Lung Cancer Detection System Documentation

Ajit Mukund Joshi

August 24, 2025

1 Project Overview

Project Name: Lung Cancer Detection System
Developer: Ajit Mukund Joshi
Version: 1.0
Description: A CNN-based system for classifying lung cancer from CT scans, achieving 95% accuracy. Developed during a Machine Learning internship at Unified Mentor Pvt. Ltd.

2 Aim

The aim is to build an automated system for early detection of lung cancer from CT images, classifying them as benign, malignant, or normal to aid radiology and improve patient outcomes.

3 Summary

This project involves a convolutional neural network (CNN) for medical image analysis:

- **Performance Metrics:** 95% classification accuracy.
- **Key Features:** Image preprocessing, CNN for feature extraction, prediction on new scans.
- **Dataset:** IQ-OTH/NCCD Lung Cancer Dataset.
- **Integration:** TensorFlow, Keras, and OpenCV.

4 Technology Used

- **Programming Language:** Python 3.x
- **Libraries and Frameworks:**

  - TensorFlow, Keras: For CNN training.
  - OpenCV: For image processing.
  - NumPy: For array manipulations.
  - Scikit-learn: For metrics.
- **Tools:** Jupyter Notebook or PyCharm.
- **Other:** ImageDataGenerator for augmentation.

5 Methodology

1. **Data Preparation:** Load and augment dataset.
2. **Model Architecture:** Sequential CNN with Conv2D, MaxPooling, Dense layers.
3. **Training:** Adam optimizer, categorical crossentropy, 10 epochs.
4. **Prediction:** Load model, preprocess, predict class.

6 Source Code

6.1 requirements.txt

```
tensorflow
keras
opencv-python
numpy
scikit-learn
```

6.2 train.py

```
import tensorflow as tf
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten,
   Dense
import numpy as np

train_datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2,
   zoom_range=0.2, horizontal_flip=True)
train_generator = train_datagen.flow_from_directory('data/',
   target_size=(150, 150), batch_size=32, class_mode='categorical')

model = Sequential([
    Conv2D(32, (3,3), activation='relu', input_shape=(150, 150, 3)),
    MaxPooling2D(2,2),
```

```
    Conv2D(64, (3,3), activation='relu'),
    MaxPooling2D(2,2),
    Flatten(),
    Dense(128, activation='relu'),
    Dense(3, activation='softmax')  # 3 classes: benign, malignant,
       normal
])

model.compile(optimizer='adam', loss='categorical_crossentropy',
   metrics=['accuracy'])
model.fit(train_generator, epochs=10)
model.save('lung_model.h5')
```

6.3 predict.py

```
import tensorflow as tf
import cv2
import numpy as np
import argparse

parser = argparse.ArgumentParser()
parser.add_argument('--image', required=True)
args = parser.parse_args()

model = tf.keras.models.load_model('lung_model.h5')
img = cv2.imread(args.image)
img = cv2.resize(img, (150, 150))
img = np.expand_dims(img / 255.0, axis=0)
pred = model.predict(img)
classes = ['benign', 'malignant', 'normal']
print(f"Prediction: {classes[np.argmax(pred)]}")
```

7 Installation and Setup

1. Download dataset to `data/`.
2. Install dependencies: `pip install -r requirements.txt`.
3. Train: `python train.py`.
4. Predict: `python predict.py –image path/to/image.jpg`.

8 Testing and Validation

- **Test Cases:** Validation on hold-out set, confusion matrix.
- **Metrics:** Accuracy (95%), Precision, Recall, F1-Score.
- **Validation Dataset:** Split from training data.

9 Conclusion

The Lung Cancer Detection System provides a reliable tool for medical image classification, with potential for integration into diagnostic workflows. Future work could include transfer learning with ResNet and web app deployment.

10 References

- TensorFlow Documentation
- Keras API Guides
- Resume: Ajit Mukund Joshi (Experience Section)