Lung Cancer Detection Using CNN

1. INTRODUCTION

Lung cancer is a significant health concern globally, with high mortality rates and a growing burden on healthcare systems. Early detection plays a crucial role in improving patient outcomes by enabling timely intervention and treatment. Conventional diagnostic methods often rely on invasive procedures and may not always detect cancer at its early stages. Therefore, there is a pressing need for non-invasive and accurate diagnostic tools for lung cancer detection. In this project deep learning techniques are used to develop a model that can automatically detect the three most common lung cancers, Lung Adenocarcinoma, Lung Squamous cell Carcinoma, and Lung Benign tissue, from medical images, offering a potential solution to improve early diagnosis and patient care.

2. DATA PREPARATION

The dataset used in this project comprises data samples taken from the Lung and Colon Cancer Histopathological Images dataset from Kaggle. Only lung cancer images were taken. This dataset consisted of 5000 images of each of the three types of cancers, with a total of 15000 samples in the dataset. The dataset was shortened by considering only the first 500 images from each lung cancer class. Before feeding the data into the model, preprocessing steps are applied to standardize the images, including rescaling them to a uniform size (between 0 and 1). Then the dataset is split into training, validation, and test sets to facilitate model training, validation, and evaluation.

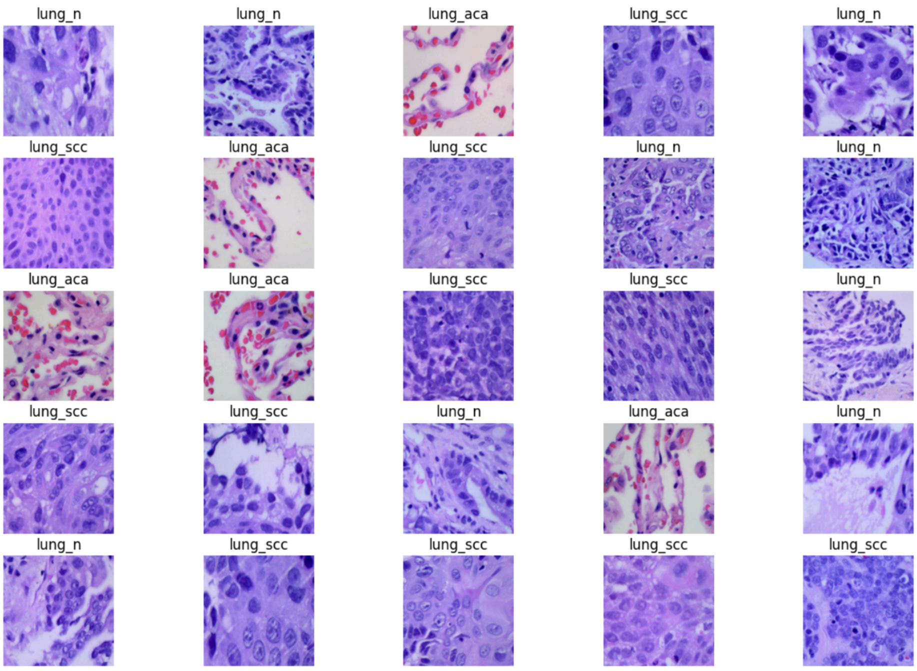

Fig 1. The lung cancer dataset

The dataset provides a diverse collection of lung cancer images. Each image is labeled with the corresponding diagnosis or condition, such as benign nodules named ‘lung_n’, Squamous Cell Carcinoma tumors named ‘lung_scc’, or Adenocarcinoma named ‘lung_aca’ lung tissue.

3. MODEL ARCHITECTURE

The proposed model architecture is based on Convolutional Neural Networks (CNNs), a class of deep learning models well-suited for image classification tasks. The CNN architecture comprises multiple layers, including four convolutional layers, Maxpooling layers between each layer, and four fully connected layers.

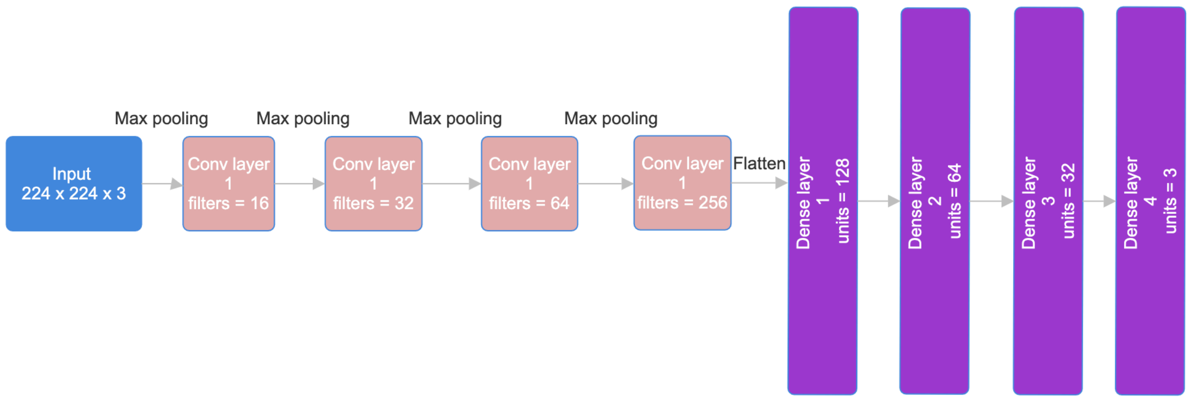

Fig 2. The proposed CNN Model.

Convolutional layers extract features from the input images by applying convolution operations, while pooling layers downsample the feature maps to reduce computational complexity and improve translation invariance. Each Convolutional layer uses ‘ReLU’ as its activation function, with padding of ‘same’ and 3 x 3 kernels. The fully connected layers at the end of the network aggregate the extracted features and perform the final classification. Overfitting is avoided by implementing a Dropout layer between each Dense layer. The output layer uses ‘Softmax’ activation at the end.

4. MODEL TRAINING

The training process involves feeding the preprocessed training data into the model and iteratively updating its weights and biases to minimize a predefined loss function. Convolutional Neural Networks (CNNs) train on image data by progressively learning to extract hierarchical features from input images. Through convolutional layers, the network applies learnable filters to detect local patterns like edges and textures. Pooling layers downsample the feature maps, preserving essential information while reducing spatial dimensions. Activation functions introduce non-linearity, enabling the network to learn complex relationships. Flattening and fully connected layers transform the features into a one-dimensional vector, which is processed by dense layers for high-level feature learning. The output layer uses softmax activation and produces probability scores for each class. During training, the network optimizes its parameters using backpropagation and optimization algorithms to minimize a chosen loss function, iteratively improving its ability to make accurate predictions. During training, the model learns to distinguish between the three different classes of lung images by adjusting its parameters. Hyperparameters such as learning rate, batch size, and number of epochs are tuned to optimize the model's performance.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 rescaling (Rescaling)       (None, 224, 224, 3)       0

 conv2d (Conv2D)             (None, 224, 224, 16)      448

 max_pooling2d (MaxPooling2  (None, 112, 112, 16)      0
 D)

 conv2d_1 (Conv2D)           (None, 112, 112, 32)      4640

 max_pooling2d_1 (MaxPoolin  (None, 56, 56, 32)        0
 g2D)

 conv2d_2 (Conv2D)           (None, 56, 56, 64)        18496

 max_pooling2d_2 (MaxPoolin  (None, 28, 28, 64)        0
 g2D)

 conv2d_3 (Conv2D)           (None, 28, 28, 256)       147712

 max_pooling2d_3 (MaxPoolin  (None, 14, 14, 256)       0
 g2D)

 flatten (Flatten)           (None, 50176)             0

 dense (Dense)               (None, 128)               6422656

 dropout (Dropout)           (None, 128)               0

 dense_1 (Dense)             (None, 64)                8256

 dropout_1 (Dropout)         (None, 64)                0

 dense_2 (Dense)             (None, 32)                2080

 dropout_2 (Dropout)         (None, 32)                0

 dense_3 (Dense)             (None, 3)                 99

=================================================================
Total params: 6604387 (25.19 MB)
Trainable params: 6604387 (25.19 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Fig 3. The total Trainable and Non-trainable parameters

5. MODEL EVALUATION

Once the model is trained, it is evaluated using the validation and test dataset to assess its performance on unseen data. Evaluation metrics such as accuracy are computed to quantify the model's classification performance.

```
47/47 [==============================] - 38s 769ms/step - loss: 0.2163 - accuracy: 0.9107
Test Loss: 0.21626800298690796
Test Accuracy: 0.9106666445732117
```

Fig 4. The test accuracy after the evaluation of the model.

This model after evaluating the test data gave an accuracy of 91.06%.

6. RESULTS

The trained model demonstrates promising results in detecting lung cancer from medical images, achieving high accuracy and robustness on both the validation and test datasets. The training accuracy achieved is 89.58% and the test accuracy achieved is 91.06%. The model's performance is evaluated across different metrics, highlighting its strengths and potential limitations.

Fig 5. Plotting the Training and Validation loss and accuracy.

Analysis of the confusion matrix reveals the model's ability to discriminate between different classes of lung images and identify areas of confusion or misclassification.

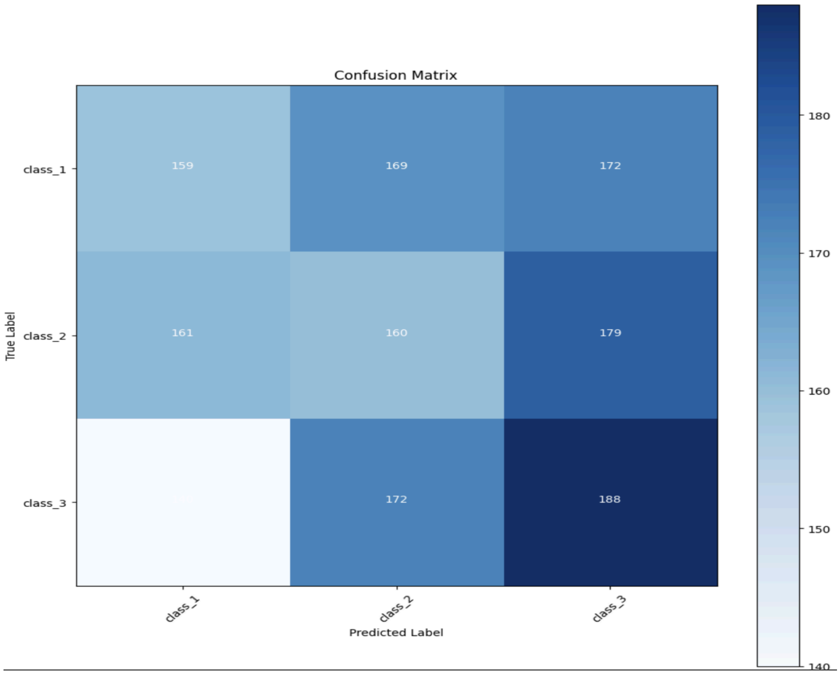

Fig 6. Confusion Matrix

Overall, the results suggest that the developed model holds promise as a reliable tool for lung cancer detection, with potential applications in clinical practice and medical research.

7. CONCLUSION

In conclusion, the project presents a comprehensive approach to lung cancer detection using deep learning techniques. By leveraging CNN and a diverse dataset of lung images, a model is developed capable of accurately identifying signs of lung cancer from medical images. The model's performance has been evaluated and validated, demonstrating its effectiveness in detecting lung cancer with a good accuracy of 91%. Moving forward, further research and development efforts can focus on enhancing the model's performance, scalability, and interpretability, ultimately contributing to improved diagnosis, treatment, and management of lung cancer patients.