CNN would be** that an image shows glaucoma.

What makes this different?

Traditional Task	This Assignment
Predict: Normal or Glaucoma	Predict: CNN's probability score
Target: Ground truth labels	Target: CNN predictions
Goal: Maximize accuracy	Goal: Minimize prediction error
Metric: Accuracy, F1-Score	Metric: MSE, MAE, R^2

🔍 Why is this Useful?

1. **Interpretability**

- Understand which morphological features the CNN relies on

- See decision paths in tree-based models
- Explain predictions to medical professionals

2. **Feature Importance**

- Discover which clinical measurements matter most
- Validate if the CNN focuses on medically relevant features
- Identify potential biases or shortcuts

3. **Model Validation**

- Check if the CNN learned clinically meaningful patterns
- Compare CNN's implicit features with expert knowledge
- Build trust in the automated system

4. **Efficiency**

- Surrogate models are faster than CNNs
- Require less computational resources
- Easier to deploy in resource-constrained settings

5. **Clinical Acceptance**

- Doctors can understand decision tree rules
- Provides transparency for medical decisions
- Meets regulatory requirements for explainability

Dataset Structure

You already have the morphological features dataset. The CNN predictions dataset has to be obtained with the predictions of your best model in the CNN glaucoma/normal classification problem:

Training Data

- `train_features.csv` - Morphological features for training images
- `cnn_predictions_train.csv` - CNN probability predictions for training images

Test Data

- `test_features.csv` - Morphological features for test images
- `cnn_predictions_test.csv` - CNN probability predictions for test images

Merging the Data

Both datasets can be merged using the `filename` column:

```
import pandas as pd

# Load data
```

```
features_df = pd.read_csv('train_features.csv')
cnn_df = pd.read_csv('cnn_predictions_train.csv')

# Merge on filename
data = features_df.merge(cnn_df, on='filename')

# Now you have features and CNN probabilities together
X = data.drop(['filename', 'glaucoma_probability'], axis=1)
y = data['glaucoma_probability']
```

🎯 Assignment Goals

Primary Goal

Build a surrogate model that accurately predicts the CNN's probability outputs using morphological features.

Questions to Answer

1. **Model Performance:**
    - How well can interpretable models approximate the CNN?
    - What is the prediction error (MSE, MAE)?
    - How much variance can be explained (R^2)?
2. **Feature Analysis:**
    - Which morphological features are most important?
    - Do they align with clinical knowledge about glaucoma?
    - Are there unexpected feature dependencies?
3. **Interpretability:**
    - Can you explain the surrogate model's decisions?
    - What rules or patterns does it learn?
    - How do predictions change with different feature values?
4. **Clinical Insights:**
    - Does the CNN focus on medically relevant features?
    - Are there any surprising findings?
    - Could the surrogate model be used in practice?

🛠️ Suggested Approaches

While you're free to explore any interpretable ML technique, here are some suggestions:

1. **Decision Trees**

Pros:

- Highly interpretable (visual tree structure)
- Can capture non-linear relationships
- No feature scaling needed
- Easy to understand rules

Cons:

- Can overfit if too deep
- May not capture complex interactions

Parameters to explore:

- `max_depth`: How deep can the tree grow?
- `min_samples_split`: Minimum samples to split a node
- `min_samples_leaf`: Minimum samples in leaf nodes

---

2. **Random Forests**

Pros:

- Better performance than single trees
- Built-in feature importance
- Reduces overfitting

Cons:

- Less interpretable than single tree
- Harder to visualize

---

3. **Linear Regression**

Pros:

- Maximum interpretability (coefficients = feature weights)
- Fast and simple
- No hyperparameters to tune

Cons:

- Assumes linear relationships
- May not capture complexity

---

4. **Polynomial Regression**

Pros:

- Captures non-linear patterns
- Still interpretable with low degree
- Extension of linear regression

Cons:

- Can overfit with high degrees
- More complex to interpret

📊 Evaluation Metrics

Your surrogate model will be evaluated using **regression metrics**:

Primary Metrics

1. **Mean Squared Error (MSE)**
    - Average squared difference between predicted and actual probabilities
    - Lower is better
    - Penalizes large errors heavily
2. **Root Mean Squared Error (RMSE)**
    - Square root of MSE
    - Same units as predictions (probabilities)
    - Easier to interpret than MSE
3. **Mean Absolute Error (MAE)**
    - Average absolute difference
    - More robust to outliers than MSE
    - Direct interpretation: average error in probability
4. **R^2 Score (Coefficient of Determination)**
    - Proportion of variance explained
    - Range: -∞ to 1 (1 = perfect, 0 = baseline)
    - Tells you how well your model fits the data

Example Interpretation

If your model achieves:

- **MAE = 0.05**: On average, predictions are off by 5%
- **RMSE = 0.08**: Typical error is about 8%
- **R^2 = 0.85**: Model explains 85% of the variance

🔬 Analysis Components

Your analysis can include:

1. **Model Performance**

- Training and test metrics

- Comparison of different models
- Learning curves (if applicable)
- Error distribution analysis

2. **Feature Importance**

- Which features matter most?
- Feature importance plots
- Correlation analysis

3. **Model Interpretation**

- Decision tree visualization (if using trees)
- Feature coefficients (if using linear models)
- Example predictions with explanations
- Cases where the model fails

4. **Comparison with CNN**

- How close is the surrogate to the CNN?
- Where does it agree/disagree?
- Residual analysis
- Worst predictions analysis

5. **Clinical Insights**

- Do learned features make medical sense?
- Comparison with known glaucoma indicators
- Unexpected patterns or findings
- Practical implications

💡 Tips for Success

Model Selection

- Start simple (linear regression, shallow tree)
- Gradually increase complexity
- Use cross-validation to avoid overfitting
- Compare multiple approaches

Hyperparameter Tuning

- Use grid search or random search
- Don't overfit to validation set
- Document your choices

Interpretation

- Don't just report numbers

- Explain what they mean clinically
- Connect findings to medical knowledge
- Be critical of your results

🤔 Discussion Questions

Consider these questions in your analysis:

1. **Fidelity:** How well does your surrogate approximate the CNN? Is the error acceptable for practical use?
2. **Complexity vs Interpretability:** Is there a trade-off? Do simpler models sacrifice too much accuracy?
3. **Feature Insights:** Which morphological features are most predictive? Do they align with clinical knowledge?
4. **CNN Validation:** Does the CNN appear to learn medically meaningful features, or does it rely on spurious correlations?
5. **Practical Utility:** Could this surrogate model be used in clinical practice? What are the advantages and limitations?
6. **Trust and Transparency:** Does the surrogate model make the CNN more trustworthy? Why or why not?