Lung Disease Prediction Deep CNN models for predicting lung diseases from X-ray Images

Alda Kola - 2071530[‡], Beliz Gunay - 2080284[†]

***Abstract*—In the field of medical diagnostics, the accurate and timely prediction of lung diseases is crucial for patient care and treatment planning. This paper addresses the challenge of lung disease prediction from X-ray images using deep learning and image analysis techniques. Our goal is to develop a reliable and efficient approach for early detection and diagnosis. We propose a transfer learning-based approach with DenseNet, a pre-trained convolutional neural network (CNN) model. By leveraging transfer learning, we benefit from the knowledge and features learned from a large dataset, adapting it to our specific image classification task. Additionally, we introduce Squeeze-and-Excitation (SE) blocks in our network architecture to enhance feature representations and improve the discriminative power of the network. To improve the model's performance and generalization ability, we incorporate batch normalization and regularization techniques. These techniques normalize intermediate activations, mitigate overfitting, and enhance the model's robustness. The significance of our research lies in its potential to improve patient outcomes, streamline healthcare workflows, and reduce the workload on medical experts. The outcomes of this study can benefit both the research community and healthcare practitioners. The proposed approach provides a robust solution for lung disease prediction and provides a contribution to paving the way for further advancements in medical image analysis.**

***Index Terms*—Lung diseases, Convolutional Neural Ntwork, DenseNet, Squeeze-and-Excitation, Image classification**

I. Introduction

In the field of medical diagnostics, the accurate and timely prediction of lung diseases plays a crucial role in patient care and treatment planning. As the availability of digital medical imaging is increasing, particularly chest X-ray images, there is a growing opportunity to leverage advanced computational techniques for automated disease prediction. This paper addresses the important setup of lung disease prediction from X-ray images, aiming to develop a reliable and efficient approach for early detection and diagnosis. By harnessing the power of deep learning and image analysis, our work focuses on creating a robust predictive model that can aid healthcare professionals in identifying lung diseases with high accuracy and efficiency. The significance of this research lies in its potential to improve patient outcomes, streamline healthcare workflows, and reduce the workload on medical experts, ultimately contributing to more effective disease management and treatment strategies [1] [2].

The accurate prediction of lung diseases from X-ray images presents a significant challenge in medical image analysis. Previous attempts at addressing this problem have often relied on traditional machine learning algorithms or simple image features, resulting in limited accuracy and generalization. Recent studies have explored the application of deep learning techniques, such as convolutional neural networks (CNNs), for improved lung disease prediction. However, these approaches have faced challenges in handling class imbalance, limited training data, and the need for extensive manual annotation. Consequently, there remains a critical need for a comprehensive and robust solution that can effectively utilize the power of deep learning to overcome these limitations and provide more accurate and reliable predictions for lung diseases [3] [4].

In this paper, we address the problem of image classification using deep learning techniques. Our proposed approach involves transfer learning with DenseNet, a pre-trained convolutional neural network (CNN) model. The relevance of our approach lies in the increasing demand for accurate and efficient image classification models in various domains, specifically healthcare in our case. By utilizing transfer learning with DenseNet, we can achieve high accuracy while reducing the computational cost and training time. We incorporate batch normalization, regularization techniques, and a dense layer to enhance the model's performance and generalization ability [5]. These modifications help prevent overfitting and improve the model's robustness.

The key contributions of our work can be summarized as follows:

- **Transfer Learning with DenseNet:** We leverage the power of transfer learning by utilizing the pre-trained DenseNet model. This enables us to benefit from the knowledge and features learned from a large dataset and adapt it to our specific image classification task.
- **Incorporation of SE Blocks:** We introduce SE (Squeeze-and-Excitation) blocks in our network architecture. These blocks capture channel-wise dependencies and enhance feature representations by recalibrating the importance of each channel. This improves the discriminative power of the network and leads to more accurate predictions.
- **Integration of Batch Normalization and Regularization:**To enhance the model's performance and generalization ability, we incorporate batch normalization and regularization techniques. These techniques normalize the intermediate activations, mitigate overfitting, and improve the model's robustness against noise and variations in the input data.

Department of Information Engineering, University of Padova
[‡]alda.kola@studenti.unipd.it [†]beliz.guenay@studenti.unipd.it

- **Efficient Model Training and Inference:**By adopting transfer learning and leveraging the pre-trained DenseNet, we significantly reduce the computational cost and training time. This allows for efficient model training and inference, making our approach suitable for real-time or resource-constrained scenarios.
- **Improved Performance:**Experimental results on our specific task of lung disease prediction from X-ray images demonstrate better results compared to traditional machine learning algorithms or simple image features. Our model achieves higher accuracy and generalization, contributing to more reliable and effective disease diagnosis and treatment planning.
- **Paper Structure:** This paper is structured as follows: In Section II, we provide an overview of influential papers in computer vision and image processing. Section III describes our processing pipeline, including dataset details and image preprocessing techniques. The learning framework is presented in Section IV, followed by the results and performance evaluation in Section VI. We conclude with a discussion of the findings in Section VII.

II. Related Work

We present an overview of influential papers that have made significant contributions to the fields of computer vision and image processing. Introducing novel architectures and attention mechanisms that have advanced the domains of biomedical image segmentation, convolutional neural networks, and semantic segmentation. These studies highlight the importance of innovative approaches and attention mechanisms in pushing the boundaries of computer vision and image processing research.

- **Convolutional Networks for Biomedical Image Segmentation:** In the field of biomedical image segmentation, the U-Net architecture proposed by Ronneberger et al. addresses the challenge of accurate segmentation with limited annotated data by employing an encoder-decoder structure. The encoder gradually reduces the spatial dimensions while capturing high-level features, and the decoder reconstructs the segmented image using the encoded features. Notably, U-Net introduces skip connections between the encoder and decoder layers, facilitating the integration of both local and global contextual information during segmentation. This innovative approach has demonstrated superior performance on challenging datasets such as the ISBI 2012 EM Segmentation Challenge dataset. The U-Net architecture, which can be seen in Fig.1, has become widely adopted in biomedical image segmentation tasks, proving its effectiveness in handling limited training data and achieving precise segmentations [3] [6].
- **Enhanced CNN Representations:**In the realm of deep learning architectures, the Squeeze-and-Excitation Networks (SENets) proposed by Hu et al. which can be seen in Fig.2, have addressed significant attention for effectiveness in enhancing the representational power

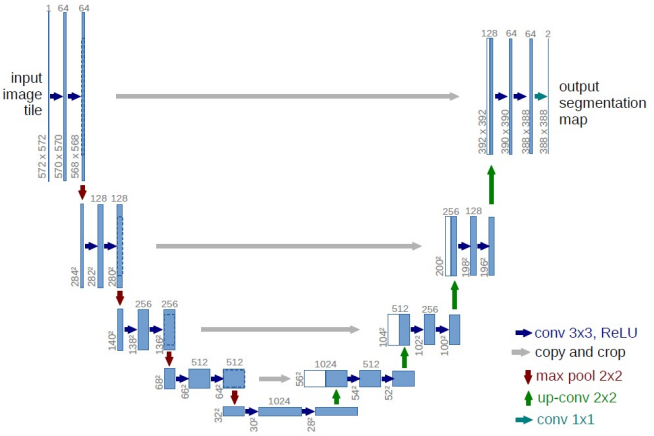

Fig. 1: U-Net Architecture [3]

of convolutional neural networks (CNNs). SENets introduce a novel attention mechanism that enables the network to adaptively recalibrate channel-wise feature responses. This is achieved through a squeeze operation that aggregates global information from the feature maps, followed by an excitation operation that learns channel-specific importance weights. By explicitly modeling interdependencies among the channels, SENets enable the network to focus on discriminative features and suppress less relevant information. Experimental results demonstrate that incorporating SENets into CNN architectures significantly improves performance across various tasks, including image classification, object detection, and semantic segmentation. SENets showcase the potential of attention mechanisms for enhancing the expressive power of CNNs and have paved the way for subsequent developments in attention-based models [4].

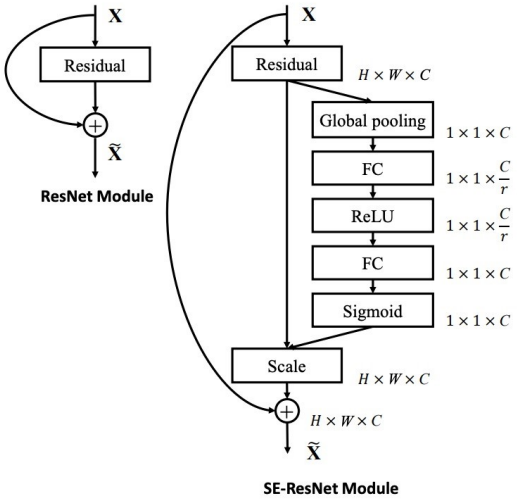

Fig. 2: SE-ResNet Architecture[4]

- **Criss-Cross Attention for Semantic Segmentation:**The "CCNet: Criss-Cross Attention for Semantic Segmentation" paper by Huang et al. presents an attention mechanism designed specifically for semantic segmentation tasks. The proposed Criss-Cross Attention (CCNet) module aims to capture long-range dependencies and enhance spatial context understanding. Unlike traditional attention mechanisms that focus on either horizontal or vertical dependencies, CCNet, which can be seen in Fig.3, captures both directions simultaneously, allowing each position to attend to informative context in a criss-cross pattern. By integrating this attention module into the segmentation network, CCNet achieves remarkable improvements in accuracy and boundary delineation. The experiments conducted on various benchmark datasets, including Cityscapes, PASCAL VOC, and ADE20K, demonstrate the superior performance of CCNet compared to segmentation methods. This work showcases the importance of capturing comprehensive contextual information and highlights the potential of criss-cross attention mechanisms for advancing semantic segmentation tasks [5].

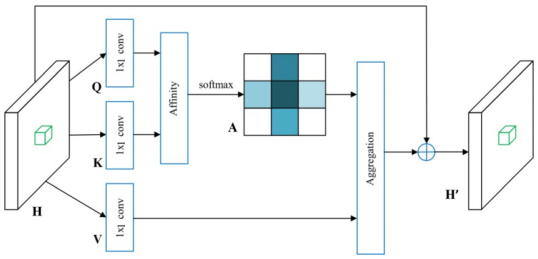

Fig. 3: Criss-Cross Attention[5]

III. Processing Pipeline

The dataset for image processing and analysis tasks consists of three subsets, training, test and validation. The training set contains 2928 images, the test set has 915 images, and the validation set has 732 images. These sets belong to 3 classes, which are resized to 128x128 dimensions during the data pre-processing stage. The label names associated with the classes in the dataset are 'covid', 'normal', 'pneumonia'. We set up the image preprocessing pipeline using Keras' ImageDataGenerator for data augmentation and normalization. It defines the image dimensions, batch size, and paths to the training and test datasets. The generators are configured to load and preprocess images from the specified directories.
Three models are defined and trained in the code section:

- **Transfer Learning with DenseNet:** It loads the pre-trained DenseNet121 model, freezes its weights, and adds custom layers on top. The model is compiled and trained using various callbacks. Training history and evaluation results are displayed.
- **CNN+DenseNet Output:** This model consists of convolutional layers, batch normalization, max pooling, dense layers, dropout, and a softmax output layer. The model is compiled and trained, and the training history and evaluation results are displayed.
- **BUNTT (CNN+SENet Custom Made Block):** A custom SE block function is defined to implement the Squeeze-and-Excitation mechanism. The model architecture includes convolutional layers, SE blocks, batch normalization, max pooling, and dense layers. The model is compiled, trained, and evaluated, and the training history and evaluation results are displayed. The pipeline of the process is shown in Fig .4.

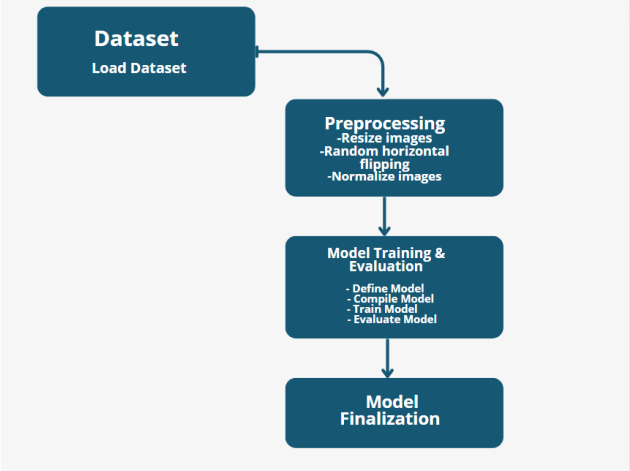

Fig. 4: Processing Pipeline

IV. Images and Features

In this paper, we focus on the classification of lung diseases from X-ray images, Fig. 5. The images used in our study were obtained from a variety of medical imaging sources, following standard clinical protocols. Each image represents a chest X-ray scan of a patient and is formatted as a 2D matrix of grayscale pixel values. The images were preprocessed using the Keras ImageDataGenerator module, which applied various transformations to augment the data. These transformations included rescaling the pixel values to a range of [0, 1] by dividing each pixel value by 255. Furthermore, the preprocessing involved horizontal flipping, random rotation (within a range of 20 degrees), and the nearest fill mode. These techniques helped enhance the training data by introducing variations and reducing overfitting.
The images were resized to a fixed resolution of 128x128 pixels. We adopted a batch size of 32 to process the images in batches during training. The dataset was split into training and validation subsets using a validation split of 0.2. This ensured that a portion of the data was reserved for evaluation and validation purposes. We employed a separate test dataset for evaluating the performance of our model. The test images underwent the same rescaling as the training and validation images.

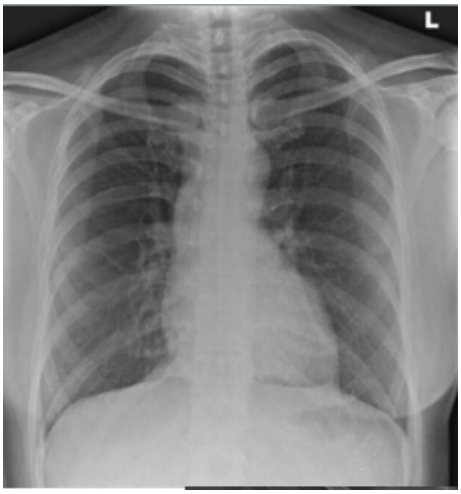

Fig. 5: A sample from the dataset

Overall, these preprocessing steps helped prepare the images for subsequent training and evaluation stages by ensuring consistency in size, applying data augmentation, and dividing the data into appropriate subsets for training, validation, and testing.

V. Learning Framework

We implemented our first model, which was a transfer learning model based on DenseNet. We used the DenseNet121 architecture pretrained on the dataset as the base model. The base model's weights were frozen, and a custom classification head was added on top. The classification head consisted of batch normalization, a dense layer with 256 units, and a softmax activation layer for the output. The model was compiled with the Adam optimizer and categorical cross-entropy loss function.

- **Input Layer:** The input is a 3D tensor of shape (128, 128, 3).
- **Zero Padding :** pads the input with a pad of (3,3).
- **Convolution Block 1:** The 2D Convolution has 64 filters of shape (7,7) and uses a stride of (2,2). BatchNormalization is applied to the channels axis of the input. After the convolution block, Zero Padding and MaxPooling comes with both use a (3,3) window and a (2,2) stride.
- **Convolution Block 2:** The 2D Convolution has (32. 32) size and several sizes of filters (64, 128, 192, 256) and uses kernel size (3,3) and a stride of (2,2). BatchNormalization is applied to the channels axis of the related part of the output. After the convolution block, Zero Padding and MaxPooling comes with both use a (3,3) kernel size window and a (2,2) stride.
- **Convolution Block 3:** The 2D Convolution has (16, 16) size and several sizes of filters (128, 256, 512) and uses kernel size (3,3) and a stride of (2,2). BatchNormalization is applied to the channels axis of the related output. After the convolution block, Zero Padding and MaxPooling comes with both use a (3,3) kernel size window and a (2,2) stride.
- **Convolution Block 4:** The 2D Convolution has (8, 8) size and several sizes of filters (128, 256, 512, 1024) and uses kernel size (3,3) and a stride of (2,2). BatchNormalization is applied to the channels axis of the related output. After the convolution block, Zero Padding and MaxPooling comes with both use a (3,3) kernel size window and a (2,2) stride.
- **Convolution Block 5:** The 2D Convolution has (4, 4) size and several sizes of filters (128, 256, 512, 1024) and uses kernel size (3,3) and a stride of (2,2).
- **BatchNormalization :** is applied to the channels axis of the output.
- **Activation :** uses ReLU as an activation function.
- **2D MaxPooling :** uses a window of shape (2,2) to extract the best features.
- **Flattening :** This flattening layer is a necessary step to correctly format the input for the following layers.
- **Fully Connected Layers :** Reduces its input with 3 Dense layers of shape 1024, 512 and 256 respectively, and 1 output layer with a linear activation function.

During training, several callbacks were used, including ReduceLROnPlateau, EarlyStopping, TensorBoard, CSVLogger, and ModelCheckpoint. These callbacks helped monitor the validation accuracy and loss, reduce the learning rate if no significant improvement was observed, early stop the training if the validation accuracy did not improve, and save the best model based on validation accuracy.

The model was trained for 30 epochs, and the best weights were loaded from the model checkpoint. The performance of the model was evaluated using classification metrics such as accuracy, loss, classification report, and confusion matrix. The results were displayed and visualized using plots and tables. Additionally, we implemented two more models for comparison.

The second model, referred to as ModelBUN, was a combination of convolutional and DenseNet layers. It consisted of several convolutional layers with batch normalization, max pooling, and a fully connected layer for classification. The model was trained using the same process as the first model.

The third model, named BUNTT, utilized a custom-made block called SE-Block, inspired by Squeeze-and-Excitation Networks. The SE-Block included global average pooling, a dense layer, and element-wise multiplication with the input tensor. The BUNTT model incorporated this SE-Block along with convolutional and DenseNet layers. Similar to the previous models, it was trained using the same process.

The results of all three models were compared in terms of accuracy, loss, classification report, and confusion matrix.

VI. Results

As we can see in the training accuracy, training loss, validation accuracy and validation loss figures and values, we built three models: DenseNet 121, CNN with DenseNet (we call this model as BUN) and CNN with SENet (we call this model BUNTT) results are seperately seen. At the third model (BUNTT), we have the best accuracy and loss values.

Training Accuracy	0.9784836173057556
Training Loss	0.18746545910835266
Validation Accuracy	0.7745901346206665
Validation Loss	1.2969897985458374

TABLE 1: Last Epoch Parameters

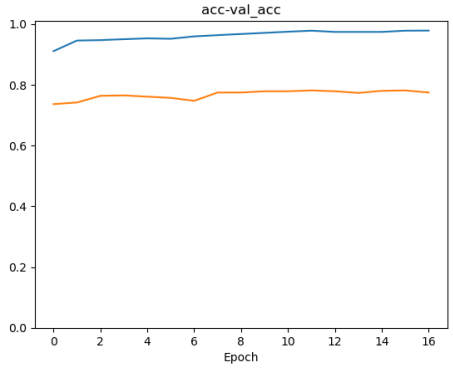

Fig. 6: DenseNet 121 training and validation accuracy

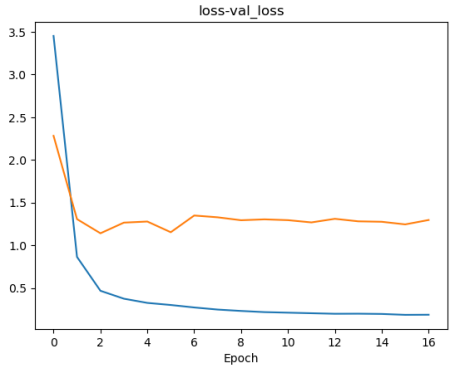

Fig. 7: DenseNet 121 training and validation loss

	precision	recall	f1-score	support
covid	0.8943	0.9705	0.9308	305
normal	0.8854	0.9115	0.8982	305
pneumonia	1.0000	0.8852	0.9391	305

TABLE 2: DenseNet 121 Classification Output

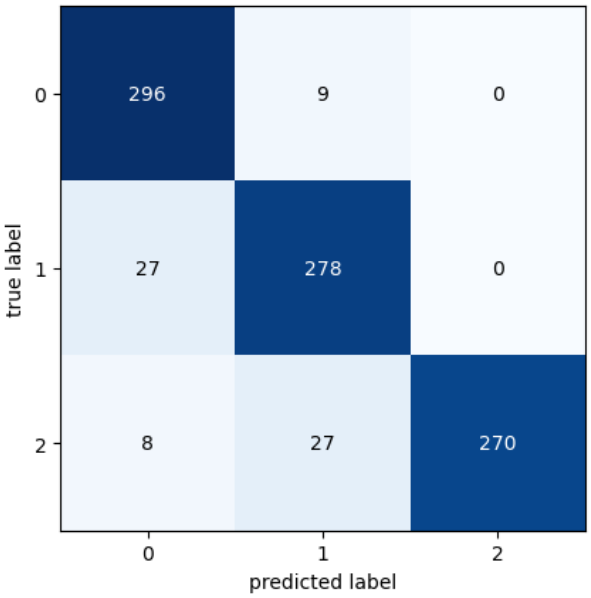

Fig. 8: DenseNet 121 confusion matrix

According to the results above and the final test accuracy value we got, which is 92,65%, shows that our model performed a successful classification. We can consider the accuracies above 90% as successful trainings. In addition to DenseNet 121, we used the CNN with DenseNet (BUN model) in our second model to obtain better results. The detailed measurements are given below.

- **Total params:** 6,432,099
- **Trainable params:** 6,428,035
- **Non-trainable params:** 4,064

Training Accuracy	0.994535505771637
Training Loss	0.0184347964823246
Validation Accuracy	0.7923497557640076
Validation Loss	4.315469264984131

TABLE 3: Last Epoch Parameters

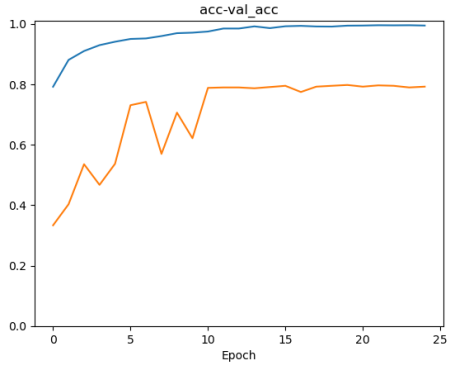

Fig. 9: BUN training and validation accuracy

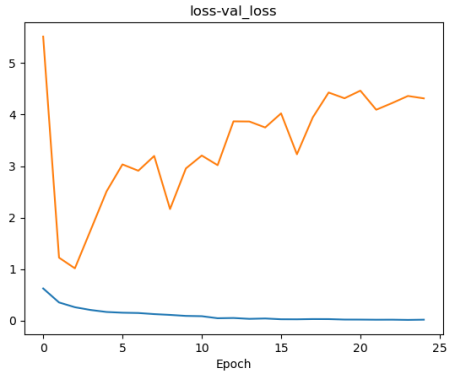

Fig. 10: BUN training and validation loss

	precision	recall	f1-score	support
covid	0.9231	0.9836	0.9524	305
normal	0.9028	0.9443	0.9231	305
pneumonia	0.9963	0.8852	0.9375	305

TABLE 4: BUN Classification Output

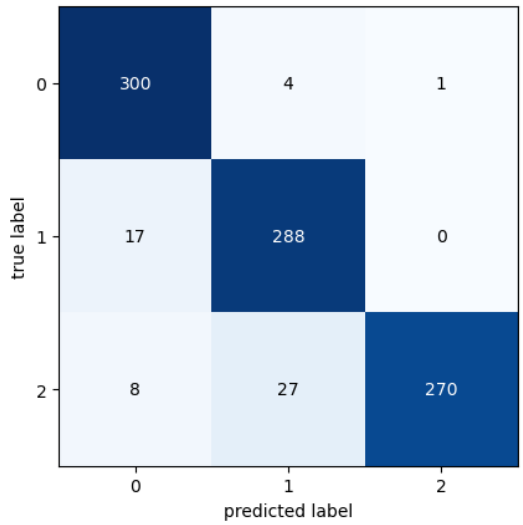

Fig. 11: BUN confusion matrix

According to the results above and the final test accuracy value we obtained, which is 94,07%, shows that our model performed a successful classification. This model has performed a more successful training and validation processs than the DenseNet 121 and the final accuracy result confirmes it.

Our third model, BUNTT, we used CNN with SENet and obtained the best results. Detailed measurements are given below.

- **Total params:** 6,609,010
- **Trainable params:** 6,604,946
- **Non-trainable params:** 4,064

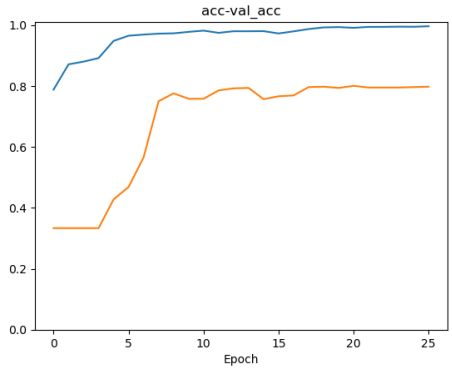

Fig. 12: BUNTT training and validation accuracy

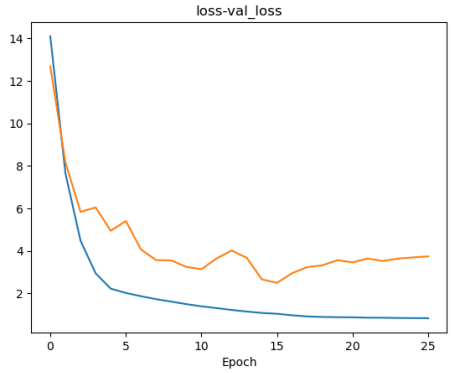

Fig. 13: BUNTT training and validation loss

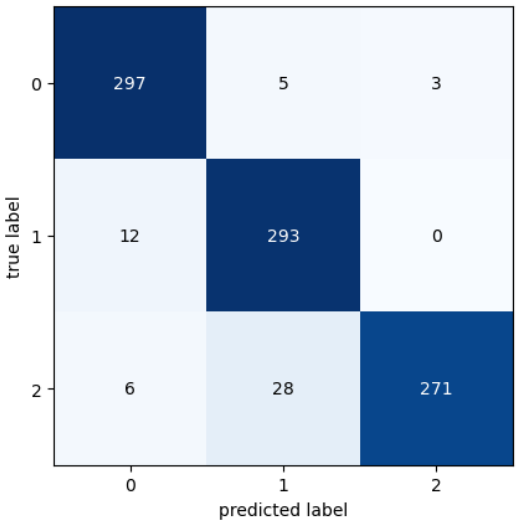

Fig. 14: BUNTT confusion matrix

Training Accuracy	0.9962431788444519
Training Loss	0.8335824608802795
Validation Accuracy	0.7978141903877258
Validation Loss	3.7450218200683594

TABLE 5: Last Epoch Parameters

	precision	recall	f1-score	support
covid	0.9429	0.9738	0.9581	305
normal	0.8988	0.9607	0.9287	305
pneumonia	0.9891	0.8885	0.9361	305

TABLE 6: BUNTT Classification Output

Finally, BUNTT has a test accuracy of 94.36%. It has performed more successful training and validation processes than DenseNet 121 and BUN.

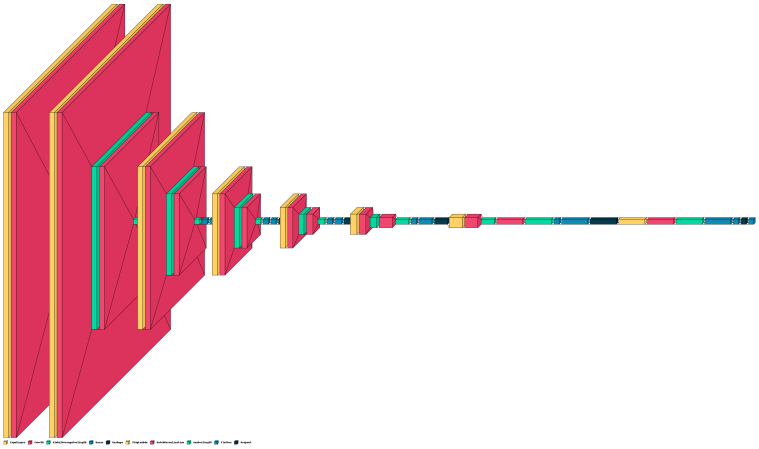

Fig. 15: BUNTT created layer image

Fig. 16: BUNTT created layer image

VII. Concluding Remarks

In this paper, we have addressed the important problem of lung disease prediction from X-ray images using deep learning techniques. We proposed a comprehensive approach that leverages transfer learning with DenseNet, incorporates SE blocks for enhanced feature representation, and integrates batch normalization and regularization techniques to improve model performance and generalization. Our approach showed promising results and better performance compared to traditional machine learning algorithms and simple image features in terms of accuracy and generalization.

The key contributions of our work include the utilization of transfer learning with DenseNet, we also introduced SE blocks to capture channel-wise dependencies and enhance feature representations, leading to more accurate predictions. Additionally, the incorporation of batch normalization and regularization techniques improved the model's robustness and generalization ability.

The model achieved higher accuracy and generalization, contributing to more reliable and effective disease diagnosis and treatment planning. The efficient model training and inference, enabled by transfer learning and the pre-trained DenseNet, make our approach suitable for real-time or resource-constrained scenarios.

In terms of future work, exploring other deep learning architectures and attention mechanisms could provide further insights into improving the accuracy of lung disease prediction. Moreover, investigating the interpretability of the model's predictions and understanding the learned features could provide valuable insights for medical professionals.

Throughout the project, we have witnessed the power of deep learning techniques in medical image analysis and their potential to revolutionize disease prediction and diagnosis. We have also realized the importance of carefully preprocessing and augmenting the data to improve model performance and reduce overfitting. Additionally, we have encountered challenges. When using Google Colab, the power and time of the processor it provided us was not enough, so the training took a long time, one epoch took about 4 minutes. After that, we continued using Kaggle, which had higher processing power performance, 1 epoch took about 1 minute. Also, during training, epochs reach the saturation point at 15-20, after that the network can't train itself but we can't stop the process, so we added an extra code. We noticed that network training did not improve when we kept the learning rate constant, so we used an adaptive learning rate.

Overall, this paper contributes to the field of medical diagnostics by presenting an efficient and accurate approach for lung disease prediction from X-ray images. The proposed techniques and methodologies can be further refined and extended to tackle other medical imaging tasks, leading to advancements in automated disease prediction and improving patient outcomes.

References

[1] L. Wang, Z. Q. Lin, and A. Wong, "Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images," *Scientific reports*, vol. 10, pp. 1–12, 2020.

[2] J. P. Cohen, L. Dao, K. Roth, P. Morrison, Y. Bengio, A. F. Abbasi, B. Shen, H. K. Mahsa, M. Ghassemi, H. Li, and T. Q. Duong, "Predicting covid-19 pneumonia severity on chest x-ray with deep learning," *Cureus*, vol. 12, p. e9448, Jul 2020.

[3] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Computer Science Department and BIOSS Centre for Biological Signalling Studies, University of Freiburg*, (Freiburg, Germany), pp. 1–122, Nov. 2015.

[4] J. Hu, L. Shen, and S. Albanie, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, June 2018.

[5] Z. Huang, X. Wang, Y. Wei, L. Huang, H. Shi, W. Liu, and T. S. Huang, "Ccnet: Criss-cross attention for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, p. 603.

[6] J. P. Cohen, P. Morrison, L. Dao, K. Roth, T. Duong, and M. Ghassem, "Covid-19 image data collection: Prospective predictions are the future," *Machine Learning for Biomedical Imaging*, vol. 1, pp. 1–38, dec 2020.