Report: Chest CT-Scan Images Dataset

> ## A Data Science Tools Project By:
>
> | Name | ID |
> |---|---|
> | عبدالله اسامه جابر منصور | 20221460119 |
> | محمد على محمد عبد العزيز | 20221459892 |
> | احلام محمد مصطفى محمد | 20221461977 |
> | آلاء عادل عبد الحميد عقيلى | 20221441500 |
> | نورهان محمد صلاح الدين علي | 210101068 |
> | يارا حاتم ابراهيم | 20221464983 |

Table of Contents

1: Introduction

Data Source: https://www.kaggle.com/datasets/mohamedhanyyy/chest-ctscan-images/

About Dataset

The used dataset is an image dataset for Chest Cancer Detection. It should help in classifying and diagnosing if the patient have cancer or not through a CT-Scan. The data consists of a CT-Scan for a patient with no cancer and 3 other types of Chest Cancer.

These types are: Adenocarcinoma, Large cell carcinoma, Squamous cell carcinoma

1. **Adenocarcinoma**

Adenocarcinoma of the lung: Lung adenocarcinoma is the most common form of lung cancer accounting for 30 percent of all cases overall and about 40 percent of all non-small cell lung cancer occurrences. Adenocarcinomas are found in several common cancers, including breast, prostate and colorectal. Adenocarcinomas of the lung are found in the outer region of the lung in glands that secrete mucus and help us breathe.

Symptoms include coughing, hoarseness, weight loss and weakness.

2. **Large cell carcinoma**

Large-cell undifferentiated carcinoma: Large-cell undifferentiated carcinoma lung cancer grows and spreads quickly and can be found anywhere in the lung. This type of lung cancer usually accounts for 10 to 15 percent of all cases of NSCLC.

Large-cell undifferentiated carcinoma tends to grow and spread quickly.

3. **Squamous cell carcinoma**

Squamous cell: This type of lung cancer is found centrally in the lung, where the larger bronchi join the trachea to the lung, or in one of the main airway branches. Squamous cell lung cancer is responsible for about 30 percent of all non-small cell lung cancers, and is generally linked to smoking.

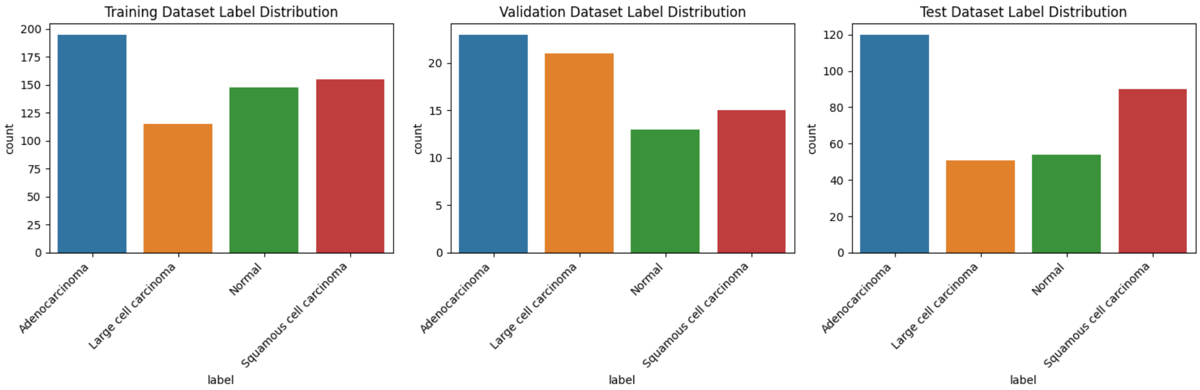
Exploratory Data Analysis

The data is consists of 3 directories (train, test & valid) each has 4 folders (one for the normal CT-Scan and 3 for the 3 cancer types). Total image files are 1000 images. Split into Training, Testing and Validation in 70%, 20% and 10% respectively.

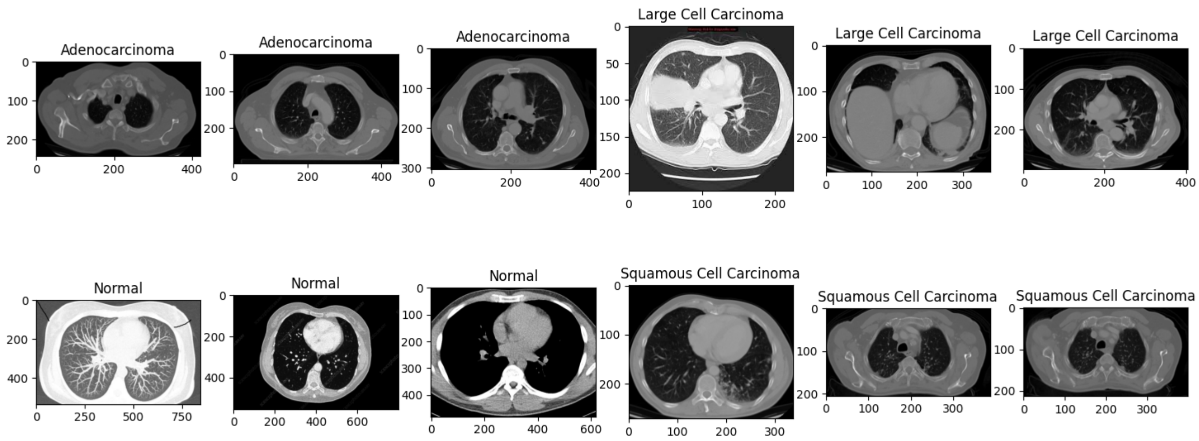
The following table shows the number of sample of each class in each folder.

Label/ Folder	**Train**	**Test**	**Valid**
Normal	148	54	13
Adenocarcinoma	195	120	23
Large cell carcinoma	115	51	21
Squamous cell carcinoma	155	90	15
Total	**613**	**315**	**72**

Which is also represented through this graph.



And this is a random sample from each label.



> **A problem we faced in preprocessing:**
> All the three folders of train, test, valid has the same subdirectories names except the test folder. They are all classified into the same 4 labels but with different naming.
> Solution: was to create a specific list of subdirectories for the test folder.

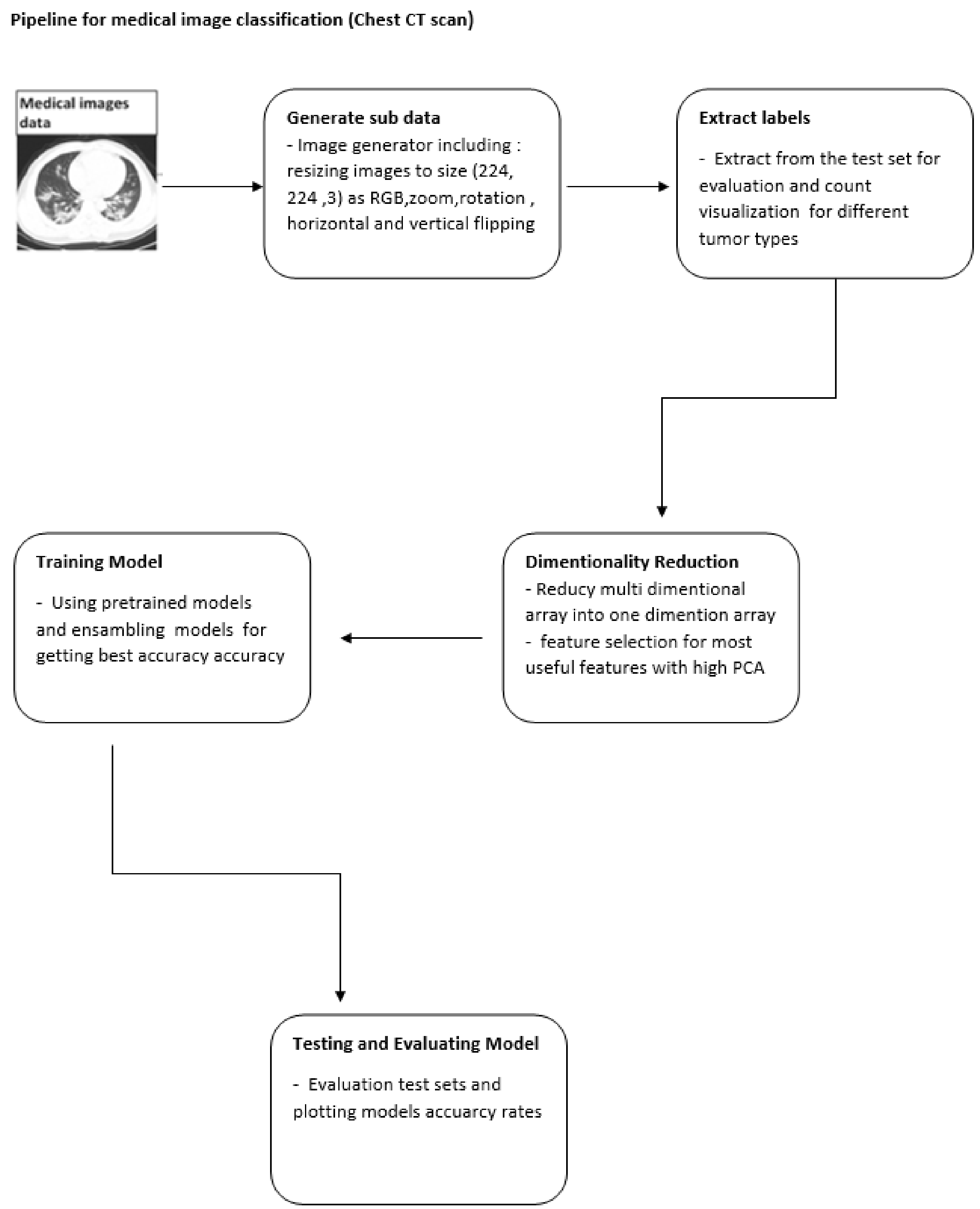
```
subdirectories = [
  "adenocarcinoma_left.lower
  "large.cell.carcinoma_left
  "normal",
  "squamous.cell.carcinoma_l
];
```

```
subdirectories_test = [

  "adenocarcinoma",
  "large.cell.carcinoma",
  "normal",
  "squamous.cell.carcinoma",

];
```

PCA

PCA is an image reduction method used to analysis the data numerically to find the highest variance features/vectors to choose the principle components that can express most of the variance in the data with the least number of features.

Pipeline Sketch



Pipeline for medical image classification (Chest CT scan)

Generate sub data

- Image generator including : resizing images to size (224, 224 ,3) as RGB,zoom,rotation , horizontal and vertical flipping

Extract labels

- Extract from the test set for evaluation and count visualization for different tumor types

Dimentionality Reduction

- Reducy multi dimentional array into one dimention array
- feature selection for most useful features with high PCA

Training Model

- Using pretrained models and ensambling models for getting best accuracy accuracy

Testing and Evaluating Model

- Evaluation test sets and plotting models accuarcy rates

2: CNN Model

Google Colaboratory

https://colab.research.google.com/drive/13vjp9hkjmNs7bwIggzo7vuAwzOtD1xo9?usp=sharing

2.1. Introduction

This model leverages the VGG16 architecture, pre-trained on ImageNet, as a feature extractor for image classification tasks. The goal is to utilize transfer learning to benefit from the knowledge acquired by the VGG16 model while adapting it to a specific image classification task.

2.2. Data Generation & Augmentation

Data Generator (Applied for Train, Test & Valid Data):

- Target size is (224, 224).
- Batch size is 32.
- Class mode is categorical.

Augmentation Techniques used:

- **Rotation:** Images are randomly rotated up to 10 degrees.
- **Shifts:** Horizontal and vertical shifts of up to 20% of the image dimensions.
- **Shear:** Random shearing with a shear intensity of 0.2.
- **Zoom:** Random zooming with a zoom range of 0.2.
- **Horizontal Flip:** Randomly flips images horizontally.

Purpose of Data Augmentation:

- **Diversity:** Augmentation introduces diversity into the training set, preventing overfitting and enhancing model generalization.
- **Robustness:** The model becomes more robust to variations in input images.

2.3. Model Architecture

VGG16 Feature Extractor:

- **Architecture:** The VGG16 model is employed as a feature extractor.
- **Weights:** Utilizes pre-trained weights from ImageNet.
- **Freezing Layers:** All layers of the VGG16 model are frozen to preserve learned features during training

Model Construction:

- **Sequential Model:** The architecture is constructed as a Sequential model.
- **Layer Customization:** The VGG16 model is followed by additional layers to tailor it to the specific image classification task.

Additional Layers:

- **Batch Normalization:** Applied after the VGG16 feature extraction to normalize the activations.
- **MaxPooling2D:** A 2x2 MaxPooling layer is employed to reduce spatial dimensions.
- **Dropout Layers:** Introduces regularization to prevent overfitting at various stages in the model.
- **Flatten:** Transforms the multi-dimensional output into a flat vector for the fully connected layers.

Fully Connected Layers:

- **Dense (1024):** A densely connected layer with 1024 units and ReLU activation.
- **Dropout:** Regularization via dropout to prevent overfitting.
- **Dense (512):** Another densely connected layer with 512 units and ReLU activation.
- **Dense (256):** A densely connected layer with 256 units and ReLU activation.
- **Dense (Output Layer):** Final densely connected layer with units equal to the number of classes (4 in this case) and softmax activation for multiclass classification.

2.4. Training Process

The hyperparameters used in the training process:

1. **Early Stopping:**
   - **Patience:** 20
   - **Restore Best Weights:** True
2. **Learning Rate Reduction on Plateau:**
   - **Factor:** 0.5
   - **Patience:** 5
   - **Monitor:** Validation Loss
   - **Minimum Learning Rate:** 0.00001
3. **Model Checkpoint:**
   - **Filename:** 'my_model.keras'
   - **Monitor:** Validation Accuracy
   - **Save Best Only:** True
4. **Optimizer:**
   - **Type:** Stochastic Gradient Descent (SGD)
   - **Learning Rate:** 0.01
   - **Momentum:** 0.9
5. **Model Training:**
   - **Epochs:** 50
   - **Verbose:** 1 (Print progress bar)
   - **Callbacks:** Early Stopping, Learning Rate Reduction, Model Checkpoint

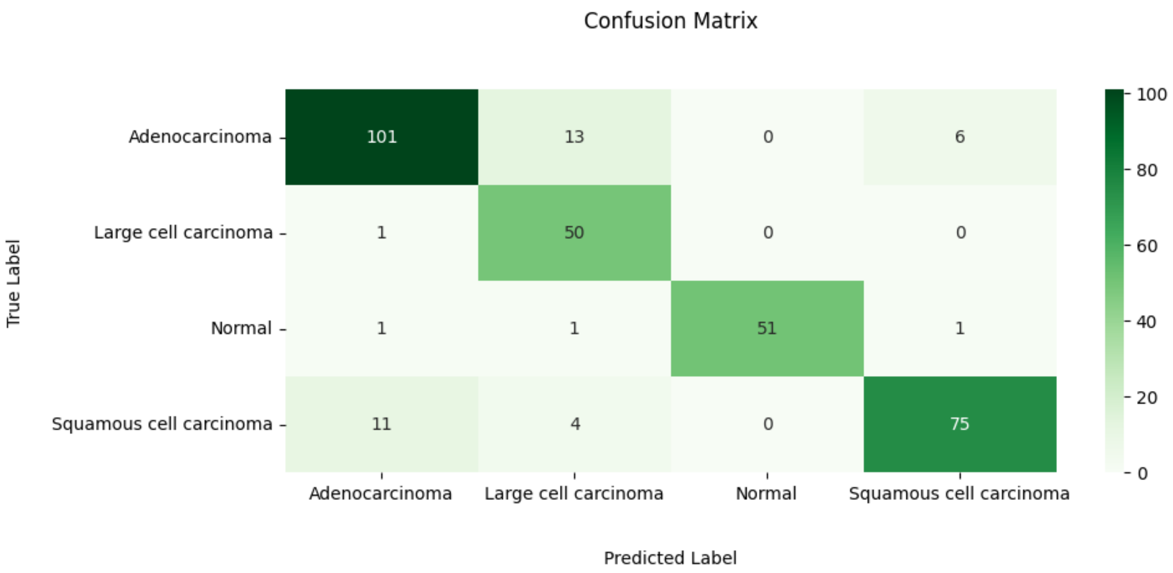
2.5. Performance Metrics

After training, the model was evaluated on the validation and test datasets, and the following performance metrics were recorded:

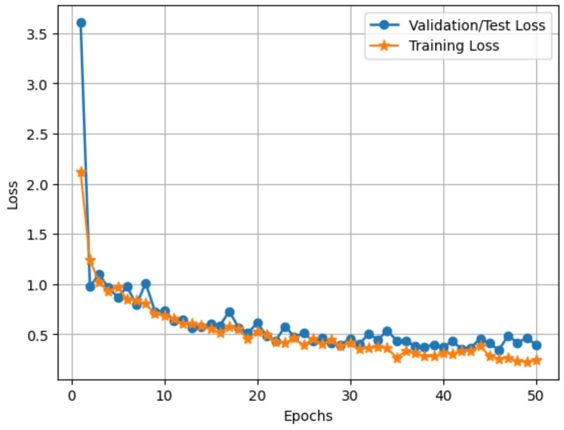
```
10/10 [==============================] - 3s 287ms/step
              precision    recall  f1-score   support

           0       0.89      0.84      0.86       120
           1       0.74      0.98      0.84        51
           2       1.00      0.94      0.97        54
           3       0.91      0.83      0.87        90

    accuracy                           0.88       315
   macro avg       0.88      0.90      0.89       315
weighted avg       0.89      0.88      0.88       315
```

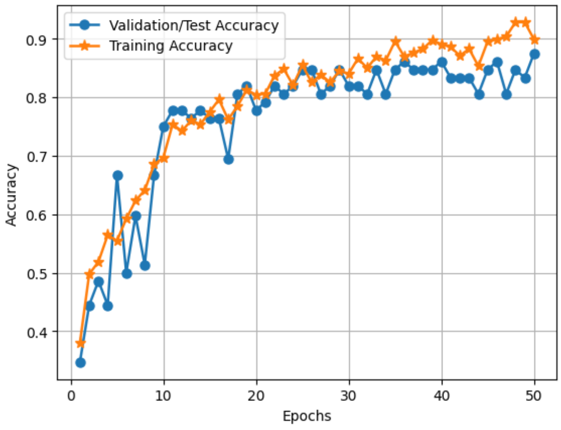
2.6. Plotting and Visualization

Confusion Matrix:



Visualization for the Accuracy and Loss metrics:





3: Ensample Model

> ### Why Pre-trained models?
>
> 1. They possess complex architectures designed to automatically extract hierarchical features from images.
> 2. These architectures have proven effective in learning intricate patterns and features, which might be beneficial for capturing relevant details in CT scan images.
> 3. Leveraging pre-trained models allows you to use knowledge learned from previous tasks (e.g., classifying everyday objects) and transfer it to specific task (identifying features in CT scans).
> 4. This approach often requires less data for fine-tuning and can lead to faster convergence and potentially better performance, especially when dealing with limited datasets, such as medical imaging datasets.

3.1. Overview

Why Ensample Model?

Improved Generalization: Diverse Models: Ensembles typically use different types of models or train multiple models with different initializations or hyperparameters. This diversity allows the ensemble to capture various aspects of the data and learn different patterns.

Reduced Overfitting: By combining multiple models that might have different strengths and weaknesses, ensembles can mitigate the risk of overfitting. Disagreements between models on certain data points can help in making more generalized predictions. Stability and Robustness: Robustness to Noisy Data: Ensembles are often more robust to noise and outliers in the data because they rely on the consensus or majority vote of multiple models.

Stability: Ensembles tend to be more stable and less sensitive to changes in the dataset or minor variations in training compared to a single model.

3.2. Data Augmentation

Expose the model to a variety of variations in the training data. This helps the model generalize better and become more robust to different orientations, scales, and perspectives of the images.

Data Generator

- **Target Size:** (224, 224)
- **Batch Size:** 32
- **Class Mode:** Categorical

Data Augmentation Techniques

- **Rescaling:** Pixel values were rescaled to a range between 0 and 1.
- **Shear Range:** Random shearing transformations with a shear range of 0.2 were applied.
- **Zoom Range:** Random zooming with a zoom range of 0.2 was performed.
- **Horizontal Flip:** Random horizontal flipping was applied.

3.3. Inception Model

3.3.1. Overview

This model leverages the InceptionV3 architecture, a powerful pre-trained convolutional neural network (CNN) designed for image classification tasks. The primary objective is to fine-tune the model for a specific classification task using a dataset containing images of shape (224, 224, 3). This report provides a concise summary of the model architecture.

About Inception Model

Multi-Scale Analysis of Lung Structures: Chest CT scans often contain intricate details of lung structures at varying scales. Inception's multi-scale feature extraction capability, facilitated by its diverse filter sizes within inception modules, allows capturing fine details (like nodules or lesions) as well as broader lung features, aiding in comprehensive analysis.

Efficient Representation of Complex Patterns: The dataset might contain diverse anomalies or conditions within the chest area. Inception's ability to efficiently represent complex patterns through its diverse convolutional operations enables the model to discern and differentiate between various abnormalities, textures, or densities present in CT images.

3.3.2. Model Architecture

Base InceptionV3 Model

The base InceptionV3 model is loaded with pre-trained weights from the 'imagenet' dataset. To adapt the model to the task at hand, the top layers are excluded (include_top=False), and the weights of the initial layers are frozen to prevent retraining.

Customized Top Layers

On top of the InceptionV3 base, custom top layers are added to tailor the model for the specific classification task. These top layers include:

- **Flatten Layer:** Converts the multi-dimensional output of the InceptionV3 base into a one-dimensional array, preparing it for dense layers.
- **Dense Layers:** Two dense layers with 512 units and ReLU activation each, followed by dropout layers with a dropout rate of 0.3. These layers serve to capture high-level features from the flattened output.
- **Output Layer:** The final dense layer consists of 4 units (matching the number of classes) with a softmax activation function for multi-class classification.

3.3.3. Training Process

Loss and Optimizer Configuration

- **Loss Function:** Categorical Crossentropy
  - The choice of Categorical Crossentropy is suitable for multi-class classification tasks.
- **Optimizer:** Adam
  - The Adam optimizer is employed with a learning rate of 0.001 for weight updates during training.

Model Checkpoint

- **Checkpoint Filepath:** 'Chest_CT_SCAN_Inception.h5'
- **Monitor:** Validation Accuracy
- **Save Best Only:** True
- **Verbose:** 1

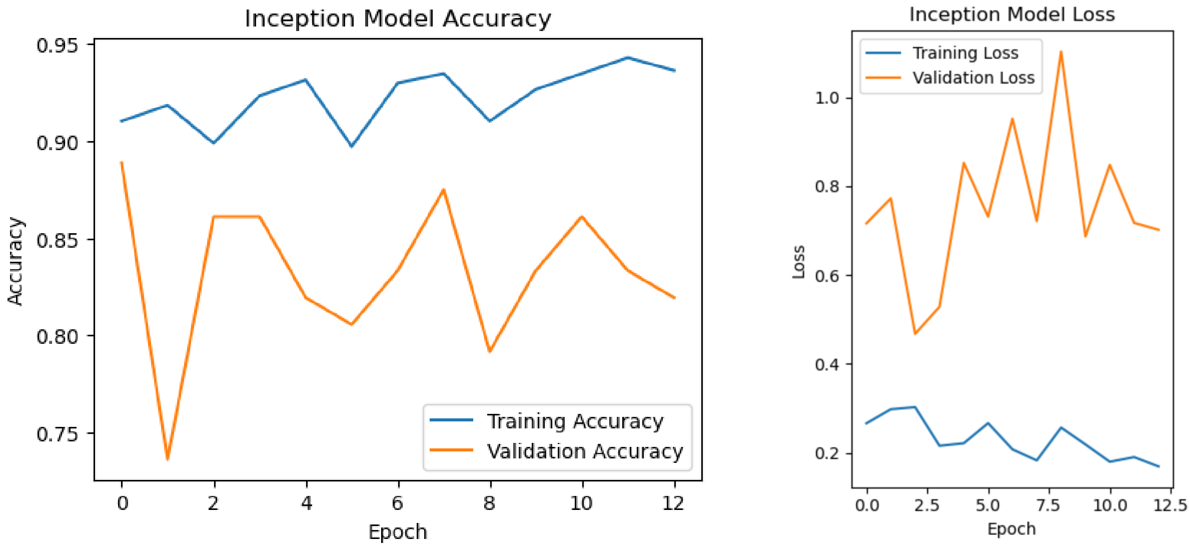
Early Stopping

- **Patience:** 10
- **Verbose:** 1

Training Epochs

- **Number of Epochs:** 20

3.3.4. Plotting and Visualization



3.4. ResNet50 Model

3.4.1. Overview

Why ResNet50?

Detection of Subtle Features: ResNet50's architecture is adept at capturing subtle, nuanced features within images. In chest CT scans, this could be crucial for identifying minute abnormalities or variations that might signify potential health issues. The residual connections in ResNet50 facilitate the learning of these intricate patterns effectively.

Handling Depth and Complex Structures: Chest CT scans often exhibit depth and complexity in anatomical structures. ResNet50's capability to train very deep networks while addressing the vanishing gradient problem is advantageous in extracting meaningful representations from these multi-layered and intricate scans.

3.4.2. Model Architecture

Base Model: ResNet50

Include Top: False

- Excludes the top layers (fully connected layers) from the pre-trained ResNet50 model.

Pooling: Global Average Pooling

- Global Average Pooling is applied after the base ResNet50 layers.

Weights: ImageNet

- The model is initialized with weights pre-trained on the ImageNet dataset.

Input Shape: (224, 224, 3)

- The expected input shape for the model.

Custom Layers (Sequential Model)

Flatten Layer:

- Flattens the multi-dimensional output from the base ResNet50 layers into a one-dimensional array.

Batch Normalization Layer:

- Normalizes the activations of the previous layer for improved stability.

Dense Layer (256 units):

- Fully connected layer with 256 units and ReLU activation.

Dropout Layer (Dropout Rate: 0.5):

- Regularization technique to prevent overfitting by randomly dropping out 50% of the units during training.

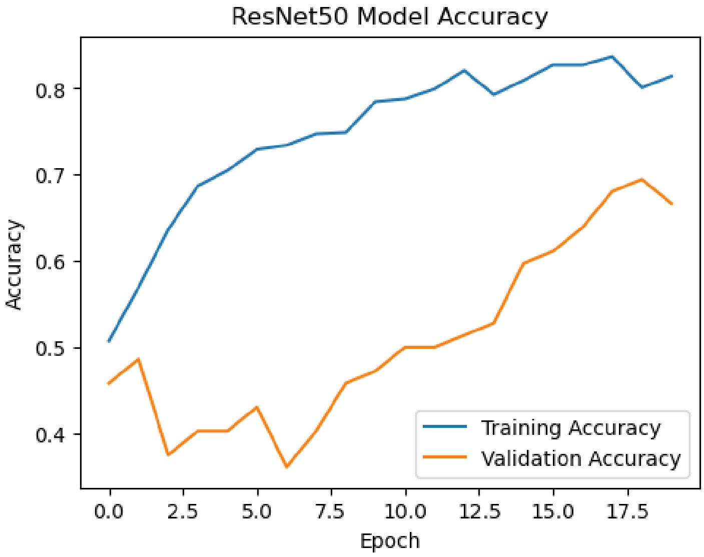
Dense Output Layer (4 units, Softmax Activation):

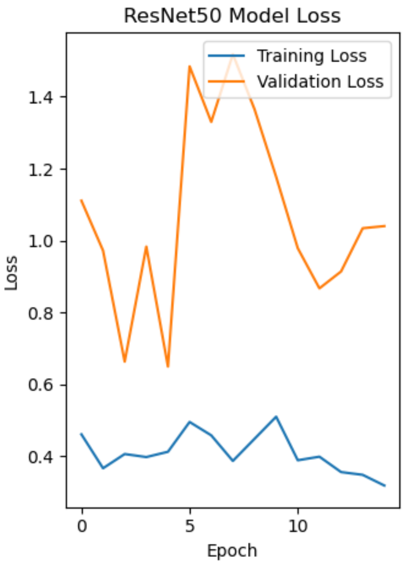
- Output layer with 4 units representing the number of classes in the classification task.
- Softmax activation is applied for multi-class classification.

3.4.3. Training Process

> **Same as for the Inception Model**

3.4.4. Plotting and Visualization


ResNet50 Model Accuracy


ResNet50 Model Loss

3.5. Model Architecture

Base Models

- Inception Model
- ResNet Model

Ensemble Model

Input Shape: (224, 224, 3)

- The expected input shape for the ensemble model.

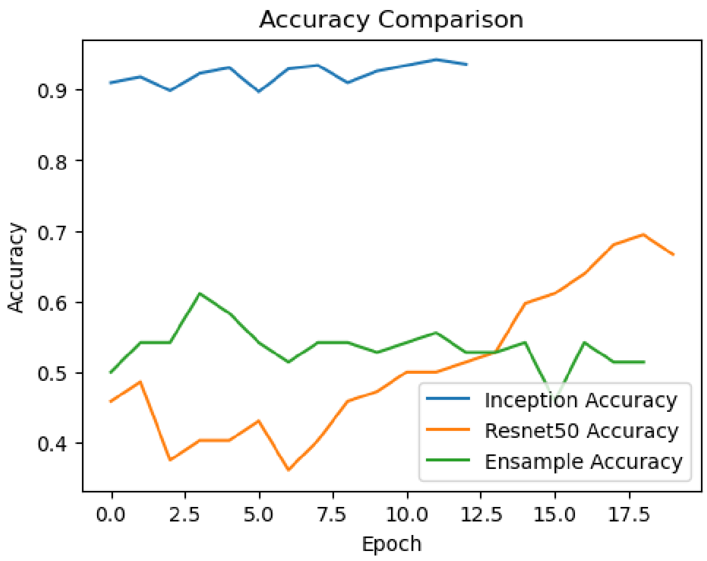
Model Outputs:

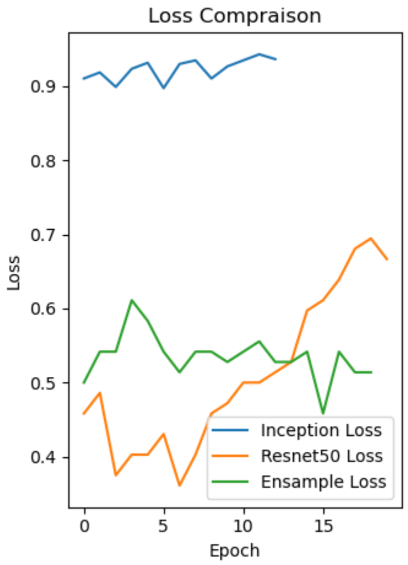
- The outputs of each base model (Inception Model and ResNet Model) are obtained by applying each model to the same input.

Ensemble Output:

- The ensemble output is generated by averaging the individual model outputs.

3.6. Plotting and Visualization





4: Results and Conclusions

Metric/ Model	**CNN (VGG16)**	**Inception**	**ResNet50**	**Ensample**
Accuracy	87.5%	82.2%	62.9%	57.5%
Loss	0.395	0.646	0.854	0.973