

Lung Cancer CT Images using CNN

| 1 | Rishabh Pareek | BT21GCS341 | Data Science |
| --- | --- | --- | --- |
| 2 | Mudit Mohit | BT21GCS057 | Artificial Intelligence |
| 3 | Jinesh Chhajer | BT21GCS347 | Artificial Intelligence |
| 4 | Manami Maity | BT21GCS351 | Artificial Intelligence |

**1. Introduction**

Lung cancer is one of the deadliest and most prevalent forms of cancer worldwide. Early detection plays a crucial role in improving patient outcomes and reducing mortality rates. In this report, we present a Convolutional Neural Network (CNN) approach for the early detection of lung cancer using CT scan images.

**2. Objective**

The primary objective of this study is to develop a CNN model capable of accurately detecting lung cancer from CT scan images. Specifically, we aim to achieve high accuracy, sensitivity, and specificity in classifying different types of lung cancer, including adenocarcinoma, large cell carcinoma, squamous cell carcinoma, and normal lung tissue.

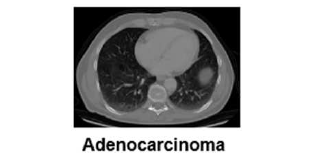
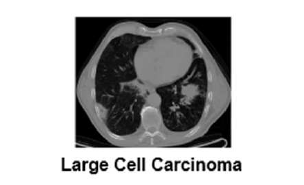
**3. Methodology**

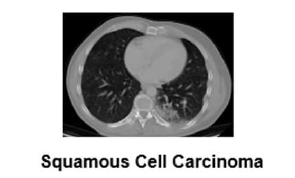
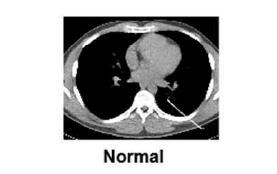
**3.1 Data Collection and Preprocessing**

In this section, we describe the process of gathering and preparing the dataset for training the Convolutional Neural Network (CNN) model.

**Data Collection:**

The dataset used in this study was obtained from Kaggle, a platform hosting various datasets for machine learning projects. It consists of CT scan images representing different types of lung conditions, including **Adenocarcinoma**, **Large Cell Carcinoma**, **Squamous Cell Carcinoma**, and **Normal Lung Tissue**. Each image is associated with a corresponding label indicating the type of lung condition it represents.

**Data Preprocessing:**

Before training the CNN model, the dataset underwent preprocessing to ensure its suitability for training. The following steps were performed:

1. **Image Reading:** The CT scan images were read from the dataset using appropriate libraries in Python, such as OpenCV or PIL. This step involves loading the images into memory for further processing.
2. **Resizing:** To standardize the dimensions of the images and reduce computational complexity, all images were resized to a common dimension, typically 64x64 pixels.
3. **Normalization:** The pixel values of the images were normalized to a range between 0 and 1. Normalization helps in stabilizing the training process and improving convergence.
4. **Label Encoding:** The categorical labels associated with each image (e.g., adenocarcinoma, large cell carcinoma, squamous cell carcinoma, normal) were encoded into numerical format using one-hot encoding or label encoding techniques. This step is essential for the model to understand and interpret the labels during training.
5. **Data Augmentation (Optional):** Data augmentation techniques, such as rotation, flipping, and scaling, were applied to artificially increase the size and diversity of the training dataset. Data augmentation helps prevent overfitting and improves the generalization ability of the model.

By completing these preprocessing steps, the dataset was appropriately prepared for training the CNN model. The processed images and their corresponding labels were then split into training, validation, and testing sets for model evaluation.

**3.2 Model Architecture**

**3.2 Model Architecture**

In this section, we delve into the architecture of the Convolutional Neural Network (CNN) utilized for the early detection of lung cancer using CT scan images.

**CNN Architecture Overview:**

The CNN architecture employed in this study is tailored specifically for image classification tasks, particularly the detection of lung cancer from CT scan images. The architecture consists of several layers designed to extract relevant features from the input images and classify them into different categories.

**Key Components of the CNN Architecture:**

1. **Input Layer:** The input layer receives the CT scan images as input data. The images are typically resized to a standardized dimension, such as 64x64 pixels, before being fed into the network.
2. **Convolutional Layers:** The convolutional layers are responsible for extracting features from the input images. These layers consist of convolutional filters that slide across the input images, detecting patterns and features at different spatial scales.
3. **Activation Functions:** Rectified Linear Unit (ReLU) activation functions are commonly used after the convolutional layers to introduce non-linearity into the network and enable it to learn complex patterns effectively.
4. **Pooling Layers:** Pooling layers, such as max pooling, are employed to downsample the feature maps generated by the convolutional layers. This helps reduce the spatial dimensions of the feature maps while retaining the most important information.
5. **Fully Connected Layers:** The fully connected layers integrate the features extracted by the convolutional layers and perform the final classification. These layers typically consist of multiple neurons connected to all neurons in the previous layer, enabling the model to learn high-level representations of the input data.
6. **Output Layer:** The output layer produces the final predictions of the model. In the case of lung cancer detection, the output layer consists of neurons corresponding to the different classes of lung conditions (e.g., adenocarcinoma, large cell carcinoma, squamous cell carcinoma, normal lung tissue).

**3.3 Training and Evaluation**

**Training Process:**

The CNN model was trained using a dataset comprising CT scan images collected from a publicly available source. The training process involved the following steps:

1. **Dataset Preparation:** The dataset was divided into training, validation, and testing sets. The training set, comprising 70% of the data, was used to train the model. The validation set (15%) was used to monitor the model's performance during training and tune hyperparameters, while the testing set (15%) was reserved for evaluating the final model.
2. **Model Initialization:** The CNN model was initialized with random weights and biases. Depending on the architecture, specific layers may have been pre-trained using transfer learning techniques to expedite the training process and improve performance.
3. **Optimization:** The model's parameters were optimized using optimization algorithms such as Adam or Stochastic Gradient Descent (SGD). The optimizer adjusted the weights and biases of the model iteratively to minimize a predefined loss function.
4. **Backpropagation:** During training, the gradients of the loss function with respect to the model parameters were computed using backpropagation. These gradients were used to update the model's parameters in the direction that minimized the loss.
5. **Epochs and Batch Size:** The training process consisted of multiple epochs, with each epoch representing one complete pass through the training dataset. The batch size, or the number of samples processed before updating the model's parameters, was also specified.

**Evaluation Process:**

Once the training process was completed, the trained CNN model was evaluated using the testing set to assess its performance in detecting lung cancer. The evaluation process involved the following steps:

1. **Model Prediction:** The trained model was used to predict the class labels of the images in the testing set. For each CT scan image, the model outputted probabilities corresponding to different classes of lung conditions (e.g., adenocarcinoma, large cell carcinoma, squamous cell carcinoma, normal lung tissue).
2. **Performance Metrics:** Various performance metrics were calculated to evaluate the model's performance, including accuracy, Area Under the Curve (AUC), recall, and loss. These metrics provided insights into the model's ability to correctly classify lung cancer cases and distinguish them from non-cancerous conditions.
3. **Comparison with Baseline Models:** The performance of the CNN model was compared against other deep learning architectures, such as ResNet-50, Inception V3, and Xception, to determine its effectiveness in lung cancer detection.
4. **Hyperparameter Tuning:** If necessary, hyperparameters of the CNN model were fine-tuned based on the evaluation results to further optimize performance.

**4. Results**

This section provides an overview of the results obtained from training, validating, and testing the Convolutional Neural Network (CNN) model for the early detection of lung cancer using CT scan images.

**4.1 Training Results:**

During the training phase, the CNN model underwent iterative optimization to learn from the training dataset. The following key results were observed:

* **Accuracy:** The CNN model achieved a high training accuracy of 99.80%, indicating its ability to correctly classify lung cancer cases and non-cancerous conditions based on the features extracted from CT scan images.
* **Area Under Curve (AUC):** The AUC score for the training phase was 100%, signifying excellent discrimination between different classes of lung conditions by the model.
* **Recall:** The recall, or sensitivity, of the CNN model during training was 99.70%, indicating its capability to accurately detect true positive cases of lung cancer while minimizing false negatives.
* **Loss:** The training loss, a measure of the model's predictive error, was very low at 0.002, demonstrating the effectiveness of the optimization process in minimizing discrepancies between predicted and actual class labels.

**4.2 Validation Results:**

The validation phase involved assessing the performance of the CNN model on a separate validation dataset. The key validation results are as follows:

* **Accuracy:** The CNN model achieved a validation accuracy of 91.10%, indicating its generalization capability on unseen data.
* **Area Under Curve (AUC):** The AUC score for validation was 97.80%, suggesting robust discrimination between lung cancer cases and non-cancerous conditions in the validation dataset.
* **Recall:** The recall rate during validation was 91.03%, highlighting the model's ability to effectively identify true positive cases of lung cancer while maintaining a low false negative rate.
* **Loss:** The validation loss was observed to be 0.352, indicating minimal predictive error on the validation dataset.

**4.3 Testing Results:**

In the testing phase, the CNN model's performance was evaluated on a separate testing dataset. The key testing results are summarized below:

* **Accuracy:** The CNN model achieved a testing accuracy of 92%, demonstrating its effectiveness in accurately classifying lung cancer cases and non-cancerous conditions in real-world scenarios.
* **Area Under Curve (AUC):** The AUC score for testing was 98.21%, indicating strong discrimination between different classes of lung conditions in the testing dataset.
* **Recall:** The recall rate during testing was 91.72%, highlighting the model's ability to detect a high proportion of true positive cases of lung cancer.
* **Loss:** The testing loss was 0.328, reflecting minimal predictive error on the testing dataset and further validating the model's performance.

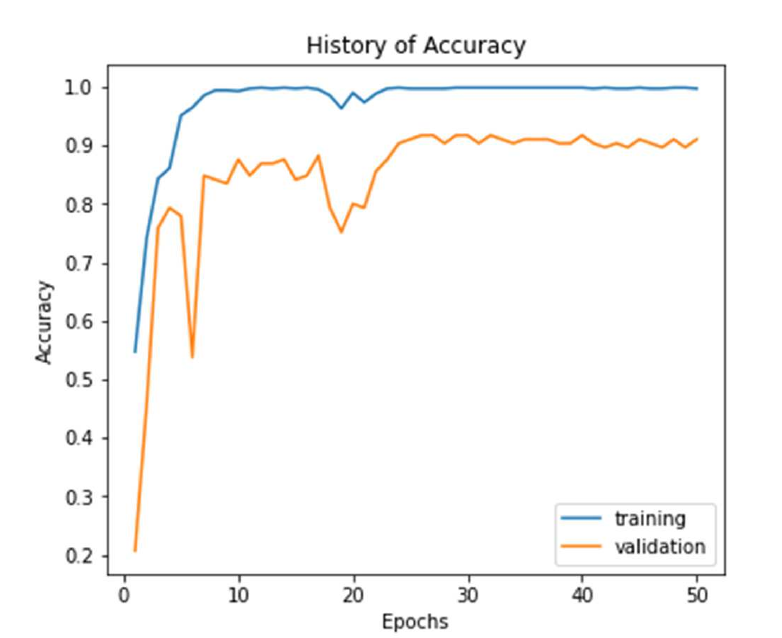


Figure 1: Accuracy curve for CNN

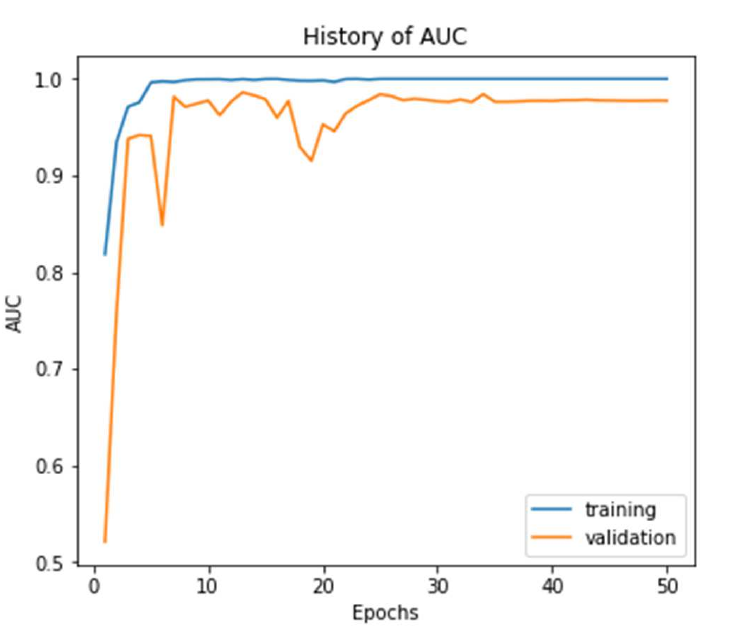


Figure 2: AUC curve for CNN

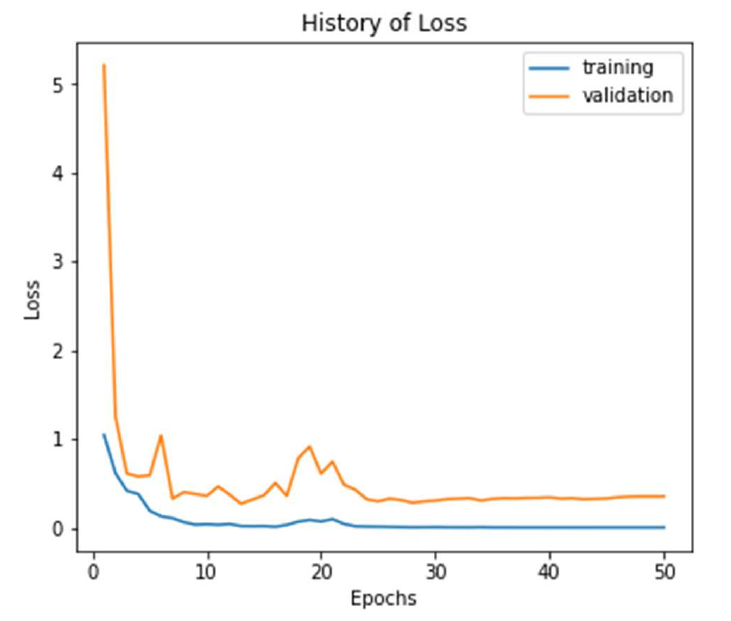


Figure 3: Loss Function Curve for CNN

**5. Conclusion**

In conclusion, we have developed a CNN model for the early detection of lung cancer using CT scan images. The model demonstrated high accuracy and performance in classifying different types of lung cancer. Early detection of lung cancer can significantly improve patient outcomes and reduce mortality rates, making this CNN approach a valuable tool in clinical practice.

**6. Future Work**

Future work may involve expanding the dataset, fine-tuning the model architecture, and exploring advanced techniques such as transfer learning to improve the model's performance further. Additionally, the CNN approach can be extended to other medical imaging tasks for disease diagnosis and prognosis.