DCT-Based Pathology Classification Using Random Forest and CNN

Rolensky Louis
Department of Electrical and Computer Engineering, Temple University
Rolensky.louis@temple.edu

Introduction: The goal of this project is to classify pathology image data into multiple classes using machine learning techniques. The dataset used in this project consists of samples represented by 3072 features. These features correspond to a flattened representation of a 3 by 32 by 32 structure derived from Discrete Cosine Transform (DCT) coefficients. Although the data is stored as a flat vector, it represents an image-like structure.

A major challenge in this dataset is that it is highly imbalanced and contains overlapping classes. In particular, classes 3, 5, and 6 are very difficult to separate due to their similarity. In addition, not all classes are considered equally important in the evaluation, since some classes are ignored in the scoring function.

To address these challenges, two different approaches were explored. The first approach uses Random Forest, which is a tree-based model that is robust to imbalance and capable of handling complex decision boundaries. The second approach uses a Convolutional Neural Network (CNN), which is designed to learn spatial patterns from structured data. These two approaches allow us to compare a traditional machine learning model with a deep learning model on the same dataset.

Algorithm No. 1 Description: The first algorithm used in this project is based on Random Forest. Random Forest is an ensemble learning method that builds multiple decision trees and combines their predictions. It is known for its robustness to noisy data and its ability to model non-linear relationships.

Initially, a baseline Random Forest model was implemented. This model attempted to classify all classes at once using the full dataset. However, the performance of this baseline model was poor, especially on the development set. This is mainly due to the imbalance in the dataset and the overlap between certain classes. The model struggled to correctly separate difficult classes such as 3, 5, and 6.

To improve the performance of the Random Forest model, a hierarchical structure was introduced. Instead of using a single model, the classification problem was divided into multiple stages. In the first stage, the model separates tissue from background. This simplifies the problem by removing irrelevant samples early in the process.

In the second stage, the model focuses only on tissue samples and separates class 0, class 2, and a group of disease classes. This further reduces the complexity of the classification task. In the third stage, a specialized model is used to distinguish between the most difficult classes, which are classes 3, 5, and 6.

In addition to restructuring the model, parameter tuning was also applied. In particular, the minimum number of samples per leaf was increased. This change prevents the trees from becoming too specific and helps reduce overfitting. By combining the hierarchical structure with improved parameter tuning, the Random Forest model achieved significantly better performance on the development set.

Algorithm No. 2 Description: The second algorithm used in this project is a Convolutional Neural Network (CNN). CNNs are widely used for image-related tasks because they can learn spatial patterns and hierarchical features directly from the data.

Although the dataset is stored as a flattened vector, it represents a 3 by 32 by 32 structure. Therefore, each sample is reshaped into this format before being passed into the CNN. This allows the model to process the data as if it were an image.

The CNN consists of multiple convolutional layers that extract features from the input. The first layers capture simple patterns, while deeper layers capture more complex relationships. The model also includes fully connected layers that map the extracted features to class predictions.

To improve the performance of the CNN, several techniques were applied. Batch normalization was used to stabilize the training process, and dropout was used to reduce overfitting. In addition, class weights were applied in the loss function to address the imbalance in the dataset. Early stopping was also used to prevent the model from overfitting during training.

Results: The performance of the models was evaluated using a custom scoring function. This scoring function computes the error for each important class and averages the results, with a small weight given to the background class. Lower scores indicate better performance.

The table below summarizes the results obtained for the different models.

Algorithm	Train	Dev Test	Eval
RF Baseline V1	10.89%	85.61%	
RF Baseline V2	5.01%	60.47%	
Hierarchical RF V1	42.73%	64.95%	
Hierarchical RF V2 (BEST)	26.53%	52.12%	55.63%
CNN Baseline	35.53%	62.18%	
CNN (BEST)	41.08%	50.84%	55.87%

Table 1. Comparison of model performance on the dataset. Lower values indicate better performance.

From the results, it can be observed that the CNN models perform better than the Random Forest models overall. The tuned CNN achieves strong performance, while the hierarchical Random Forest also shows significant improvement over the baseline.

Conclusions: In this project, Random Forest and CNN were used to classify pathology data. The Random Forest model initially performed poorly due to class imbalance and overlapping classes. However, introducing a hierarchical structure significantly improved its performance by simplifying the classification task.

The CNN model performed better overall, especially after tuning. The improvements in the CNN model come from better regularization and training techniques, which help the model generalize better.

The results show that both the choice of model and the way the problem is structured are important. Random Forest benefits from restructuring the problem, while CNN benefits from improved training and feature learning.

Future work could explore more advanced neural network architectures or hybrid approaches to further improve performance.