**Classification of Digital Pathology Images Using Machine Learning**

Md Abdullah Al Mamun

Department of Electrical Engineering and Computer Engineering, Temple University

abdullah18@temple.edu

**Introduction:** Digital pathology involves converting traditional glass histology slides into high-resolution digital images, enabling advancements in clinical diagnosis, research, and quantitative analysis. However, challenges such as large file sizes, tissue complexity, and computational costs persist. This paper explores the classification of digital pathology images using machine learning techniques. The dataset comprised samples represented by 3072 Discrete Cosine Transform (DCT) coefficients, categorized into nine labels. Both non-neural (XGBoost) and neural network approaches, including various Convolutional Neural Network (CNN) architectures (ResNet, MobileNet, InceptionNet, VGG16, DenseNet, EfficientNet) and Vision Transformers (ViT), were evaluated. Data preprocessing involved applying the Inverse DCT (IDCT) to reconstruct spatial domain images for neural network input. A custom weighted average error rate metric was used for evaluation, focusing on clinically relevant classes. Experiments included comparing standard and weighted loss functions to better align training with the custom evaluation metric. Cross-validation techniques were employed to assess model robustness. Results indicated varying performance across models, with pretrained networks generally showing promise, and highlighted the impact of weighting strategies on performance according to the custom metric.

**Materials and Methods**

* Dataset: The dataset used consisted of digital pathology image samples represented by 3072 Discrete Cosine Transform (DCT) coefficients each. The target variable, indicating the class label, was provided at index 0 for each instance. There were nine distinct labels: norm (0), artf (1), nneo (2), infl (3), susp (4), dcis (5), indc (6), null (7), and bckg (8). The data was divided into a training set (10,066 samples), a development set (5,958 samples), and an evaluation set (6,260 samples).
* Custom Evaluation Metric: A custom evaluation metric, the Weighted Average Error Rate (%), was defined, where lower scores indicate better performance.
* Data Preprocessing for Neural Networks: For neural network models, the input DCT coefficients needed conversion back to the spatial domain. This was achieved by applying the Inverse Discrete Cosine Transform (IDCT). Since the original features were 32x32 DCT coefficients from a 64x64 transform, the IDCT successfully recovered spatial image representations suitable for CNN and ViT input.
* Models Explored:
	+ **XGBoost:** A non-neural approach using the XGBoost classifier was tested.
	+ **Neural Networks:** A wide range of deep learning models were investigated:
		- CNNs: ResNet (trained from scratch and pretrained), MobileNet (pretrained), InceptionNet (pretrained), VGG16 (pretrained), DenseNet (pretrained), and various EfficientNet versions (B0-B7, V2S, V2L - all pretrained).
		- Transformers: Vision Transformer (ViT - base and large variants, pretrained).

* **Weighted Loss Function:** To align the training process more closely with the custom evaluation metric (which ignores certain classes), a weighted loss function was implemented. This function assigned a weight of zero to the loss contribution from samples belonging to the ignored classes (artf, susp, null) and a weight of one to all other classes relevant to the final score calculation.
* **Cross-Validation:** To evaluate model robustness and generalization, K-Fold Cross-Validation was employed. Both 5-fold and 10-fold strategies were considered, with 5-fold ultimately selected as it provided strong generalization and was deemed sufficient for the dataset size while managing computational load.

**Results and Discussion**

XGBoost: The XGBoost model, trained on down-sampled data, achieved a development set weighted error of 0.46. Using the custom evaluation metric, it scored 1.69% on the train set, 63.45% on the development set, and 61.01% on the evaluation set.

*Table 2: Results on round 2 with weighted strategy + pretrain*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Dev Error (%)** | **Label Error (%)** | **Bckg Error (%)** | **Score (%)** |
| Weighted B0 | 32.78% | 31.79% | 11.99% | 29.81% |
| Weighted B1 | 31.94% | 32.25% | 8.91% | 29.92% |
| Weighted V2S | 32.63% | 33.27% | 7.57% | 30.70% |
| Weighted V2L | 31.45% | 32.72% | 9.21% | 30.37% |
| **Weighted B7** | **32.61%** | **32.12%** | **8.76%** | **29.78%** |
| Weighted ViT-Base | 37.16% | 34.43% | 3.82% | 31.37% |

|  |  |  |
| --- | --- | --- |
| **Model** | **Train Error (%)** | **Dev Error (%)** |
| ResNet (scratch) | 17.96% | 30.75% |
| ResNet (pretrained) | 0.06% | 28.90% |
| MobileNet (pretrained) | 2.27% | 27.64% |
| InceptionNet (pretrained) | 1.24% | 29.76% |
| VGG16 (pretrained) | 20.28% | 35.55% |
| DenseNet (pretrained) | 29.83% | 31.29% |
| **EfficientNet B0 (pretrained)** | **7.06%** | **26.97%** |
| **EfficientNet B7 (pretrained)** | **20.60%** | **26.85%** |
| ViT-Base (pretrained) | 0.04% | 27.76% |
| ViT-Large (pretrained) | 14.09% | 28.26% |

*Table 1: Results on round 1 with Pretrained strategy*

|  |  |
| --- | --- |
| **Fold** | **Best Score** |
| Fold 1 | 0.2446 |
| Fold 2 | 0.2535 |
| Fold 3 | 0.2173 |
| Fold 4 | 0.2446 |
| Fold 5 | 0.2341 |
| Overall BestTable 3: 5-fold weighted b7 | 0.2173 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Dataset** | **Label Error (%)** | **Bckg Error (%)** | **Score (%)** |
| Train | 11.73% | 2.84% | 10.84% |
| Dev | 12.45% | 6.14% | 11.82% |
| Eval | 34.54% | 7.28% | 31.81% |

Table 4: Final results (weighted score)

**Neural Networks (Round 1 - Standard Loss):** Initial experiments compared various neural network architectures using a standard loss function. Table 1 summarizes the training and development set errors (%) for these models.

**Neural Networks (Round 2 - Weighted Loss):** To better align with the evaluation criteria, models were trained using a weighted loss function. Table 2 presents the development set performance, including label error (%), background error (%), and the final custom score (%).

**Cross-Validation (Round 3 - EfficientNet B7):** Cross-validation was performed to assess the robustness of the EfficientNet B7 model. Table 3 shows the validation error (%) per fold for the weighted EfficientNet B7 using 5-fold cross-validation.

The **final performance** of the selected model configuration across the training, development, and evaluation datasets is detailed in Table 4, showing the label error (%), background error (%), and the overall custom score (%). The final evaluation score of 31.81% indicates the challenging nature of the task, likely stemming from the inherent complexity of histopathological images and potential limitations of using DCT coefficients as input features compared to raw pixels.

This study demonstrated the application of various machine learning models, from XGBoost to advanced deep learning architectures like EfficientNets and Vision Transformers, for classifying digital pathology images based on DCT features. Preprocessing using IDCT enabled the use of image-based neural networks.

In conclusion, this work provides a comprehensive comparison of machine learning techniques for digital pathology image classification, emphasizing the importance of tailored evaluation metrics and loss functions. EfficientNet models demonstrated strong potential, and the methodologies explored offer valuable insights for future research in automated histopathological analysis.