Comparative Analysis of CNN and ScatNet for Image Classification

Visual Intelligence Project

Name: Sogand Ghasemi

Matricola no.: VR512037

Date: 27-June-2025

1. Introduction

Lung cancer is one of the leading causes of cancer-related deaths worldwide. Accurate and early classification of histopathological images is essential for effective diagnosis and treatment planning. In this project, we explore and compare two classification pipelines: one using a Convolutional Neural Network (CNN) and another using a Scattering Transform (ScatNet). The goal is to evaluate their performance, interpretability, and effectiveness in distinguishing between benign and adenocarcinoma tissue samples. To further enhance model transparency, we apply Explainable AI (XAI) techniques to identify which features contribute most to each model's predictions, helping us better understand their decision-making process and potentially guide future model improvements.

2. Dataset and Preprocessing

We used a publicly available histopathological lung image dataset from Kaggle, containing microscopic tissue slides categorized into two classes: Adenocarcinoma (labeled as 1) and Benign (labeled as 0). The images were organized in class-specific folders, making the dataset easy to load and manage for binary classification tasks.

CNN Preprocessing: For the CNN model, we preserved the color information by loading all images in RGB format, which is important for histological analysis. Each image was resized to 224×224 pixels to match the CNN's input requirements. To improve generalization and reduce overfitting, data augmentation techniques such as random horizontal flips were applied. The images were then converted to PyTorch tensors and normalized to the range [−1, 1] using a mean and standard deviation of 0.5 per channel. Random seeds were fixed in both PyTorch and NumPy to ensure reproducibility.

ScatNet Preprocessing: The ScatNet pipeline required a different preprocessing approach. First, the images were converted to grayscale to reduce complexity while maintaining essential structural features. They were resized to 64×64 pixels, the required shape for the Scattering Transform. Using the Kymatio library's Scattering2D function with parameters J=2 (scales) and L=8 (orientations), we extracted 81 scattering coefficients per image in the shape [81, 16, 16]. We then applied mean pooling across each coefficient map to produce a fixed-length feature vector of shape [81] per image. As in the CNN pipeline, normalization and random seed setting were applied to ensure consistent and reproducible feature extraction.

3. Methodology

3.1. CNN Pipeline The CNN model consists of three convolutional layers followed by max-pooling, ReLU activation, and dropout. The network ends with two fully connected layers for binary classification. The model was trained using 5-fold cross-validation, with early stopping based on validation loss.

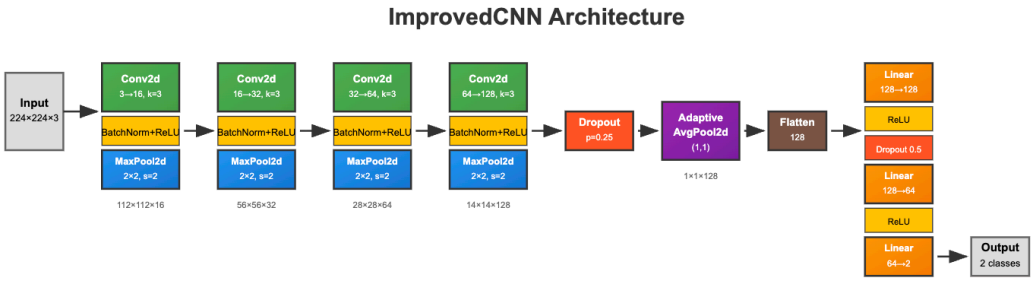

Fig 1: CNN Architecture

3.2. ScatNet Pipeline In the ScatNet pipeline, we used the scattering transform to extract fixed features from the images. These features were then pooled using global average pooling and fed into a neural network. This model was also trained using 5-fold cross-validation.

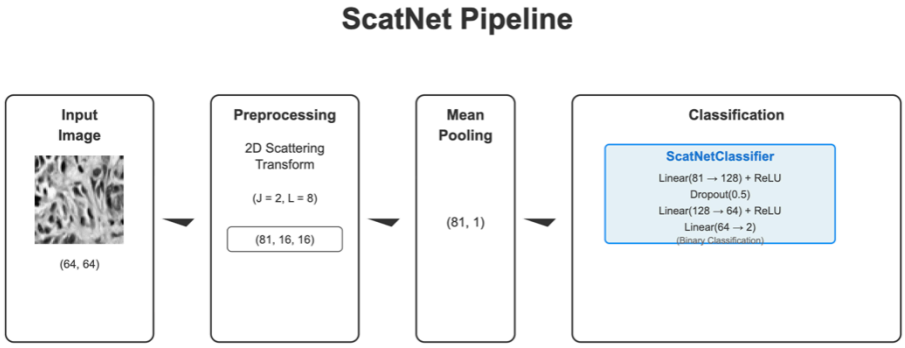

Fig 2: ScatNet Architecture

3.3. Training Both the CNN and ScatNet models were trained using 5-fold cross-validation to ensure class balance and provide a reliable estimate of generalization performance. In each fold, the dataset was split into training and validation sets, and model performance was monitored using both accuracy and weighted F1 score. Each model was trained for 10 epochs with a batch size of 64, and the model with the highest validation F1 score was saved and used for final evaluation. Importantly, although the input data and feature representations were different, both pipelines used the same classifier architecture consisting of fully connected layers. This allowed for a fair comparison between learned CNN features and handcrafted ScatNet features, isolating the impact of the feature extraction method on performance. Despite architectural differences in the input stages (convolutional vs. scattering), the classification heads were identical, ensuring consistency in evaluation.

4. Results

4.1. Learning Curves The CNN model showed excellent performance on the training set but exhibited occasional spikes in validation loss, indicating mild overfitting in some folds. In contrast, the ScatNet-based model had smoother and more consistent learning curves across folds, reflecting better generalization.

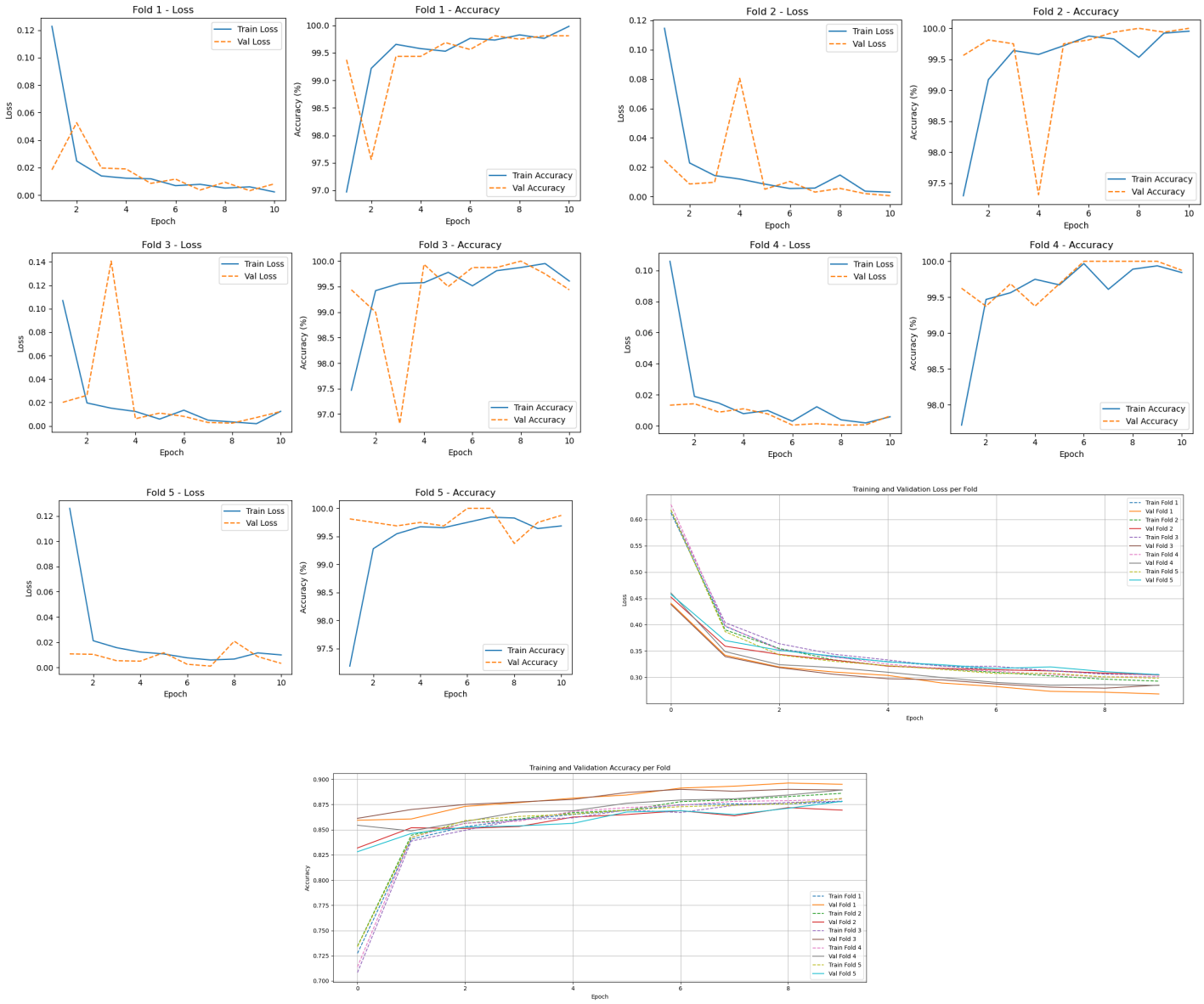

Fig 3: Learning Curves

4.2. Final Evaluation on Held-Out Test Set On the held-out test set (2000 samples), the CNN achieved perfect performance:

- Accuracy: 100%
- F1-score: 1.00

The ScatNet model also performed well but slightly below the CNN:

- Accuracy: 89.30%
- F1-score (Macro): 0.89

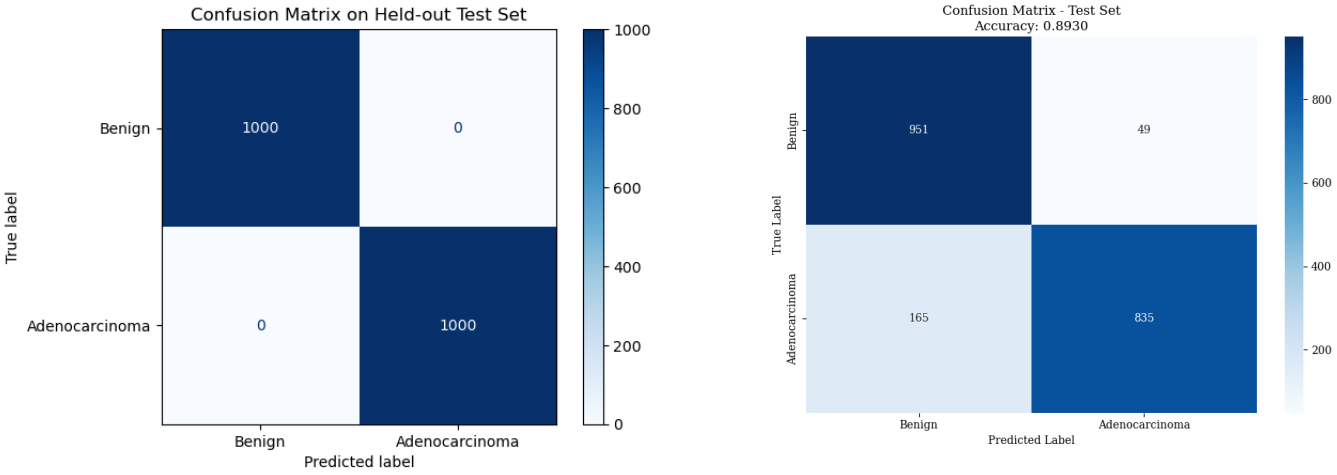

Fig 4: Confusion Matrixes

5. Discussion

5.1. Model Differences CNN learns its own filters from data, which makes it flexible and powerful but also less interpretable. ScatNet uses predefined wavelet filters, making it more transparent and mathematically explainable.

5.2. Training and Evaluation CNN achieved higher accuracy and perfect classification on the test set. However, the loss and accuracy changed more irregularly during training. ScatNet had slightly lower accuracy but more stable performance and better generalization across folds.

5.3. Filters Used in the Models CNN filters are learned and often difficult to interpret, especially in deeper layers. ScatNet uses fixed wavelet filters of known shape and scale, which makes the model more understandable. These filters help us better analyze what features are extracted.

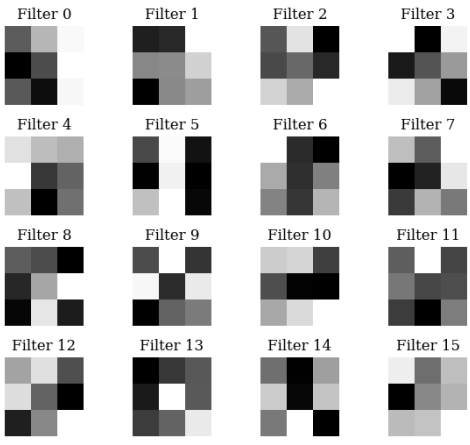

Fig 6: CNN Conv1 Filters(Channel 0)

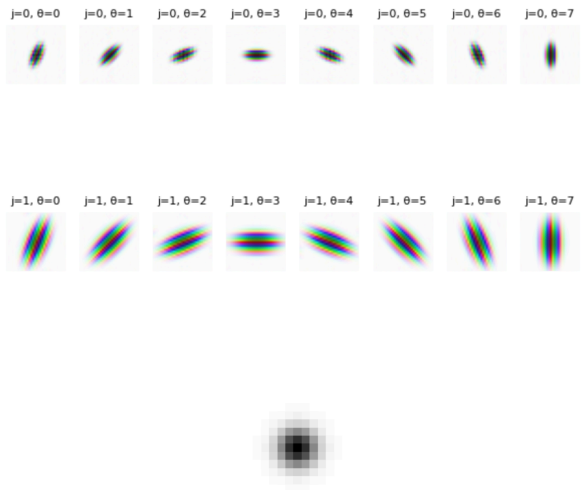

Fig 7: ScatNet Filters: Wavelets (ψ) and Low-Pass Filter (ϕ)

5.4. Explainable AI (XAI) For the CNN model, we used two Explainable AI (XAI) methods: SHAP (Shapley Additive Explanations) and Saliency Maps. With SHAP, we explained the model's predictions by masking parts of the image using a blur technique—this allowed us to estimate how much each blurred region contributed to the final classification.To gain more localized and fine-grained insights, we also applied Saliency Maps, which use gradients to highlight the most influential pixels in the image

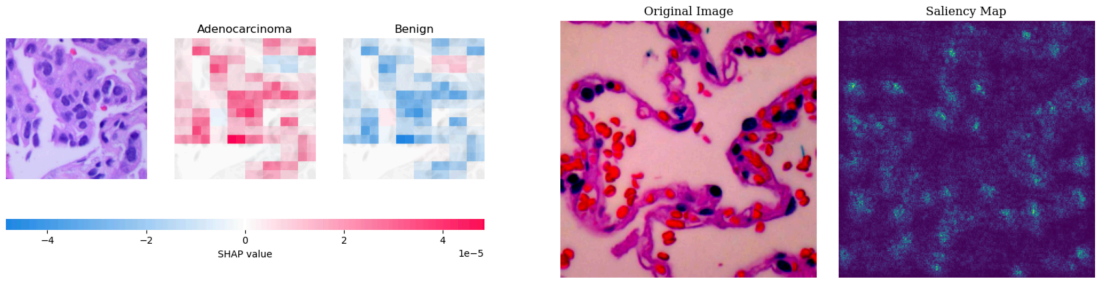

Fig 8: CNN Attribution Maps

In ScatNet, since we already have known features, we applied Shapley value sampling to determine which scattering channels were most influential. The first feature, corresponding to low-frequency averaging, had the highest importance.

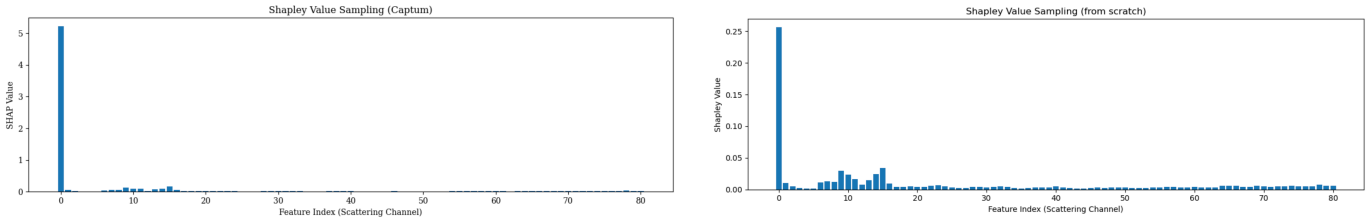

Fig 9: Scattering Feature Importance via Shapley Values

This analysis suggests that we could reduce input dimensions by removing less important features, improving efficiency without sacrificing accuracy.

6. Further Improvements Future improvements could include:

- Feature selection using SHAP scores to reduce redundancy.
- Combining mean and standard deviation pooling.
- Using deeper or attention-based classifiers.

- Creating hybrid models that combine CNN and ScatNet features for a balance of interpretability and learning power.

7. Conclusion Both CNN and ScatNet pipelines showed strong classification performance on lung histopathology images. CNN achieved perfect accuracy but lacks interpretability, while ScatNet offered a more explainable framework with slightly lower performance. Overall, ScatNet is a promising option for tasks requiring both accuracy and transparency.

8. Appendix

Dataset Source:
The histopathological lung image dataset used in this project is publicly available on Kaggle:
https://www.kaggle.com/datasets/rm1000/lung-cancer-histopathological-images

Explainable AI Reference (SHAP):
For implementing XAI and understanding Shapley value-based explanations, we referred to the official SHAP documentation:
https://shap.readthedocs.io/en/latest/index.html