Enhancing CNN Interpretability
Using Multi-Layer Feature Extraction with Decision Trees

Chetan Maviti, Kanishk Sivanandam

2/2/2025

Abstract

Convolutional Neural Networks (CNNs) are revolutionary implementations of neural networks used in image classification, though they sacrifice interpretability for accuracy, functioning as black-box models. Common machine learning and computer vision applications such as medicine require both accuracy and justification in the decision-making process, limiting learning methods including CNNs. Previous work combating these issues tend towards a combination of decision trees and neural networks, often forgoing accuracy in the process. This paper presents a novel approach to the aforementioned approach, enhancing CNN interpretability by combining multi-layer features from a pre-trained ResNet50 with a Decision Tree (DT) trained on the extracted features. Unlike existing approaches, this method maintains valuable intermediary data found in CNNs. To address the high dimensionality of CNN feature maps, we also test Principal Component Analysis (PCA) for dimensionality reduction before training the DT. Our results on the Pneumonia Chest X-ray dataset demonstrate that multi-layer feature extraction for CNN + DT achieves competitive transparency with improved accuracy, while PCA accelerates training without significant information loss.

Introduction

Convolutional Neural Networks (CNNs) have become the go-to method for image classification, often matching or even surpassing human accuracy. However, CNNs typically function as "black boxes," giving us a prediction without explaining *why* they made that decision. This lack of transparency is a major concern in medical imaging, where doctors need to understand *how* a model reaches its conclusions, else a misdiagnosis or unexplained outcome with serious consequences may occur.

To address this issue, we combine a pre-trained CNN, *ResNet50* [1], with a Decision Tree (for interpretability of the model). *ResNet50* is a publicly open CNN consisting of 50 layers and trained on millions of images from the ImageNet dataset. We extract features from multiple different layers of the CNN—specifically Conv2, Conv4, and the Fully Connected (FC) layer—so we capture both basic edge information *and* more complex patterns. Since these features can be extremely high-dimensional, we use Principal Component Analysis (PCA) to reduce their size while keeping the most important information (accounting for the most variance) in the form of various eigenvectors. Our model takes in a 224x224 chest X-ray as input and outputs a binary label (either "Normal" or "Pneumonia").

We compare three different setups:

1. Standalone CNN: Standard ResNet50 classification
2. CNN + DT (FC layer): Decision Tree on raw CNN features (and PCA-reduced features)

3. CNN + DT (Conv2, Conv4, FC): Decision Tree on raw CNN features (and PCA-reduced features)

These setups allow us to evaluate how dimensionality reduction (PCA) and interpretable tree structures (Decision Trees) affect accuracy, interpretability, and computational efficiency. By striking a balance between CNN's complex feature extraction and Decision Trees' explainability, we aim to create a more transparent model suited for high-stakes environments like medical diagnostics.

Related Work

Previous research aimed at enhancing CNN interpretability typically falls into two categories: combining CNNs with easily interpretable models and using dimensionality reduction techniques on CNN-extracted features.

Several studies integrated interpretable models directly with CNNs. Zhang et. al [2] developed a decision-tree-based approach to explain CNN predictions clearly by breaking down complex CNN features. Though informative their method mainly utilizes deep-layer CNN features, missing valuable information from lower layers. Zhang, Wu, and Zhu [3] presented Interpretable CNNs that associate CNN filters directly with recognizable patterns. Their method offers clear visual explanations but involves substantial changes to the actual CNN structure, limiting adaptability.

Another group of methods measures interpretability in CNN representations. Bau et al. [4] proposed Network Dissection, which evaluates interpretability by linking CNN elements to other easily identifiable concepts. However, this method generates explanations after the CNN has already been trained, rather than embedding interpretability directly into the CNN architecture or training process.

Dimensionality reduction methods on CNN features are also popular. Gao et al. [5] introduced Neural Discriminative Dimensionality Reduction, which merges CNN features from multiple layers to enhance performance. Although effective for accuracy purposes, its complexity can reduce transparency making it challenging for straightforward interpretation in critical areas like medical diagnosis.

NASA researchers [6] created a classifier system combining CNNs with dimensionality reduction to generate understandable decision rules. Despite strong performance, their method's complexity in generating rules may limit clarity, especially in clinical settings where clarity is absolutely necessary.

Traditionally, tasks like medical diagnosis rely solely on manual analysis by clinicians, highlighting the necessity for clear and trustworthy automated methods. Our project uniquely addresses these limitations by combining multi-layer CNN feature extraction, PCA for dimensionality reduction, and simple decision trees, balancing clarity and effectiveness.

Dataset and Features

The dataset used in this study is a publicly available chest X-ray [7] dataset from Kaggle, containing 2,186 labeled X-ray images classified as either Normal (healthy lungs) or Pneumonia (infected lungs, including both bacterial and viral cases). The dataset is split into 1,562 training images (70%) and 624 test images (30%) to ensure proper model training and evaluation.

Several preprocessing steps are applied to standardize the dataset and optimize model performance. First, all images are resized to 224 x 224 pixels, which is the required input image size for ResNet50. This ensures that each image has a uniform dimension, preventing inconsistencies that could arise from different image sizes. Next, images are converted from BGR to RGB format since OpenCV loads images in BGR, while most deep learning models (including ResNet50) expect images in RGB. This conversion prevents color misinterpretation when using ResNet50 that could impact feature extraction and classification.

To further enhance the model's stability, we apply normalization using ImageNet's predefined mean and standard deviation values: mean = [0.485, 0.456, 0.406] and standard deviation = [0.229, 0.224, 0.225]. Normalization ensures that pixel values remain within a consistent range, preventing any extreme values/outliers from having too much effect. This ensures stable weight updates and a smoother model training process.

Additionally, images are loaded in smaller batches of 32 to balance efficiency and performance. Processing one image at a time would update the weights too frequently, leading to instability, while loading the entire dataset at once would require excessive memory. Smaller batches allow for controlled weight updates, improving training stability and speed. Lastly, the dataset is also shuffled before each epoch to prevent the model from memorizing image order, leading to better model generalization.

Methods

Feature Extraction from CNN

X-ray images are forward passed into a pre-trained ResNet50, where hooks are used to extract feature maps from Convolutional Layer 2, 4, and the FC layer. Hooks are functions which intercept the output of individual layers in a neural network, storing the output maps in vectors.

These features capture hierarchical representations of input images, from low-level edges and regions to high-level abstract patterns. The extracted feature maps from each layer are flattened to create a single feature vector for each image, which is used to train the decision tree in the following step. Extracting features from multiple layers allows the model to leverage both intermediary information and fine-grained details, significantly improving the decision-making process of the subsequent classifier.

Principal Component Analysis (PCA)

Given the high dimensionality of CNN-extracted features (~250,000 per image), PCA is applied to reduce feature size while preserving ~80% of the variance. This transformation improves computational efficiency and mitigates overfitting. PCA works by projecting the high-dimensional feature space onto a lower-dimensional area while identifying the combinations of features which capture the most variance and retaining only the most significant components. By reducing redundancy in the features, PCA enables the Decision tree to generalize better and train faster.

Decision Tree Training

A Decision Tree Classifier is trained on both the CNN-extracted feature vector and the PCA-transformed feature vectors. The tree splits on activation values, creating interpretable decision rules for classification. We use Gini impurity as the splitting criterion, with $Gini\ Impurity = 1 - \sum_{i=1}^{K} p_i^2$, where K = # class labels and p_i is the proportion of the i^{th} class label, and set a maximum depth of 10 to balance complexity and interpretability in the tree. The Decision Tree structure enables us to trace classification decisions back to specific CNN feature maps, allowing deeper analysis of feature importance. Since decision trees operate by recursively splitting the feature space, PCA helps improve efficiency by ensuring that splits occur in the most effective directions.

Experiments/Results/Discussion

As mentioned in the introduction, our project experimented with 3 different setups:

1. Standalone CNN: Standard ResNet50 classification
2. CNN + DT (FC layer): Decision Tree on raw CNN features (and PCA-reduced features)
3. CNN + DT (Conv2, Conv4, FC): Decision Tree on raw CNN features (and PCA-reduced features)

In this section, we compare the results of the three aforementioned studies using the following evaluation metrics:

1. Run Time: Duration required to train the model and test the data
2. Accuracy: Ratio of Accurately Classified Points = (TP + TN) / (TP + TN + FP + FN)
3. Precision: Proportion of positive predictions actually positive = TP / (TP + FP)
4. Recall: Proportion of actual positives classified as positive = TP / (TP + FN)

Setup 1 acts as a baseline model for accuracy, with a ResNet50 CNN fully trained over 5 epochs, allowing for analysis of changes in accuracy when interpretability is prioritized in the other two experiments. Setup 2 replicates the methods of previous work, combining only the feature vector from the final fully connected layer of the CNN with a decision tree, while Setup 3 implements our novel approach, combining feature vectors from multiple CNN layers with decision trees. Each decision tree also presents a visualization of its structure along with the evaluation metrics for a look into the interpretability of the model. These three setups allow for a direct comparison of the impact of using multi-layer CNN feature extraction versus using only the final-layer CNN features.

Results

Model	Runtime	Accuracy	Precision	Recall
Standalone CNN	~**2.5 hours**	83.81%	0.8	0.98
CNN + DT (FC layer)	~10:32 (extract) ~0:02 (tree) = **10:35 mins**	74.52%	0.74	0.93
CNN + PCA + DT (FC layer)	~10:32 (extract) ~0:006 (PCA+tree) = **10:33 mins**	73.56%	0.74	0.90
CNN + DT (Conv2, Conv4, FC)	~11:10 (extract) ~8:25 (tree) = **19:36 mins**	79.65%	0.77	0.95
CNN + PCA + DT (Conv2, Conv4, FC)	~11:10 (extract) ~1:43 (PCA+tree) = **12:54 mins**	79.17%	0.79	0.92

Table 1. Depicts a table of results for each experiment, containing runtimes, accuracy, precision, and recall values

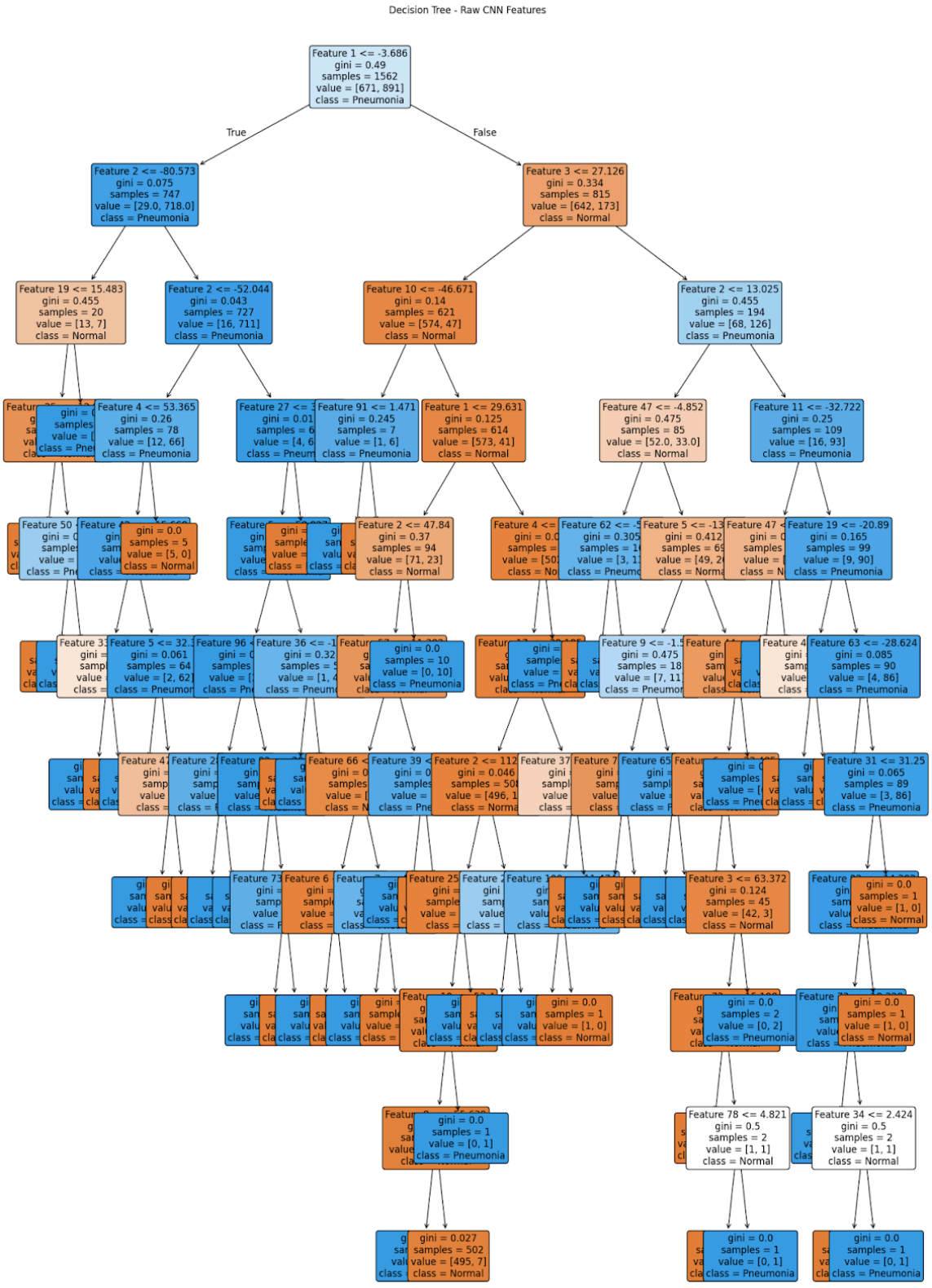

Figure 1. Decision tree structure for experiment 2, CNN + DT (FC layer). Each node highlights the threshold value used to split the tree for each feature.

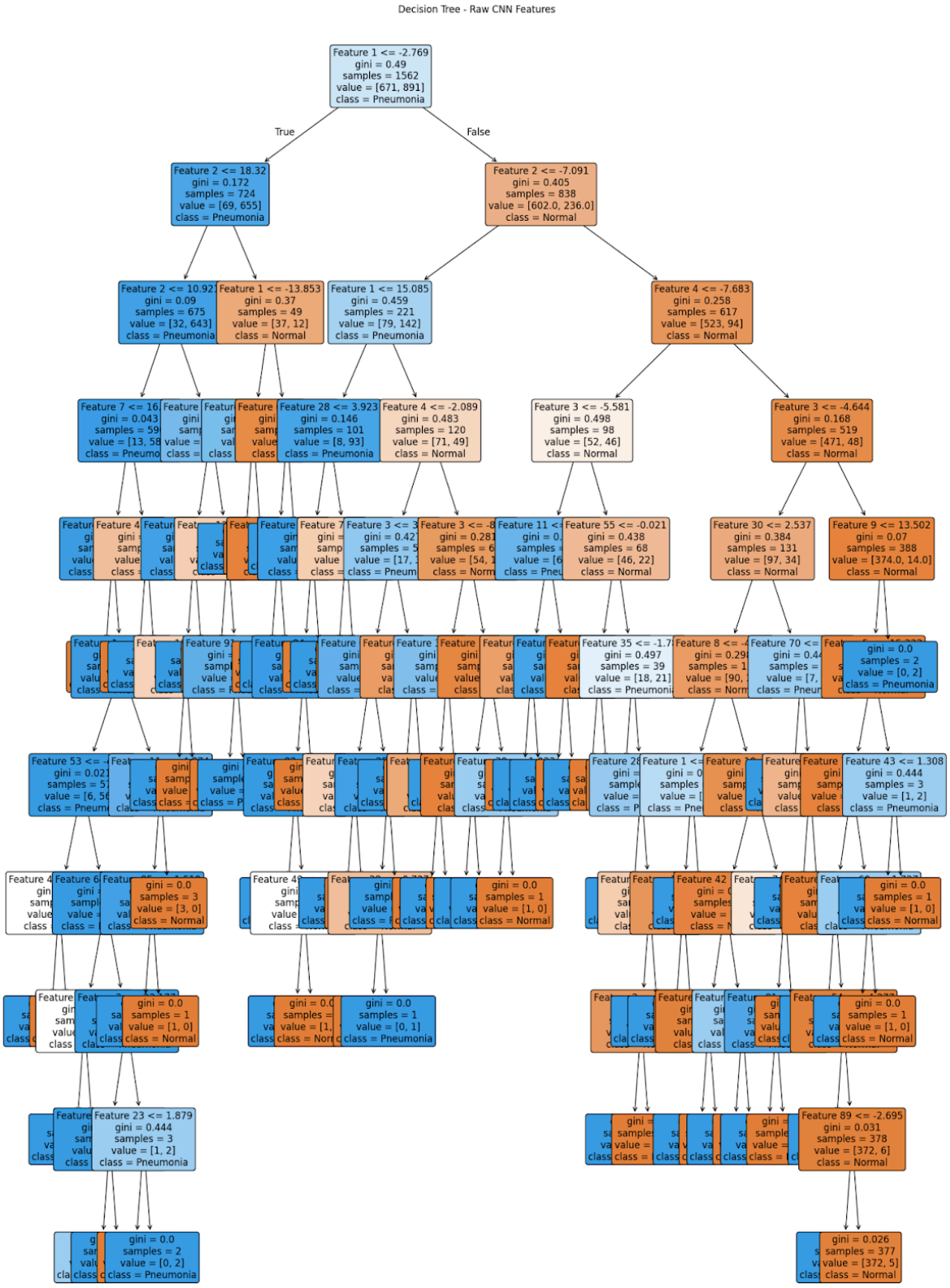

Figure 2. Decision tree structure for experiment 3, CNN + DT (Conv2, Conv4, FC). Each node highlights the threshold value used to split the tree for each feature.

Discussion

As seen in Table 1, the standalone CNN model (ResNet50) achieved the highest accuracy at 80.45%, demonstrating its strong classification performance. However, as a black-box model, it lacks transparency in decision-making, making it less suitable for applications requiring interpretability like for x-ray classification. This model also had the longest runtime, taking 2.5 hours for merely 5 epochs. The CNN + Decision Tree (Multi-Layer) approach retained strong accuracy while significantly improving interpretability and runtime, demonstrating that a small trade-off in accuracy can provide meaningful decision explanations. In contrast, the CNN + Decision Tree (Final Layer Only) approach performed worse, with an accuracy of 74.52%, confirming that leveraging multi-layer features improves classification performance. Additionally, PCA slightly reduced accuracy but played a crucial role in improving decision tree training efficiency. By reducing the feature dimensionality, PCA led to shorter runtimes for building decision trees, making the approach more computationally efficient without significantly impacting classification results. Figure 1 and 2 depict interpretability achieved through the decision tree models, outputted by experimental setup 2 and 3, respectively. At each node, the trees depict threshold values for each feature analyzed with ResNet50. This highlights exactly which features are used for each classification, a vast interpretability improvement from the black-box nature of regular CNNs.

Conclusion/Future Work

Our study demonstrates that the hybrid CNN + PCA + Decision Tree model balances interpretability and accuracy, making it more suitable for medical AI applications than traditionals CNN models. CNN-only models, particularly ResNet50, achieved the highest accuracy but lacked transparency, limiting their usability in high-stakes decision-making. The CNN + Decision Tree (Multi-Layer) model retained strong accuracy while significantly improving explainability, confirming that using multi-layer features enhances performance. The final-layer-only Decision Tree model performed the worst (74.52%), reinforcing the importance of utilizing multiple feature layers. PCA slightly reduced accuracy in each model, but led to better decision tree efficiency (nearly 7 mins faster) by lowering feature dimensionality, which improved model runtime without sacrificing much information (≤1% accuracy loss).

Future improvements can focus on optimizing PCA by experimenting with different numbers of principal components to achieve a better balance between interpretability and accuracy. Testing alternative classifiers, such as Random Forest and XGBoost [8], may enhance generalization while maintaining transparency. Additionally, refining feature selection by exploring different CNN layers, such as Conv3 and Conv5, could provide more informative features for classification. Expanding our dataset to include diverse medical imaging data, including MRI and CT scans, would allow for better generalization across medical fields. Furthermore,

incorporating explainability techniques like SHAP or LIME [9] can offer deeper insights into how CNN-derived features contribute to decision tree predictions, further improving trust in model decisions. Addressing these areas will help refine our approach, making AI-driven diagnostics more interpretable and applicable in real-world technical settings.

Contributions

Code: Kanishk and Chetan
Abstract: Kanishk
Introduction: Chetan
Related Work: Kanishk and Chetan
Dataset and Features: Chetan
Methods: Kanishk
Experiments/Results/Discussion: Kanishk
Conclusion/Future Work: Chetan

References

[1] "ResNet-50 convolutional neural network - MATLAB resnet50," *www.mathworks.com*. https://www.mathworks.com/help/deeplearning/ref/resnet50.html
[2] Q. Zhang, Y. Yang, H. Ma, Y. Wu, S. Jiao, and T. University, "Interpreting CNNs via Decision Trees." Available: https://openaccess.thecvf.com/content_CVPR_2019/papers/Zhang_Interpreting_CNNs_via_Decision_Trees_CVPR_2019_paper.pdf
[3] Q. Zhang, Y. Wu, and S.-C. Zhu, "Interpretable Convolutional Neural Networks." Available: https://openaccess.thecvf.com/content_cvpr_2018/papers/Zhang_Interpretable_Convolutional_Neural_CVPR_2018_paper.pdf
[4] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network Dissection: Quantifying Interpretability of Deep Visual Representations," *arXiv.org*, 2017. https://arxiv.org/abs/1704.05796
[5] Y. Gao, J. Ma, M. Zhao, W. Liu, and A. L. Yuille, "NDDR-CNN: Layerwise Feature Fusing in Multi-Task CNNs by Neural Discriminative Dimensionality Reduction," *arXiv.org*, 2018. https://arxiv.org/abs/1801.08297
[6] J. Owens, K. Datta Gupta, X. Yan, L. Zeleke, and A. Homaifar, "Interpretable Convolutional Learning Classifier System (C-LCS) for higher dimensional datasets." Available: https://ntrs.nasa.gov/api/citations/20220014172/downloads/SMC22_0709_FI.pdf
[7] P. MOONEY, "Chest X-Ray Images (Pneumonia)," *www.kaggle.com*, 2018. https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia
[8] Nvidia, "What is XGBoost?," *NVIDIA Data Science Glossary*, 2024. https://www.nvidia.com/en-us/glossary/xgboost/

[9] S. Fathima, “LIME vs SHAP: A Comparative Analysis of Interpretability Tools,” *www.markovml.com*, Feb. 26, 2024. https://www.markovml.com/blog/lime-vs-shap