Contents

1.Abstract	2
2.Introduction	3
3.Objectives	4
4.Dataset	5
5.Data Preprocessing	5
6.Methodology	8
7.Training	19
8.Results	21
9.Conclusion	23
10.Future Research Scope	25
11.References	25
12.Repository links	27

Abstract

This project aims to advance the field of medical image analysis with a specific focus on COVID-19 detection in CT scans. The primary objective is to investigate trends in loss and accuracy for deep learning models, namely ResNet50 and DenseNet-121, and an encoder-based feature extraction technique. The key aspect of this research is to assess how the size of the training dataset influences the performance of these models. The overarching goal is to enhance the precision and effectiveness of COVID-19 detection by gaining a deeper understanding of how different training sizes impact model accuracy and loss. This project carries significant implications for improving the reliability and efficiency of medical image-based COVID-19 diagnosis, especially when labeled data is scarce.

Introduction

Amidst the persisting challenges posed by the COVID-19 pandemic and as there are many classifiers to classify the images, our project focuses on the comparative analysis of three prominent deep learning models: DenseNet121, ResNet50, and an autoencoder-decoder framework, tailored specifically for the classification of COVID-19 CT scan images. With the critical importance of accurately identifying COVID-19 manifestations from CT scans, our primary goal is to assess how these models perform when presented with datasets of varying sizes. Through a careful examination of the validation losses and accuracies associated with each model across different dataset scales, our study seeks to uncover their respective strengths and limitations in effectively recognizing COVID-19 patterns within CT scan images.

The use of COVID-19 CT scan images as the dataset for our project underscores the critical relevance of our work in real world clinical applications. Leveraging deep learning techniques, our project aims to provide a thorough analysis, not only comparing the performance of these models but also offering crucial insights into their potential impact on enhancing the efficiency and accuracy of COVID-19 diagnosis through automated image classification.

Objectives

- To Assess Model Performance: Evaluate the performance of deep learning models, including ResNet50, DenseNet-121, and an encoder-based technique, in accurately classifying COVID-19 cases in CT scans.
- To Analyze the Impact of Training Data Size: Investigate how different training data sizes influence model accuracy and loss in COVID-19 detection. This analysis provides insights into the trade-off between data size and model performance.
- To Explore Feature Extraction Techniques: Examine the effectiveness of an encoder-based feature extraction method in enhancing model performance, specifically in scenarios where labeled data is limited.
- To Understand Trends in Loss and Accuracy: Identify trends in loss and accuracy as training data size varies. Determine whether larger training datasets consistently lead to improved model performance.
- To Contribute to COVID-19 Diagnosis: Contribute to the field of COVID-19 diagnosis by optimizing the design and training of deep learning models for medical image classification, with a focus on CT scans.
- To Provide Insights for Resource Allocation: Offer insights into the efficient allocation of resources for model training in medical image analysis, particularly in situations with restricted access to labeled data.
- To Inform Future Research: Generate valuable findings and trends that can guide future research in medical image analysis, deep learning, and COVID-19 detection using CT scans.

Dataset

Source: Kaggle
Dataset Link:
https://www.kaggle.com/datasets/ssarkar445/covid-19-xray-and-ct-scan-image-dataset/

This COVID-19 dataset consists of Non-COVID and COVID cases of both X-ray and CT images. The associated dataset is augmented with different augmentation techniques to generate about 17099 X-ray and CT images. The dataset contains two main folders, one for the X-ray images, which includes two separate sub-folders of 5500 Non-COVID images and 4044 COVID images. The other folder contains the CT images. It includes two separate sub-folders of 2628 Non-COVID images and 5427 COVID images.

Data Preprocessing

Before using the images in the autoencoder decoder, ResNet, and DenseNet models, they were all adjusted to the same size. Each image was resized to a shape of 256 pixels in height, 256 pixels in width, and with 3 color channels (RGB). This was done to make sure that all images had the same dimensions and could be processed consistently by the models. Resizing the images in this way helped to avoid any issues with varying image sizes and made the data more manageable for the models to work with.

Data Augmentation

In the domain of deep learning, data augmentation techniques are instrumental in enhancing model performance and generalization, especially in tasks such as image classification. So for DenseNet121 and ResNet50 we did the following augmentations :

1. **Rotation Range**: The *rotation_range = 30* parameter allows for the application of random rotations to the images within the range of 0 to 30 degrees, facilitating diversity in image orientation.

2. **Zoom Range**: The *zoom_range = 0.1* parameter controls the degree of random zooming applied to the images, contributing to the creation of varying scales within the dataset.

3. **Horizontal and Vertical Flipping**: Enabling *horizontal_flip = True* and *vertical_flip = True* allows for the introduction of additional variations through random horizontal and vertical flipping of the input images, thereby expanding the dataset with mirrored representations of the original images.

fig : original image

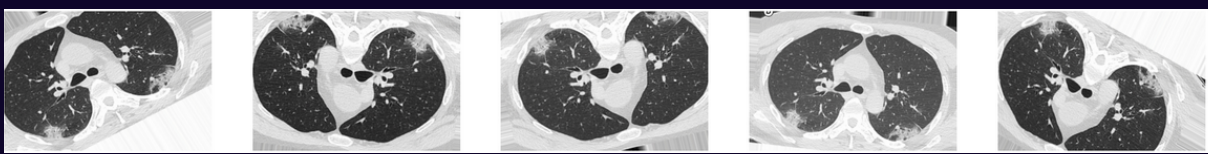

fig : augmented images

Methodology

Classification based on Convolutional Autoencoder

Autoencoder-decoder is a powerful neural network architecture commonly used particularly in the field of deep learning and data compression. The autoencoder-decoder model consists of two primary components, the autoencoder and the decoder, which work in combination to reconstruct input data. The autoencoder is responsible for compressing the input data into a latent-space representation, capturing its essential features, while the decoder reconstructs the original input data from this compressed representation. This architecture finds extensive applications in various domains, such as image processing, natural language processing, and anomaly detection. By learning to efficiently represent and reconstruct input data, the autoencoder-decoder architecture has proven effective in tasks such as image denoising, dimensionality reduction, and data generation. Its ability to learn meaningful representations of complex data has made it a valuable tool for various data-driven applications, facilitating the extraction of meaningful patterns and features from large datasets.

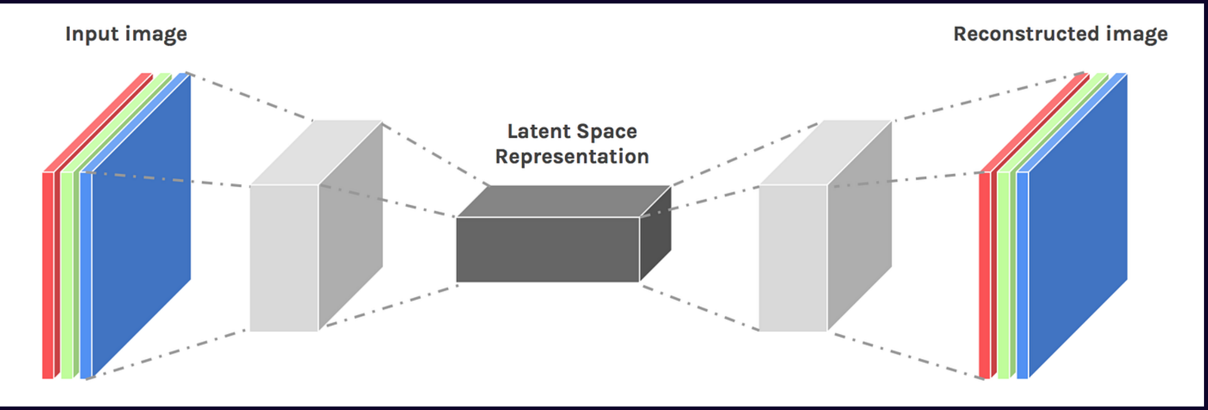

fig : autoencoder decoder architecture

Layer (type)	Output Shape	Param #
INPUT (InputLayer)	[(None, 256, 256, 3)]	0
conv2d (Conv2D)	(None, 256, 256, 16)	448
conv2d_1 (Conv2D)	(None, 256, 256, 16)	2320
max_pooling2d (MaxPooling2D)	(None, 128, 128, 16)	0
conv2d_2 (Conv2D)	(None, 128, 128, 8)	1160
conv2d_3 (Conv2D)	(None, 128, 128, 8)	584
conv2d_4 (Conv2D)	(None, 128, 128, 8)	584
conv2d_5 (Conv2D)	(None, 128, 128, 3)	219
CODE (MaxPooling2D)	(None, 64, 64, 3)	0

fig : model architecture of encoder

```
 CODE (MaxPooling2D)         (None, 64, 64, 3)         0

 conv2d_transpose (Conv2DTra  (None, 64, 64, 3)        84
 nspose)

 conv2d_transpose_1 (Conv2DT  (None, 64, 64, 8)        224
 ranspose)

 conv2d_transpose_2 (Conv2DT  (None, 64, 64, 8)        584
 ranspose)

 conv2d_transpose_3 (Conv2DT  (None, 64, 64, 8)        584
 ranspose)

 up_sampling2d (UpSampling2D  (None, 128, 128, 8)      0
 )

 conv2d_transpose_4 (Conv2DT  (None, 128, 128, 16)     1168
 ranspose)

 conv2d_transpose_5 (Conv2DT  (None, 128, 128, 16)     2320
 ranspose)

 up_sampling2d_1 (UpSampling  (None, 256, 256, 16)     0
 2D)

 OUTPUT (Conv2D)             (None, 256, 256, 3)       435

=================================================================
Total params: 10,714
Trainable params: 10,714
Non-trainable params: 0
```

fig : model architecture of decoder

The Autoencoder decoder is trained with 4000 COVID images.Later, after training encoder-decoder based architecture, the encoder is decoupled and connected to densenet121 for further classification.

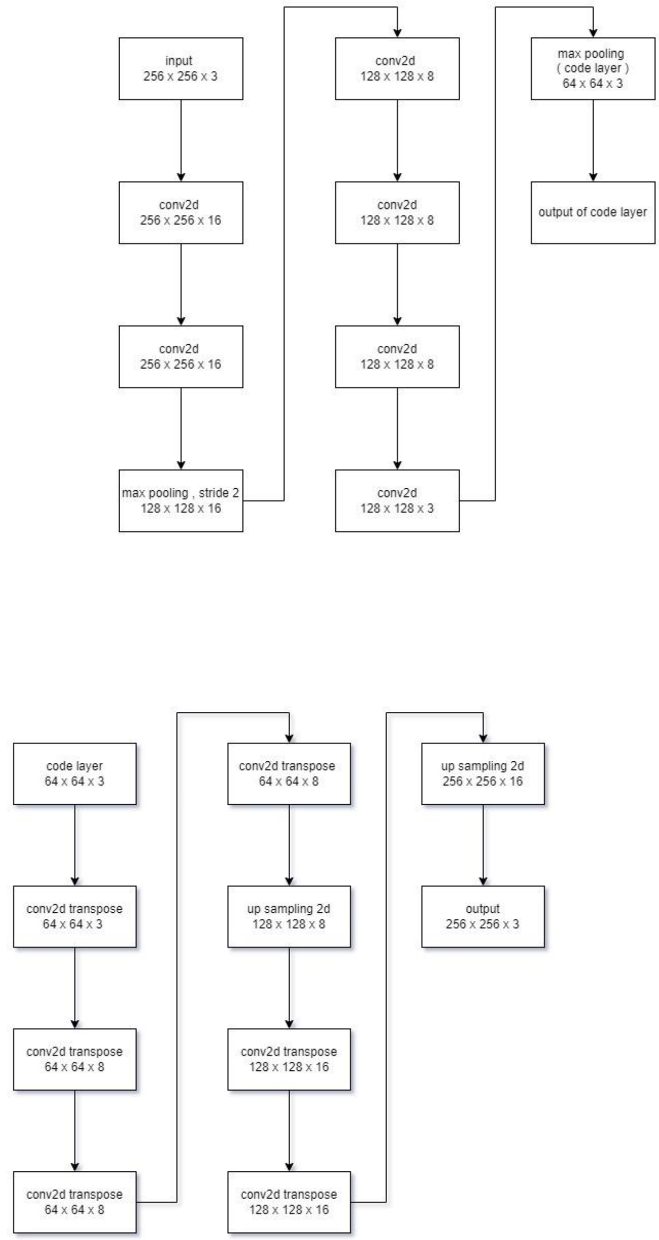

fig : our model architecture of encoder decoder

ResNet50

ResNet50, a variant of the ResNet (Residual Network) model, is a deep convolutional neural network that has made significant contributions to the field of computer vision and image recognition. ResNet50 is characterized by its 50-layer deep architecture, which addresses the vanishing gradient problem that often occurs in training very deep neural networks. The core innovation of ResNet50 lies in the introduction of residual blocks, which enable the learning of residual functions with respect to the layer inputs, thereby allowing the model to effectively optimize and learn more complex features. By utilizing skip connections that bypass one or more layers, ResNet50 enables the network to learn the residual mappings, facilitating the training of much deeper networks without encountering the degradation of accuracy. This architectural design has proven instrumental in improving the performance of deep learning models in various image recognition tasks, including image classification.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_8 (InputLayer)        [(None, 256, 256, 3)]     0

 conv2d_6 (Conv2D)           (None, 256, 256, 8)       224

 conv2d_7 (Conv2D)           (None, 256, 256, 3)       219

 resnet50 (Functional)       (None, None, None, 2048)  23587712

 global_average_pooling2d_3   (None, 2048)             0
 (GlobalAveragePooling2D)

 batch_normalization_6 (Batc  (None, 2048)             8192
 hNormalization)

 dropout_6 (Dropout)         (None, 2048)              0

 dense_3 (Dense)             (None, 256)               524544

 batch_normalization_7 (Batc  (None, 256)              1024
 hNormalization)

 dropout_7 (Dropout)         (None, 256)               0

 root (Dense)                (None, 2)                 514

=================================================================
Total params: 24,122,429
Trainable params: 24,064,701
Non-trainable params: 57,728
```

fig : model architecture of resnet50

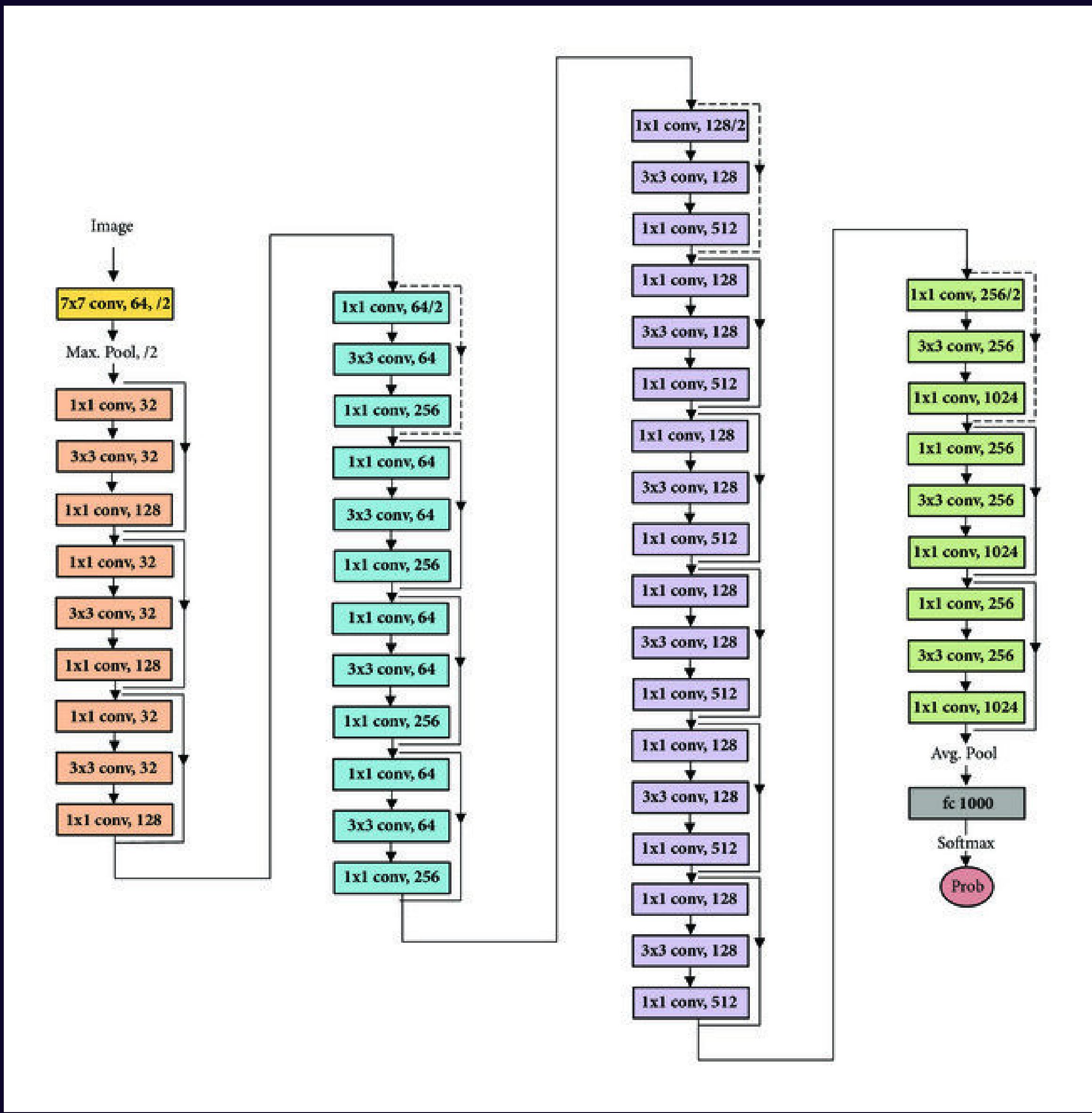

fig : block diagram of a residual network with 50 layers (ResNet 50)

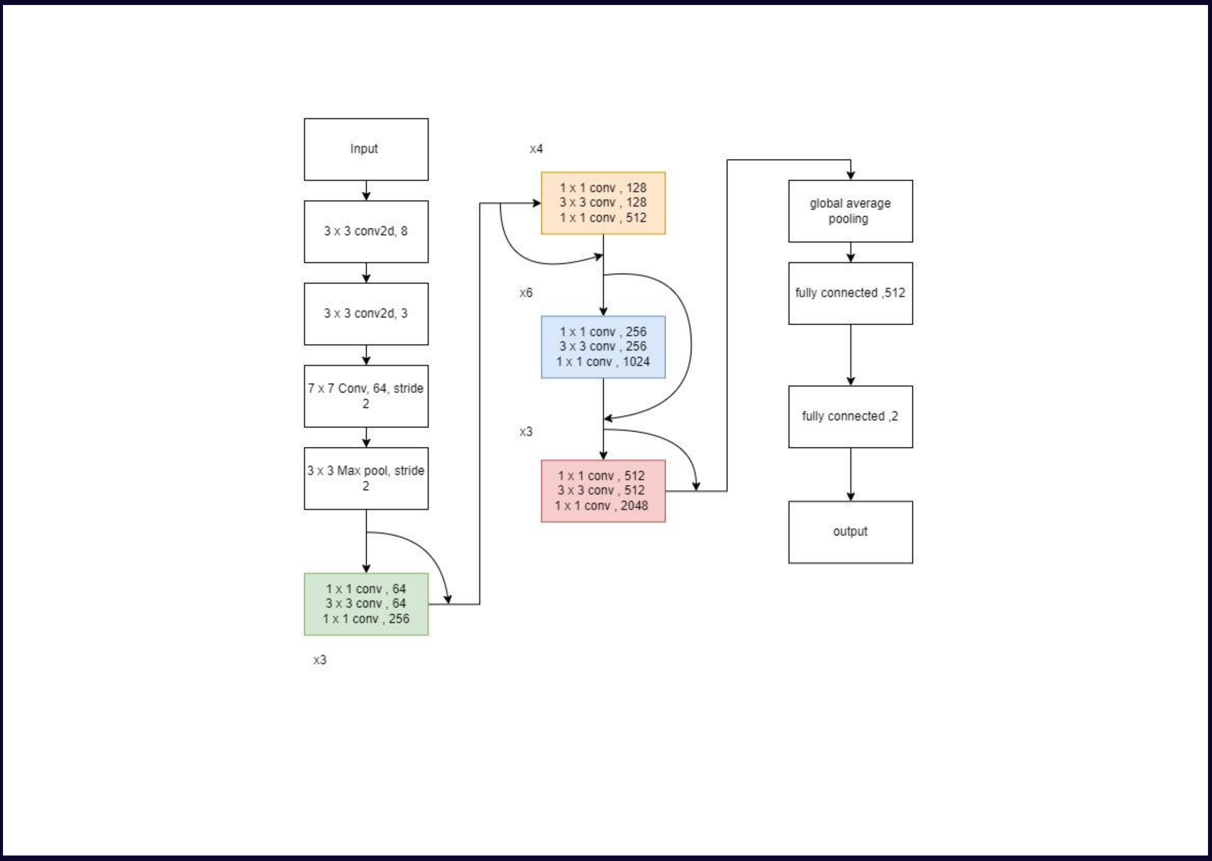

fig: our model architecture

DenseNet121

DenseNet121 is a convolutional neural network architecture that has acquired significant attention in the field of deep learning . DenseNet121 is known for its efficiency in addressing the vanishing gradient problem, similar to ResNet, by leveraging dense connectivity patterns among its layers. Unlike traditional architectures that pass information only in a forward direction, DenseNet121 facilitates direct connections between all layers within a dense block. This unique approach allows the network to efficiently learn from earlier layers, utilizing feature reuse and promoting feature propagation throughout the network. The architecture is named '121' to represent its depth, signifying the use of 121 layers, which consists of multiple dense blocks and transition layers. DenseNet121 has demonstrated remarkable performance in various image-related tasks, showcasing its strength in image classification. Its ability to efficiently integrate information from preceding layers has made it a valuable tool for extracting intricate features from complex visual data.

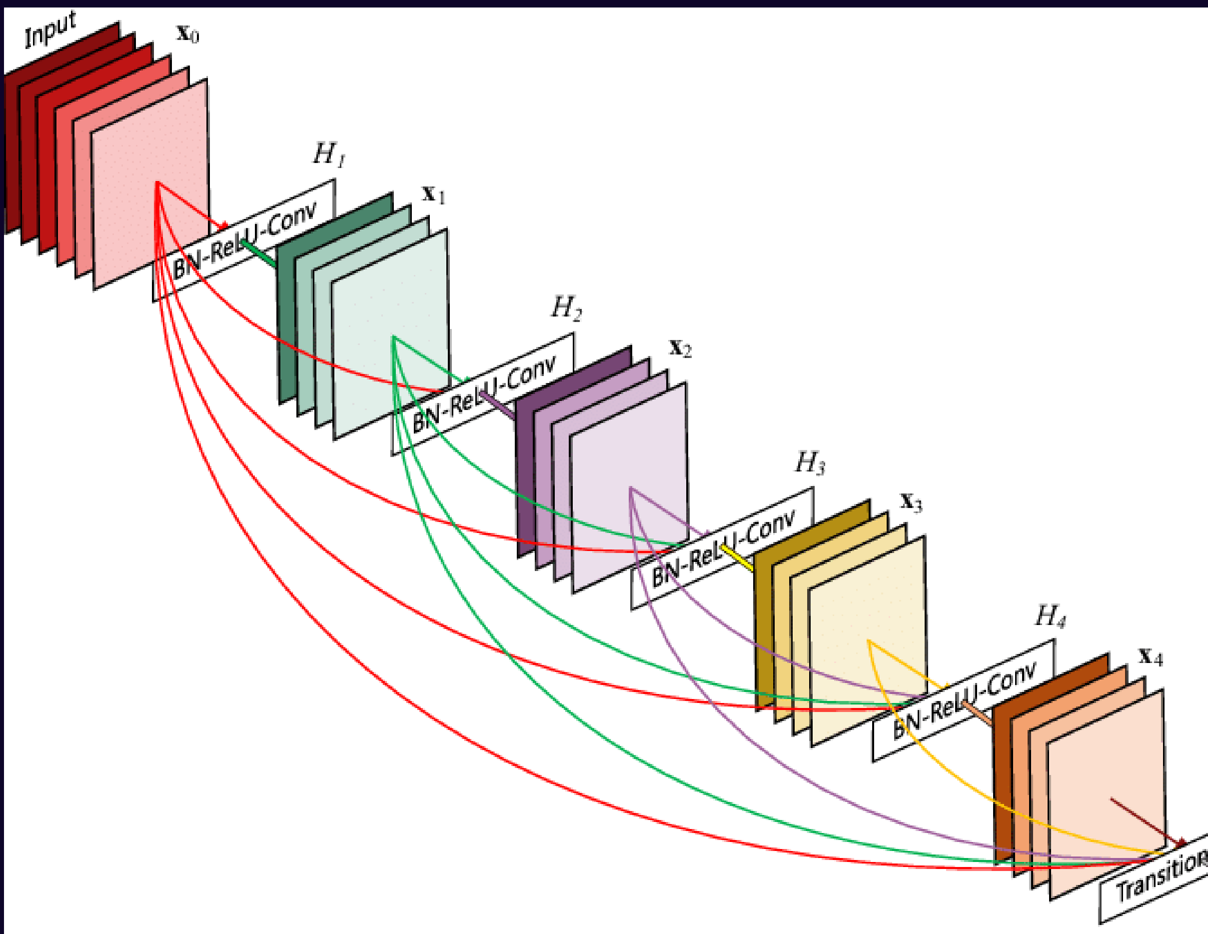

fig : A 5-layer dense block with a growth rate of k = 4. Each layer takes all preceding feature-maps as input

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 256, 256, 3)]     0

 conv2d (Conv2D)             (None, 256, 256, 8)       224

 conv2d_1 (Conv2D)           (None, 256, 256, 3)       219

 densenet121 (Functional)    (None, None, None, 1024)  7037504

 global_average_pooling2d (G  (None, 1024)             0
 lobalAveragePooling2D)

 batch_normalization (BatchN  (None, 1024)             4096
 ormalization)

 dropout (Dropout)           (None, 1024)              0

 dense (Dense)               (None, 256)               262400

 batch_normalization_1 (Batc  (None, 256)              1024
 hNormalization)

 dropout_1 (Dropout)         (None, 256)               0

 root (Dense)                (None, 2)                 514

=================================================================
Total params: 7,305,981
Trainable params: 7,219,773
Non-trainable params: 86,208
_________________________________________________________________
```

fig : model architecture of densenet121

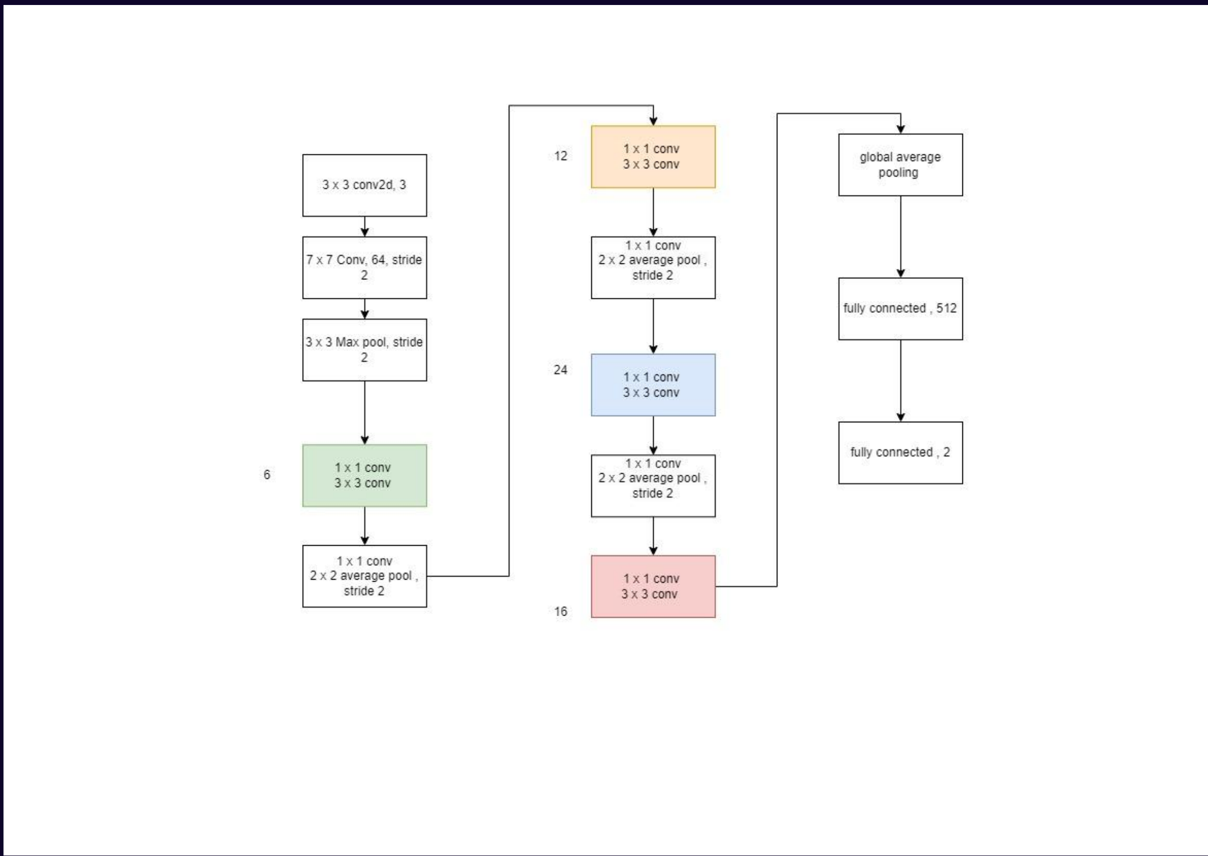

fig : our densenet model architecture

Training

During the training stage, we used different setups to train the model based on the different sizes of the datasets. For the datasets with 100 and 200 images, we used a batch size of 4 and trained over 200 epochs. This helped the model to handle smaller batches of data over multiple rounds of training, improving the learning process for the limited dataset size.

Similarly, for the larger datasets containing 500 and 1000 images, we changed the batch size to 16 and limited the training to 100 epochs. With the increased batch size, the model managed to process larger amounts of data in each round of training, thus making the training process more efficient for the bigger dataset. The reduced number of epochs was chosen to balance the computational resources and training time, while still ensuring the best possible convergence and model performance.

And also for encoder based classification , the preprocessed images of shape 256 * 256 * 3 are converted into 64*64*3 through latent representation in which useful features are captured.

We also implemented data shuffling for all the sets of experiments. This ensured that the model encountered a varied and randomized sequence of data points during each training epoch, preventing it from learning specific patterns tied to the dataset's order.

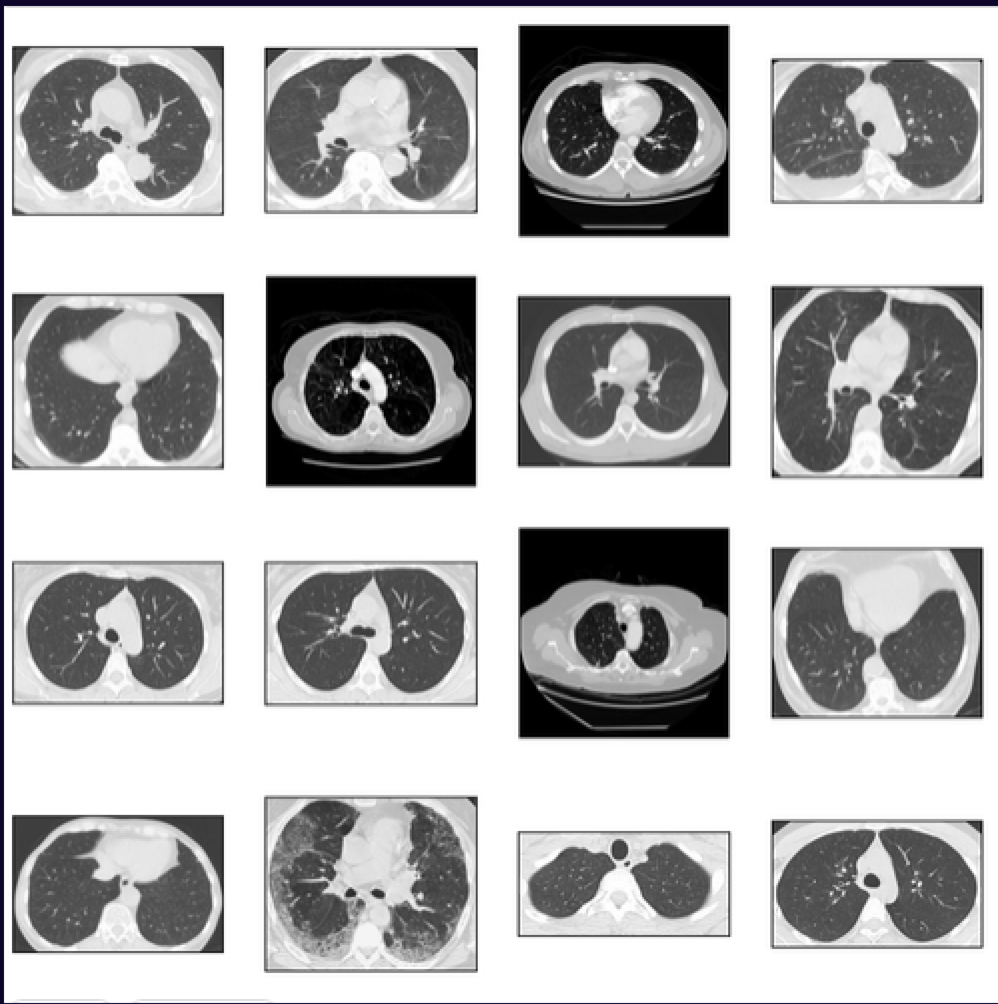

fig : batch of images for training

Loss Metrics : Log Loss (Categorical cross Entropy)

$$\text{Log loss} = \frac{1}{N} \sum_{i=1}^{N} -\left(y_i * \log(p_i) + (1-y_i) * \log(1-p_i) \right)$$

yi : Original label

pi : predicted label

Results

The models are tested with 3000 validation images and we got the following results :

ACCURACY

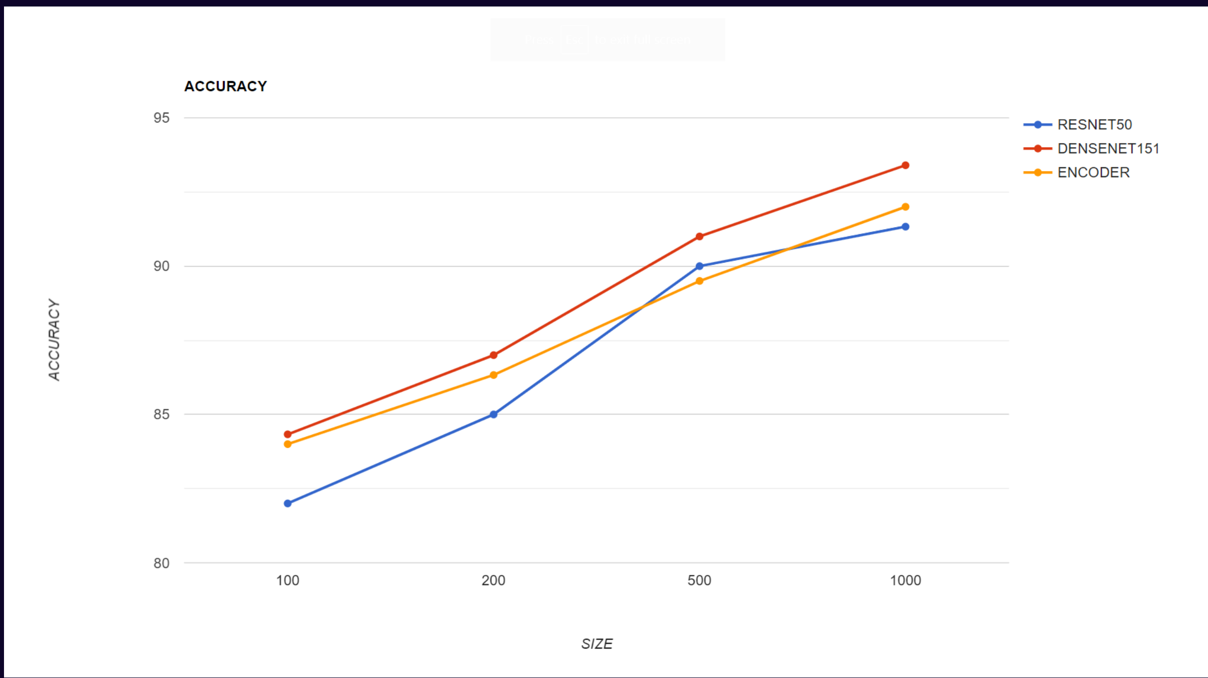

fig : a plot of accuracies of each model vs the size of the training data

ACCURACY

No.of Images Model	100	200	500	1000
RESNET50	82	85	90	91.6
DENSENET121	84.33	87	91	93.4
ENCODER BASED CLASSIFICATION	84	86.33	89.5	92

VALIDATION LOSS

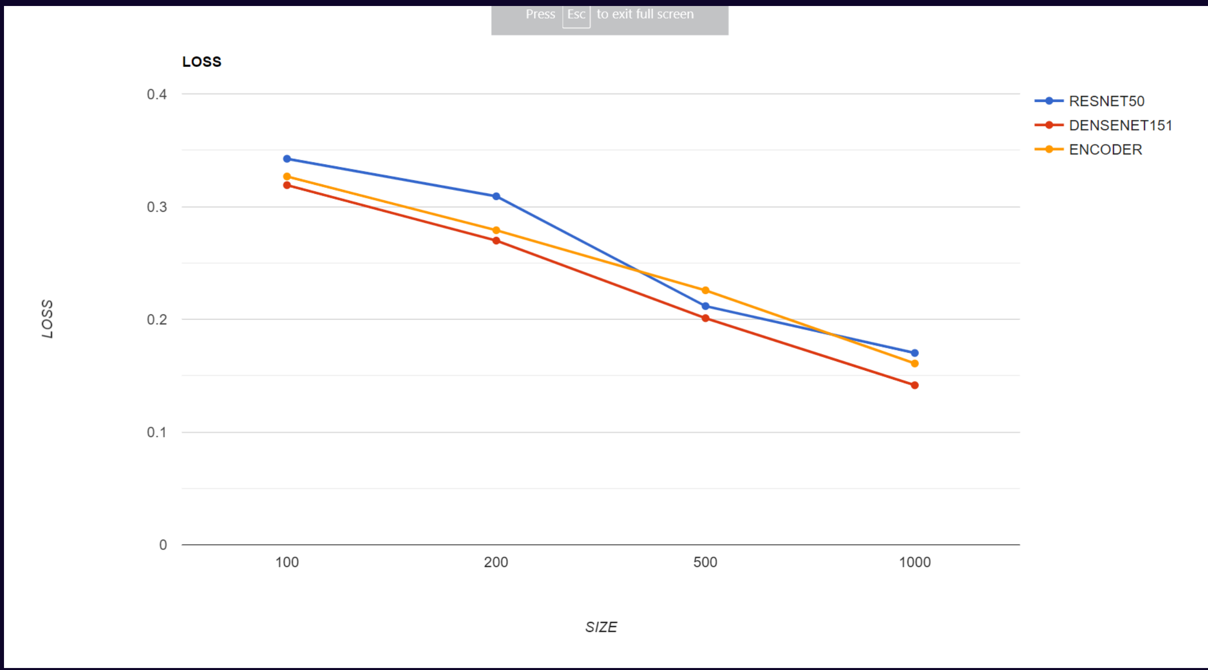

fig : a plot showing the validation losses of each model versus the size of the dataset

VALIDATION LOSS

No.of Images Model	100	200	500	1000
RESNET50	0.3424	0.3091	0.2117	0.1701
DENSENET121	0.3190	0.2698	0.2010	0.1415
ENCODER BASED CLASSIFICATION	0.3267	0.2790	0.2257	0.1608

Conclusion

The observed trend in accuracy, with DenseNet-121 outperforming both the encoder-based classification and ResNet-50 in COVID-19 image classification(by feature extraction) with extremely limited training data, highlights the significance of dense connectivity. Notably, the encoder-based classification, incorporating self-supervised learning during training and feature extraction, demonstrated improved performance over ResNet-50. However, despite leveraging self-supervised techniques, the encoder-based approach fell short of DenseNet-121.In practical applications with minimal training data, prioritizing DenseNet-121 is advisable for enhanced classification accuracy.

Probable Reasons for the Observed Trend:

The observed trend, with DenseNet-121 exhibiting superior performance in COVID-19 image classification(by feature extraction) compared to both the encoder-based classification (utilizing self-supervised learning during training and feature extraction) and ResNet-50, can be attributed to several factors.

Dense Connectivity in DenseNet-121:
DenseNet-121's dense connectivity pattern, where each layer receives input from all previous layers, enables efficient information flow and feature reuse. In scenarios with limited training data, this architecture excels in capturing and leveraging intricate patterns, contributing to its higher accuracy.

Self-Supervised Learning in the Encoder-Based Technique:
While the encoder-based classification incorporates self-supervised learning during both training and feature extraction, it appears that this alone might not be sufficient to overcome the advantages offered by DenseNet-121's dense connectivity. Self-supervised learning enhances feature extraction by learning representations from unlabeled data, but it seems that the complexity of COVID-19(by feature extraction) image patterns demands the additional richness provided by dense connectivity.

ResNet-50's suboptimal performance :
In COVID-19 image classification(by feature extraction) with limited training data can be attributed to the architecture's relatively simpler residual connections, which, while effective for deep network training, may struggle to capture nuanced features from a constrained dataset. The absence of dense connectivity, a characteristic feature of DenseNet-121, limits ResNet-50's ability to efficiently reuse features and grasp intricate patterns in the data. The model's less intricate feature extraction might be inadequate for discerning the complex relationships present in COVID-19 images(by feature extraction), leading to a lower classification accuracy compared to DenseNet-121.

As we saw initially the difference between the accuracies of densenet121 and our encoder-based model is less, which is a positive sign, we can even get better results than densenet121 if we try exploring other types of encoding techniques.

Future Research Scope

A promising avenue for future research lies in exploring the enhancement of the accuracy of the encoder technique for COVID-19 (by extracting the features)image classification. The current findings indicate room for improvement, and one potential direction involves leveraging autoencoders with transformer architectures instead of traditional convolutional autoencoders. Transformers have demonstrated remarkable success in various natural language processing tasks, and their application to image-based autoencoders holds potential for capturing intricate features and patterns more effectively. Investigating the integration of transformer-based autoencoders into the encoder-based classification approach may yield improved feature extraction and representation, potentially elevating the accuracy of COVID-19 image classification in scenarios with limited training data.

References

Deep Residual Learning for Image Recognition -Paper by Kaiming He , Xiangyu Zhang , Shaoqing Ren

Densely Connected Convolutional Networks -Paper by Gao Huang, Zhuang Liu, Laurens van der Maate

A Deep Convolutional Auto-Encoder with Pooling - Unpooling Layers in Caffe -Paper by Volodymyr Turchenko, Eric Chalmers, Artur Luczak

Repository Links

Densenet121/Resnet50 Classifier :
https://www.kaggle.com/code/rishiyangala/fork-of-classifier/edit/run/149969021

Convolutional Autoencoder :
https://www.kaggle.com/code/rishiyangala/fork-of-generator/edit/run/149697333

Encoder connected to Densenet121:
https://www.kaggle.com/code/rishiyangala/fork-of-generator-with-classifier/edit/run/150002435