Interpretable Image Classification

Using Neural ProtoTrees for fine-grained interpretable image classification

By:
Team 14
Dhaval Potdar
Simrun Sharma
Xueqing Wu

Table of Contents

Abstract

At present, patients experience a two-day delay in receiving their chest X-ray results, causing anxiety and treatment delays. To expedite this process, we aim to develop a machine learning model for disease identification from chest X-rays. While, most current models lack interpretability, leading to traceability risks in critical medical decisions. Our project focuses on identifying Pleural Effusion, Cardiomegaly, and normal findings from 9,000 chest X-ray images using an interpretable machine learning model, ProtoTree. We evaluate the accuracy and interpretability trade-off by comparing the ProtoTree model with a baseline ResNet50 model. While the ProtoTree model achieves a comparable accuracy to the ResNet50 model (62% vs. 72%), it provides a decision tree that explains the decision making process, enhancing interpretability. Our future work aims to improve accuracy, generalize to more diseases, and address class imbalance challenges.

Introduction

A chest X-ray is a crucial diagnostic tool used in the early detection of various diseases in the chest region affecting the heart, lungs, and bones. Typically, a radiologist interprets the X-ray images, with results taking one to two days to be processed, leading to heightened patient anxiety ("Chest X-ray (CXR)," n.d.). In a healthcare setting, having fast and accurate diagnostic tools is crucial for timely treatment decisions and patient care. While deep learning models like CNNs offer high accuracy in image recognition tasks, they often lack transparency in their decision-making process. This opacity can be a significant limitation in healthcare, where understanding why a model makes a particular diagnosis is as important as the accuracy of the diagnosis itself.

Our project introduces the Neural Prototype Tree, or ProtoTree, as an interpretable method for image recognition in chest X-rays. This model provides interpretable insights into how it arrives at a diagnosis, making it a valuable tool for improving decision-making in healthcare settings. It combines a Convolutional Neural Network (CNN) with a soft decision tree, trained using a cross-entropy loss function. Unlike conventional black box models, the ProtoTree integrates interpretability directly into its structure, addressing the growing need for models that can be easily understood by non-specialists. The ProtoTree aims to assist overburdened radiologists and physicians unfamiliar with X-ray interpretation, either by providing additional support in decision-making or by highlighting details that may have been overlooked.

Our study focuses on the trade-off between accuracy and interpretability by comparing the performance of a baseline ResNet50 model with our interpretable ProtoTree model. By evaluating both models, we aim to determine the optimal approach for healthcare diagnosis, balancing accuracy with timely results.

Background

When exploring chest X-ray image classification problem space for machine learning models, we encounter various approaches and methodologies. Sreejith & George conducted a study using a ResNet50 model, which serves as a component for our ProtoTree model. They focused on

detecting Covid-19 from 1,000 chest X-rays, achieving an accuracy of 96% in Covid-19 detection (Sreejith & George, 2021). However, despite achieving high accuracy, this study's methodology lacks explainability in the decision-making process for Covid-19 detection.

Similarly, Tang et al. utilized the NIH chest X-ray dataset, the same dataset used in our experiment, to compare the performance of deep learning models UNet and XLSor for chest image classification. The UNet model achieved an accuracy of 97%, while the XLSor method achieved an accuracy of 95% (Tang et al., 2019). Despite their high accuracy, both models lack interpretability in decision-making, crucial for stakeholders such as patients and physicians in high-stakes medical care settings.

To address this interpretability challenge, we turn to ProtoTree modeling. Developed in 2021 by Nauta & Bree, ProtoTree adds a soft decision tree to the original CNN, combining prototype learning with decision trees to create a globally interpretable model by design. Additionally, ProtoTree can provide local explanations for single image predictions by outlining the decision path through the tree.

In Nauta and Bree's study, they trained the ProtoTree model on a hummingbird dataset to identify hummingbird species. The model produced a hierarchical structure of the decision tree, explaining how it distinguishes between scarlet tanagers and summer tanagers based on feather color, achieving an accuracy of approximately 80% in identifying bird species (Nauta & Bree, 2021). Our goal is to achieve a similarly high accuracy while maintaining an interpretable decision tree to aid in chest medical diagnosis.

Data

The dataset we are utilizing is the NIH Chest X-ray from Kaggle, containing 112,120 chest X-rays with disease labels from 30,805 unique patients. However, for our study, we have narrowed down our training dataset to three classes we are focusing on. We used a 30-70 test-train split on 1093 images for Cardiomegaly, and 3000 for Effusion and No Finding each. This indicates potential challenges in modeling due to class imbalance.

For our study, the dataset comprises three distinct categories: No Findings, Cardiomegaly, and Pleural Effusion. As shown in **Figure 1**, Cardiomegaly, characterized by an enlarged heart, typically arises as a consequence of an underlying pathological process and can manifest in various forms of primary or acquired cardiomyopathies (Amin & Siddiqui, 2022). This condition can affect the right, left, or both ventricles or the atria of the heart. We selected this class to aid in the identification of larger heart diseases, as Cardiomegaly often serves as a primary indicator of other heart conditions.

Seen in **Figure 2,** Pleural Effusion is defined as the accumulation of fluid in the pleural cavity, the space between the lungs and the chest wall. This condition can be caused by various underlying diseases or conditions, including congestive heart failure, cancer, pneumonia, and pulmonary embolism (Jany & Welte, 2019). Of these, lung cancer is particularly noteworthy as a common cause of malignant Pleural Effusion.

Lastly as shown in **Figure 3**, the "No Findings" class indicates instances where neither condition nor any other chest disease is detectable in the X-ray image. Our objective was to gain insights into early signs of heart and lung conditions, which could serve as primary indicators of more serious chest illnesses. Identifying these conditions early is crucial for preventing them from developing into more severe problems.

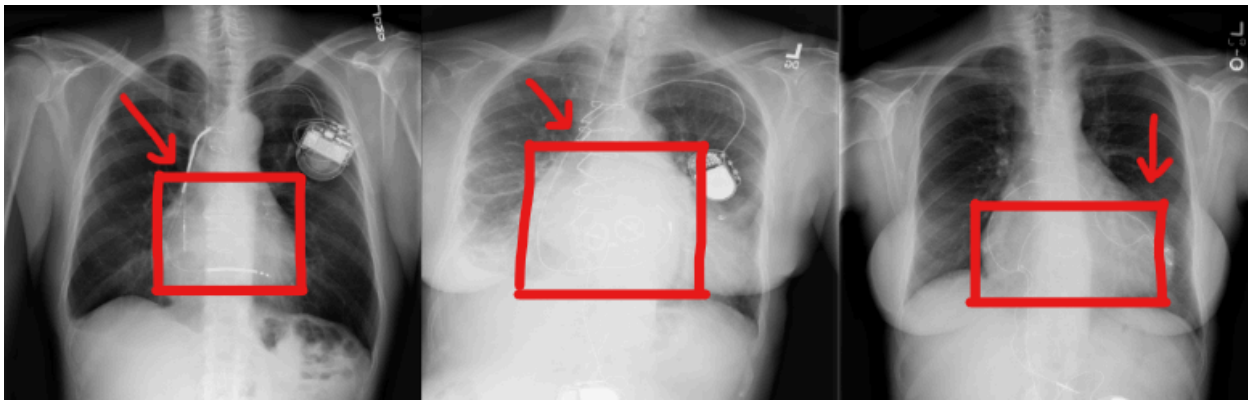


Figure 1: Cardiomegaly

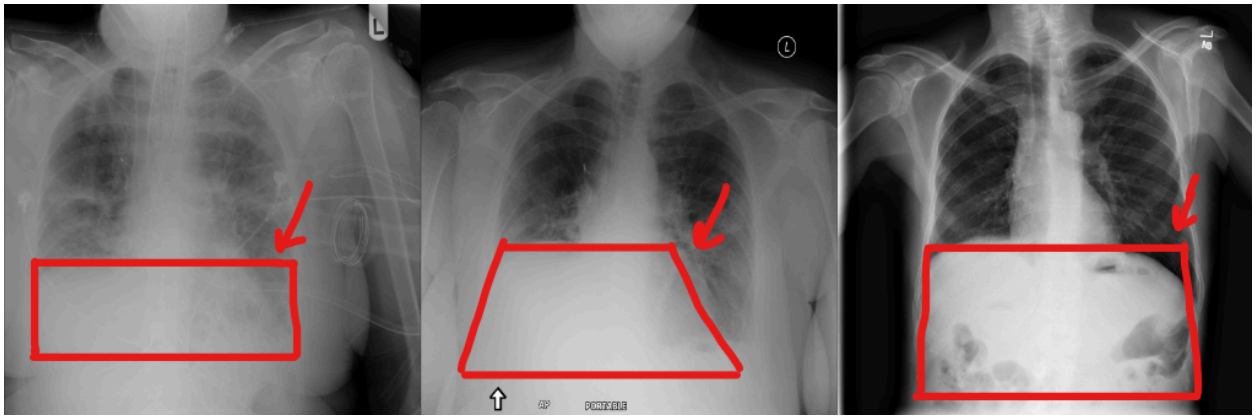


Figure 2: Pleural Effusion

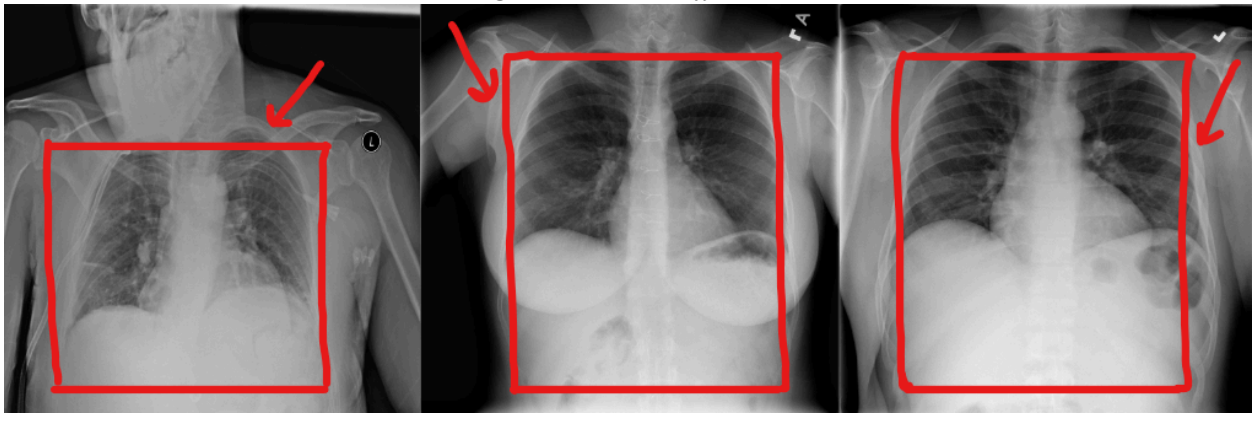


Figure 3: No Finding

Experimental Design

The experiments are divided into three major parts: data preprocessing, modeling, and model evaluation shown below in **Figure 4**.

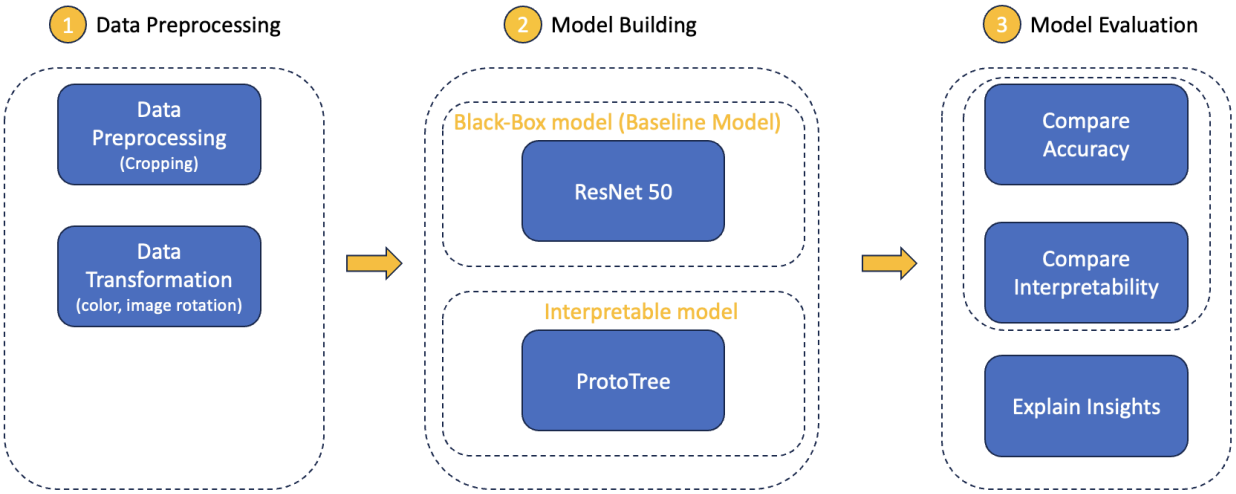


Figure 4: Flowchart detailing ProtoTree Experimental Design

In this experiment, we aim to investigate the tradeoff between accuracy and interpretability in machine learning models. To establish a fair comparison, we will first assess the performance of a baseline ResNet50 model against our ProtoTree model. This comparison will be based on four evaluation metrics: F1 scores, accuracy, precision, and recall, analyzed using a 30-70 test-train split. Additionally, we will evaluate the models' ability to provide local and global explanations, which serves as an interpretability metric between the two models.

Our experiment aims to showcase the tradeoff between accuracy and interpretability in machine learning models. By comparing the ProtoTree model, which provides both accuracy and interpretability metrics, with the baseline ResNet50 model, which only provides accuracy metrics, we aim to determine which model is more valuable based on the perspective of the end user.

Data Preprocessing

In this report, data preprocessing involves several key steps. First, all images are resized to 224×224. Image transformation is then performed to adjust color, with parameters tuned based on the ProtoTree method and other relevant papers detailed in the Background section. Further data tuning to optimize X-ray quality before modeling can be found in the **Appendix** labeled Data Preprocessing.

Modeling

We frame our project as a classification problem. Given an input image, our model generates a class probability distribution over the labels. We train the model by minimizing the cross-entropy objective.

Baseline Model

We employed an untrained ResNet50 model as our baseline for performance comparison with our ProtoTree model. ResNet50 was selected for its use of skip connections, which aid in faster convergence during training, thereby mitigating the vanishing gradient problem, and for its deep residual learning approach, which has been known to deliver state-of-the-art performance in image recognition tasks (Kirandeep, Kaur, & Dhir, n.d.). Depending on the chosen number of classes, the baseline model can adjust the number of nodes in the final layer of the neural network. For our purposes, we configured the model to have 3 nodes in its final layer.

ProtoTree Model

The ProtoTree model is a combination of a Convolutional Neural Network (ResNet50 in this case), and a soft binary decision tree structure. An input image is forwarded through convolution network which consists of D two-dimensional $H \times W$ feature maps. The feature maps serve as the input to the binary tree. The tree itself consists of a set of leaf nodes and edges. The probability of routing a sample through the right edge is calculated using a similarity function, which measures the similarity between a patch of the image and the learned prototype. The final prediction is thus the product of all the probabilities along the route the image traverses within the tree. A explaining the general architecture of the ProtoTree can be found in the **Appendix**.

Results

We trained the ProtoTree for 100 epochs; **Figure-5** shows the training curves. We chose 100 epochs because we didn't see significant gains in further training. To make it a fair comparison, we also trained the ResNet50 for 100 epochs.

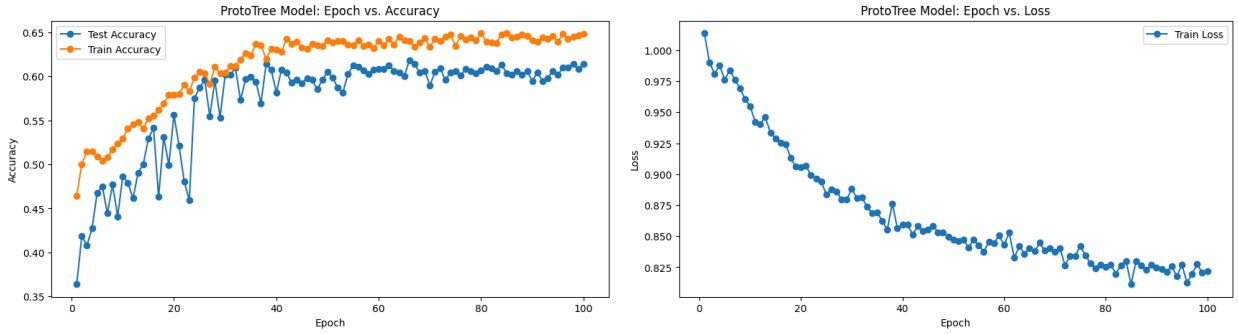


Figure 5: ProtoTree training curves

The following **Table-1** summarizes the results for both the ResNet50 and the ProtoTree models trained on 70% of the data.

		ResNet50			ProtoTree		
Label	**Support**	**Precision**	**Recall**	**F1 Score**	**Precision**	**Recall**	**F1 Score**
Cardiomegaly	316	75%	54%	63%	60%	2%	2%
Effusion	895	75%	85%	79%	66%	90%	76%
No Finding	501	65%	61%	63%	52%	50%	51%

Table 1: Performance Metrics

The overall accuracy for the baseline ResNet is 72% and 62% for ProtoTree. The advantage of the ProtoTree as seen in **Table-2**, is that for each prediction, we have a visualization of the exact decision-making process as explained in the following section. And so, we not only have a high-level perspective on how the model would work given an input image, but also the representation of how it actually does.

Interpretability Dimension	ResNet50	ProtoTree
Local Explanations	No	Yes
Global Explanations	No	Yes

Table 2: Interpretability Metrics

The confusion matrices presented in **Figure-6** for both the models show comparable performance for Effusion and No Finding for both models. The greatest dip in performance comes from the Cardiomegaly class. In our dataset we only have 1093 samples of Cardiomegaly as opposed to 3000 in the other two classes. This shows that ProtoTree is susceptible to class imbalance. To remedy this situation, we tried to upsample the images in the underrepresented class, but this led to even worse performance due to overfitting, and so we left rectifying the class imbalance as part of our future work.

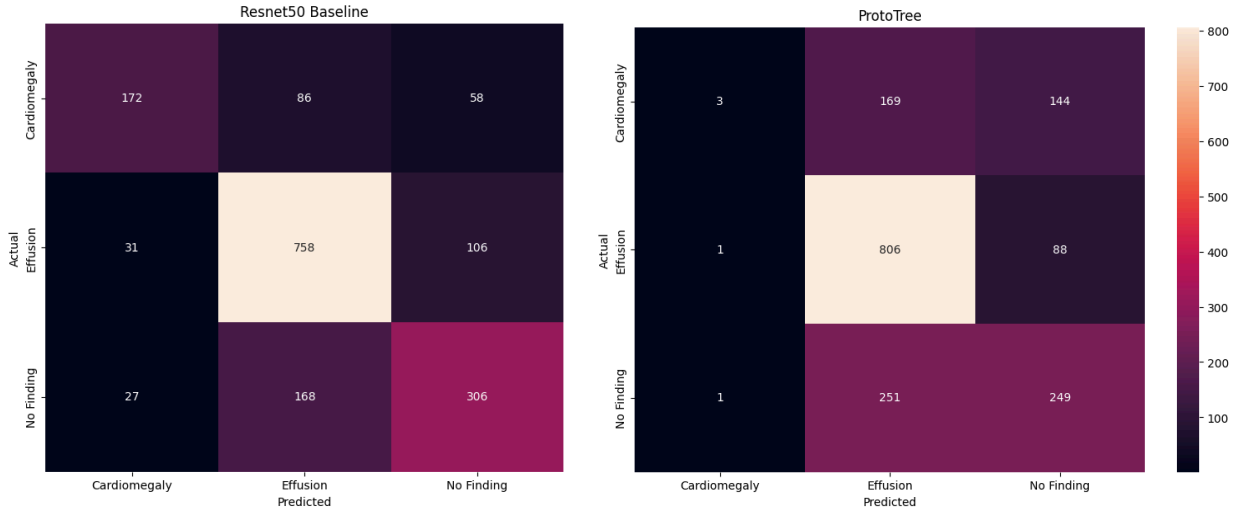


Figure 6: Global Explanation of ProtoTree Model

In our analysis, we shift our focus towards the interpretability metric, specifically aiming to compare the baseline ResNet50 model with the ProtoTree model. This comparative evaluation entails a thorough assessment of both global and local explanations offered by the ProtoTree model.

Global Explanation

The ProtoTree model initiates its process by selecting a set of kernels from the ResNet50 model, considering them as candidate prototypes. These kernels are then structured into a decision tree through a greedy splitting approach. Once a kernel is assigned to a specific node in the tree, it is not reused for further node decisions. The resulting decision tree is constructed to contain the most suitable prototypes, with the number of prototypes determined by the depth of the tree (5 in our case). This holistic explanation framework provides insights into the path a test image would take through the model.

In **Figure 7**, we present the resultant decision tree generated for the chest X-ray dataset. At each node within this tree, a prototype is depicted on the left, while the corresponding image patch from which the prototype was extracted is displayed on the right. The prediction at each leaf node is computed by aggregating probabilities along the path leading to that particular leaf. These probabilities are then summed up by class, followed by normalization, resulting in the final prediction represented as a probability distribution over the *K* classes.

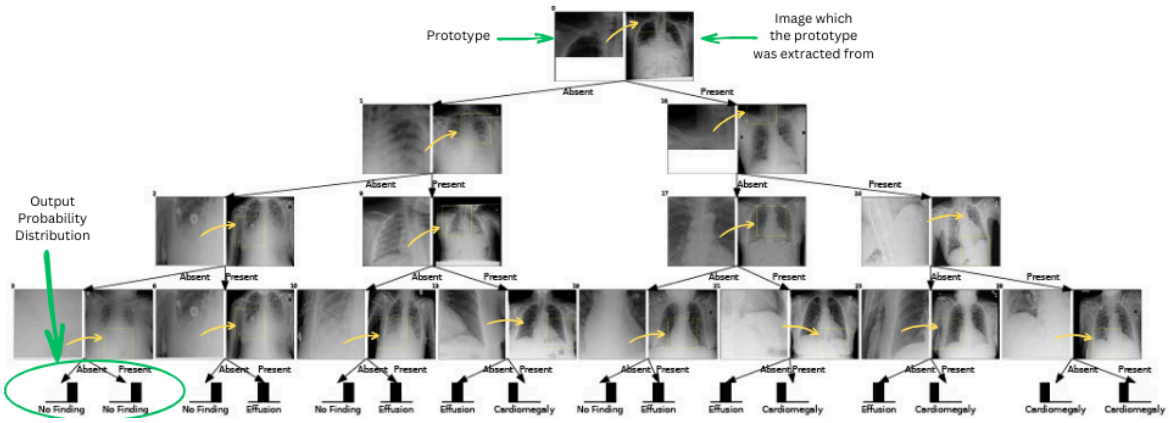


Figure 7: Global Explanation of ProtoTree Model

Local Explanation

In **Figure 8**, we illustrate the process of generating a local explanation for a given test image belonging to the Pleural Effusion class, depiction of this class is denoted by the milky shaded region in the middle of the chest X-ray. At each node in the tree, we compute a similarity score between a prototype and the image. This score reflects the degree to which the prototype's features are present within the test image. At the root node, the similarity score is found to be 33%. Consequently, the model assigns 33% probability to the right branch and 66% to the left branch. Moving along the "winning"

branch (in this instance, the left one), at the subsequent node, a new prototype is considered, and a similarity score of 54% is computed. This score indicates the presence of a patch resembling the prototype within the test image, leading to the selection of the right branch as the "winning" branch for this node. This iterative process continues until the termination of the tree. It is essential to emphasize that at no stage do we discard the "losing" branch; rather, we continue the same process for it. The final prediction is determined at each leaf by multiplying the probabilities along all the branches leading to that leaf.

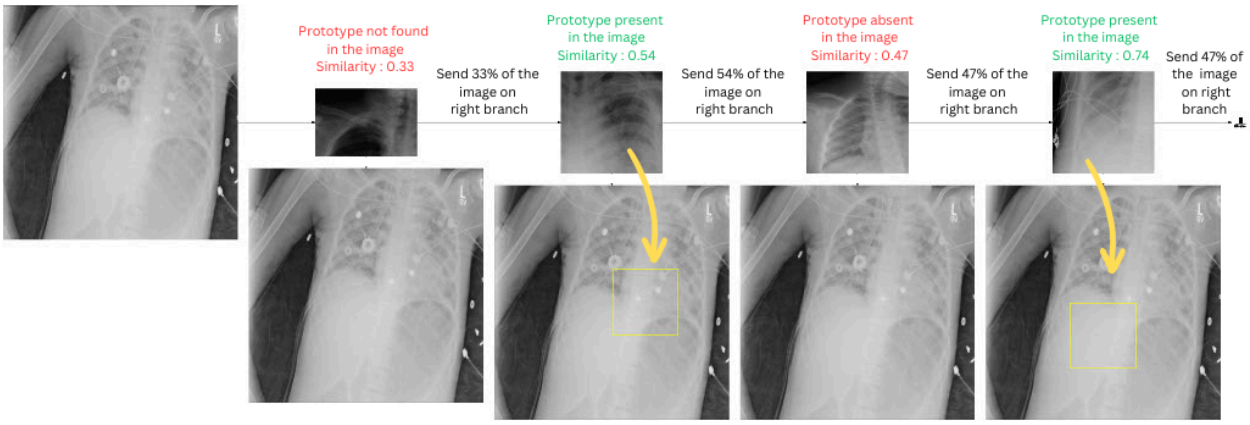


Figure 8: Local Explanation of the Image: Pleural Effusion

With the ResNet50 baseline, we cannot explain actual predictions, and although we achieve 10% higher accuracy, it is at the cost of a black-box architecture. Once a prediction has been generated it's difficult to determine its faithfulness because we don't have insight into how the prediction was generated. The ProtoTree, on the other hand, lays out the exact decision-making process end-to-end, and we can bring in external knowledge to either corroborate the decision-making process or discard the prediction. We can also debug the ProtoTree model more swiftly than a black-box model by identifying any obvious gaps in the global structure of the ProtoTree.

Conclusion

In summary, we successfully developed an interpretable machine learning model for disease identification from chest X-rays. The ProtoTree model, although slightly less accurate (62%) than the ResNet50 model (72%), offers invaluable interpretability that is crucial for high-stakes decision-making in healthcare. However, there are key areas for future work and challenges that need to be addressed.

One significant future direction is to improve the overall accuracy of the model. While interpretability is essential, accuracy is paramount in medical decision-making. One approach is to train a ResNet50 model from scratch on a larger, more diverse dataset of chest X-ray images. Alternatively, we could explore using pre-trained models specifically trained on chest X-ray data to enhance the model's proficiency in chest X-ray detection.

Another challenge is the class imbalance, particularly concerning Cardiomegaly. Collecting more images related to Cardiomegaly could help mitigate this issue and improve the model's performance on this specific condition. Additionally, we aim to generalize the model to identify more types of diseases beyond Pleural Effusion, Cardiomegaly, and normal findings. This will require further experimentation and hyperparameter tuning to ensure the accuracy and reliability of the model across different disease categories.

In conclusion, while we have made significant progress in developing an interpretable model for disease identification from chest X-rays, there are ongoing challenges and opportunities for improvement. By addressing these challenges and expanding the model's capabilities, we can enhance its utility in clinical practice and improve patient outcomes.

Roles

Simrun:

- Research on domain knowledge
- Image data cleaning
- Interpret global and local explanation

Annie:

- Image preprocessing
- Model evaluation
- Project management

Dhaval:

- Setup environment for model building
- Train and tune ResNet50 model
- Train and tune ProtroTree model

All members have participated in coding. Each member formulated the presentation and report for the parts they have done.

References

Angelov, P., & Soares, E. (2020, February 2). Towards deep machine reasoning: A prototype-based deep neural network with decision tree inference. arXiv.org. https://doi.org/10.48550/arXiv.2002.03776

Amin H, Siddiqui WJ. Cardiomegaly. [Updated 2022 Nov 20]. In: StatPearls [Internet]. Treasure Island (FL): StatPearls Publishing; 2024 Jan-. Available from: https://www.ncbi.nlm.nih.gov/books/NBK542296/

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PLOS ONE, 10(7). https://doi.org/10.1371/journal.pone.0130140

Better Health Channel. (n.d.). X-ray examinations. Better Health Channel. https://www.betterhealth.vic.gov.au/health/conditionsandtreatments/x-ray-examinations

Brendel, W., & Bethge, M. (2019). Approximating CNNs with bag-of-local-features models works surprisingly well on ImageNet. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net.

Hossain, Md. B., Iqbal, S. M. H., Islam, Md. M., Akhtar, Md. N., & Sarker, I. H. (2022). Transfer learning with fine-tuned deep CNN RESNET50 model for classifying covid-19 from chest X-ray images. Informatics in Medicine Unlocked, 30, 100916. https://doi.org/10.1016/j.imu.2022.100916

Jany, B., & Welte, T. (2019). Pleural Effusion in Adults-Etiology, Diagnosis, and Treatment. Deutsches Arzteblatt international, 116(21), 377–386. https://doi.org/10.3238/arztebl.2019.0377

Kirandeep, Kaur, R., & Dhir, V. (2023). Image recognition using ResNet50. https://www.eurchembull.com/uploads/paper/3d3038bc3e7d1fad63f18fa13fe711ac.pdf

Lee, G. M., & Walker, C. M. (2023). Pleural thickening: Detection, characterization, and differential diagnosis. Seminars in Roentgenology, 58(4), 399–410. https://doi.org/10.1053/j.ro.2023.06.001

Nauta, M., & Bree, R. (2021). Neural Prototype Trees for Interpretable Fine-grained Image Recognition.

National Institutes of Health. (2017). NIH Chest X-rays. Kaggle. Retrieved from https://www.kaggle.com/datasets/nih-chest-xrays/data

Saito, A., Hakamata, Y., Yamada, Y., Sunohara, M., Tarui, M., Murano, Y., Mitani, A., Tanaka, K., Nagase, T., & Yanagimoto, S. (2019). Pleural thickening on screening chest X-rays: A

single institutional study. Respiratory Research, 20(1). https://doi.org/10.1186/s12931-019-1116-9

Sercan, A., & Tomas, P. (2019). ProtoAttend: Attention-Based Prototypical Learning.

Sreejith, V., & George, T. (2021). Detection of COVID-19 from chest X-rays using resnet-50. Journal of Physics: Conference Series, 1937(1), 012002. https://doi.org/10.1088/1742-6596/1937/1/012002

Tang, Y.-B., Tang, Y.-X., Xiao, J., & Summers, R. M. (2019). XLSor: A Robust and Accurate Lung Segmentor on Chest X-rays Using Criss-Cross Attention and Customized Radiorealistic Abnormalities Generation.

Zheng, H., Fu, J., Mei, T., & Luo, J. (2017). Learning multi-attention convolutional neural network for fine-grained image recognition. 2017 IEEE International Conference on Computer Vision (ICCV). https://doi.org/10.1109/iccv.2017.557

Appendix

Data Processing

The parameters we finally use for the data preprocessing:

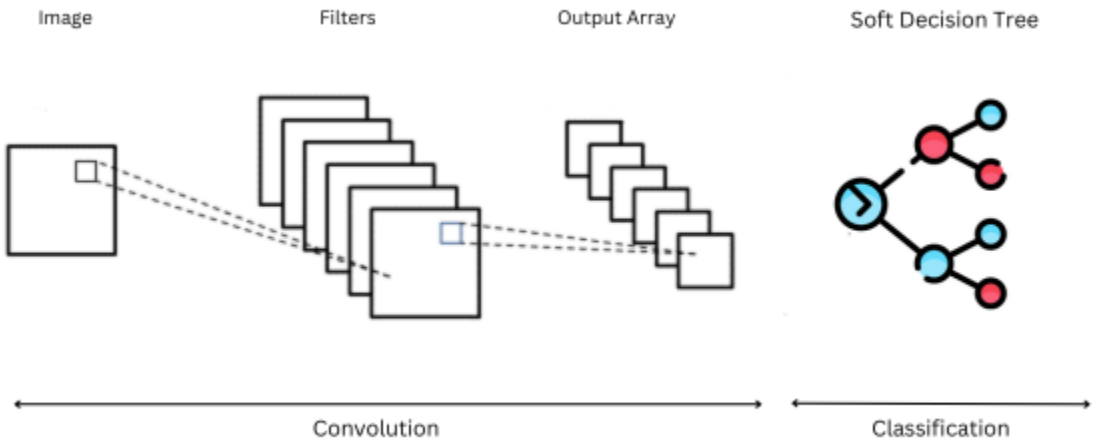
- **Transforms.normalize:** Our default values for the mean are (0.485, 0.456, 0.406) and standard deviation values are (0.229, 0.224, 0.225).
- **Transforms.Resize:** this function is used to resize the images to a specified size, which in our case adds an additional 32 the img_size to give 224.
- **Transforms.RandomPerspective:** this function is used to control the effect of viewing the image from a different angle or perspective. The two arguments that are modified are distortion scale and p. Distortion scale controls the degree of distortion applied to the image with 0 being no distortion and 1 being max distortion. We have set this to 0.5. "P" is the probability of the image being transformed, the default value of 0.5, meaning there is a 50% change of the transformation being applied.
- **Transforms.ColorJitter:** this function is used to augment images with variations in brightness, contrast, saturation, and hue. Each argument has a minimum or maximum and each value should be a non-negative number between 0 and 1.
- **Transforms.RandomHorizontalFlip:** this function is used to horizontally flip an image randomly with a given probability. "P" is the probability of the image being flipped horizontally between 0 and 1, there is a 50% chance the image will be flipped horizontally.
- **Transforms.RandomAffine:** this function is used to rotate, translate, scaling, and shearing with different ranges for each parameter. We have specified the range of rotation to be 15 degrees. Shearing displaces each point in the image by a distance proportional to its perpendicular distance from the axis of shearing. We have specified the value of (-2, 2) meaning the shear is only applied to the x-axis.
- **Transforms.RandomCrop:** this function is used to randomly crop an input image (currently our default image size is 224) however if someone were to pick a different image size, random crop is used to fit the size (224,224).
- **Transforms.ToTensor():** this function is used to convert the input image into a PyTorch tensor. It converts the image data from a NumPy array (or a PIL image) in the range [0, 255] to a float tensor of shape (C, H, W) where C is the number of channels (3 for RGB images) and H and W are the height and width of the image. This is necessary before feeding the images to a neural network.

ResNet 50 Parameters

- samplewise_center=True: Centers the data around the mean per sample (image), meaning the mean pixel value for each sample will be subtracted from all pixels in that sample.
- samplewise_std_normalization=True: Divides the data by the standard deviation per sample, ensuring that each sample has a standard deviation of 1.
- horizontal_flip=True: Randomly flips images horizontally, which can increase the diversity of the training dataset and improve model generalization.

- vertical_flip=False: Determines if images can be flipped vertically. In this case, vertical flipping is disabled.
- height_shift_range=0.2: Randomly shifts the height of the image by up to 20% of the total height. This augmentation can help the model learn to detect objects at different positions in the image.
- width_shift_range=0.2: Randomly shifts the width of the image by up to 20% of the total width. Similar to height shifting, this augmentation can improve the model's ability to detect objects at different positions.
- rotation_range=10: Randomly rotates the image by up to 10 degrees. This augmentation can help the model become more robust to variations in object orientation.
- shear_range=0.1: Applies a shear transformation to the image, where the shear angle is randomly chosen from the range [-0.1, 0.1]. Shearing can help the model learn to recognize objects from different angles.
- fill_mode='nearest': Determines the method used for filling in newly created pixels that may appear after a rotation or a width/height shift. 'Nearest' means that the value of the nearest pixel will be used.
- zoom_range=0.15: Randomly zooms into the image by up to 15%. This augmentation can help the model learn to recognize objects at different scales.

ProtoTree Architecture



The model consists of two distinct components: the standard convolution operation and the soft binary tree. The convolution operation operates in the conventional manner, while the soft binary tree constitutes the latter half of the model. The incorporation of a soft binary tree serves a specific purpose: to ensure the network remains differentiable, thereby facilitating training through gradient descent optimization techniques.