Lung Cancer Detection

Eswar Naveen Teja Bojja
Department of Mathematics and Statistics
Indian Institute of Technology Kanpur
naveenteja23@iitk.ac.in

Marada Teja Satvik
Department of Computer Science and Engineering
Indian Institute of Technology Kanpur
maradateja23@iitk.ac.in

Habeeb Ramith Kumar
Department of Computer Science and Engineering
Indian Institute of Technology Kanpur
hramith23@iitk.ac.in

Pasala Bosu Akil Teja
Department of Computer Science and Engineering
Indian Institute of Technology Kanpur
pbosuateja23@iitk.ac.in

Shashi Bhidodiya
Department of Computer Science and Engineering
Indian Institute of Technology Kanpur
shashib23@iitk.ac.in

***Abstract*—This paper presents a custom convolutional neural network (CNN) designed to detect and classify lung cancer from CT scan images. The network distinguishes among four classes: adenocarcinoma, large cell carcinoma, squamous cell carcinoma, and normal lung tissue. The dataset is divided into a training set of 612 images and a validation set of 72 images, for a total of 684 labeled images. Our approach involves comprehensive data preprocessing, advanced augmentation techniques, and class weighting to mitigate data imbalance. Cross-validation demonstrates robust performance, and the final training on the entire dataset achieves a training accuracy of 96.07%. The results highlight the effectiveness of the proposed model in helping early detection of lung cancer.**

***Index Terms*—Lung Cancer Detection, Convolutional Neural Network, CT Scan, Data Augmentation, Class Imbalance, Early Stopping.**

I. Introduction

Lung cancer is one of the leading causes of cancer-related deaths worldwide. Early detection through imaging modalities such as CT scans significantly improves treatment outcomes. However, manual examination of CT images can be time consuming and prone to differences in interpretation between doctors. In this work, we propose a CNN-based solution that classifies CT scan images into four categories: (1) adenocarcinoma, (2) large cell carcinoma, (3) squamous cell carcinoma, and (4) normal (non-cancerous) tissue. Our methodology emphasizes detailed preprocessing, data augmentation, and training strategies, with an aim to address challenges like data imbalance and overfitting.

II. Data Preprocessing and Augmentation

A. Dataset Overview

The dataset comprises 684 CT scan images. These images are split into a training set of 612 images and a validation set of 72 images. Each image is associated with one of four classes:

- **Adenocarcinoma:** 194 images
- **Large Cell Carcinoma:** 115 images
- **Normal:** 148 images
- **Squamous Cell Carcinoma:** 155 images

B. Preprocessing Steps

1) **Label Encoding:** Images are stored in directories named after their respective classes. Each class name is mapped to an integer label.
2) **Image Resizing and Grayscale Conversion:** All images are resized to 256×256 pixels and converted to a single channel to maintain grayscale intensity information.
3) **Normalization:** Pixel values are scaled to the range [0, 1] to stabilize training and ensure consistent input magnitudes.

C. Data Augmentation

To enrich training data and reduce overfitting, we apply:

- **Random Flip:** Horizontal flips.
- **Random Rotation:** Up to 5% rotation.
- **Random Zoom:** Up to 5% zoom.
- **Random Contrast:** Up to 5% adjustment.

These augmentations enable the model to learn robust features that are not affected by minor spatial and intensity changes.

D. Class Imbalance Handling

The four classes are not evenly represented, with large cell carcinoma having the fewest images (115). To ensure balanced training, class weights are applied:

- Adenocarcinoma = 1.0
- Large Cell Carcinoma = 2.0
- Normal = 1.5
- Squamous Cell Carcinoma = 1.4

This approach increases the loss contribution from underrepresented classes, encouraging the model to pay more attention to them.

III. Model Architecture

The proposed CNN is developed using TensorFlow/Keras and follows a sequential design:

A. Convolutional and Pooling Layers

- **Conv Layer 1:** 32 filters, 3×3 kernel, ReLU activation
- **MaxPooling2D:** 2×2 pool size, reducing the spatial dimensions by half (e.g., 256×256 to 128×128).
- **Conv Layer 2:** 64 filters, 3×3 kernel, ReLU activation
- **MaxPooling2D:** Another 2×2 pool, further halving dimensions (e.g., 128×128 to 64×64).
- **Conv Layer 3:** 128 filters, 3×3 kernel, ReLU activation
- **MaxPooling2D:** 2×2 pool, reducing dimensions again (e.g., 64×64 to 32×32).

By stacking these convolutional blocks, the network progressively extracts more abstract and higher-level features from the input images.

B. Fully Connected Layers

After the final pooling layer, the feature maps are flattened into a one-dimensional vector. This vector is fed into:

- **Dense Layer:** 128 neurons with ReLU activation to learn complex patterns.
- **Dropout:** A dropout rate of 0.3 to mitigate overfitting by randomly deactivating neurons during training.
- **Output Layer:** A dense layer with 4 neurons (softmax activation) that outputs the probability distribution over the four classes.

C. Compilation

The network is compiled using:

- **Optimizer:** Adam with a learning rate of 3×10^{-4}.
- **Loss Function:** Sparse categorical cross-entropy, suitable for integer-based labels.
- **Metrics:** Accuracy for monitoring performance.

IV. Training Strategy and Hyperparameter Tuning

A. Cross-Validation

We use 5-fold cross-validation to tune hyperparameters and assess stability. Each fold uses one subset for validation and the remaining for training, providing a robust generalization estimate.

B. Callbacks

During training, the following callbacks are employed:

- **Early Stopping:** Stops training if validation loss shows no improvement for **5** epochs, restoring the best weights.
- **ReduceLROnPlateau:** Reduces the learning rate by 0.5 if no validation loss improvement occurs for **3** epochs, with a floor of 1×10^{-7}.

C. Final Training

After tuning hyperparameters and confirming model stability via cross-validation, the final model is retrained on the entire training set of 612 images (with augmentation). This process is run for up to 60 epochs or until early stopping criteria are met. The model achieves a training accuracy of **96.07%** on the entire dataset.

V. Experimental Results

A. Cross-Validation Performance

The validation accuracies from 5-fold cross-validation are:

- Fold 1: 87.80%
- Fold 2: 91.87%
- Fold 3: 92.62%
- Fold 4: 88.52%
- Fold 5: 91.80%

The average cross-validation accuracy is approximately 90.53%.

B. Evaluation on Held-Out Validation Set

On a held-out validation set of 72 images, the final model achieves:

- **Accuracy:** 88.89%
- **Precision:** 88.98%
- **Recall:** 88.89%
- **F1-Score:** 88.81%

These metrics demonstrate balanced performance across classes.

C. Class-Specific Performance

- **Adenocarcinoma:** Precision = 0.86, Recall = 0.78, F1 = 0.82 (Support: 23)
- **Large Cell Carcinoma:** Precision = 0.90, Recall = 0.90, F1 = 0.90 (Support: 21)
- **Normal:** Perfect classification with scores of 1.00 (Support: 13)
- **Squamous Cell Carcinoma:** Precision = 0.82, Recall = 0.93, F1 = 0.88 (Support: 15)

While the model is generally robust, slight variability in performance indicates opportunities for further optimization, especially for adenocarcinoma.

VI. Discussion

Our CNN-based approach accurately classifies lung CT images into four categories. Data augmentation and class weighting mitigate imbalance, while cross-validation, early stopping, and adaptive learning rate reduction promote stable convergence and prevent overfitting. Performance may still improve with more data or advanced architectures.

VII. Conclusion and Future Directions

This study demonstrates that a carefully designed CNN reliably detects and classifies lung cancer types from CT scans, achieving a training accuracy of 96.07

A. Future Directions

- **Larger, Diverse Datasets:** Incorporate more CT scans from various institutions to boost generalizability.
- **Advanced Architectures:** Explore deeper or specialized models (e.g., ResNet, DenseNet, attention-based).
- **Transfer Learning:** Use pretrained models on large-scale medical imaging data to enhance accuracy.
- **Hyperparameter Tuning:** Optimize settings like dropout rate and batch size to improve class-specific performance.

References

[1] Wang, L. (2022). Deep Learning Techniques to Diagnose Lung Cancer. *Cancers*, 14(22), 5569. https://doi.org/10.3390/cancers14225569

[2] Shah, A. A., Malik, H. A. M., Muhammad, A., Alourani, A., & Butt, Z. A. (2023). Deep Learning Ensemble 2D CNN Approach Towards the Detection of Lung Cancer. *Scientific Reports*, 13, 2987. https://doi.org/10.1038/s41598-023-29656-z