Integrating Auxilary Cross Entropy in the InceptionV3 Network for Automatic Detection and Classification of Lung Disease

Beyene Awlachew Shiferaw

A Thesis Submitted to the Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in

Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

September,2024

Adama,Ethiopia

Integrating Auxilary Cross Entropy in the Inception Network for Automatic Detection and Classification of Lung Disease

Beyene Awlachew Shiferaw

Advisor: Worku Jifara Sori(PhD)

A Thesis Submitted to the Department of Computer Science and Engineering

School of Electrical Engineering and Computing

Presented in Partial Fulfillment of the Requirement for the Degree of Master's in

Computer Science and Engineering

Office of Graduate Studies

Adama Science and Technology University

September,2024

Adama,Ethiopia

Declaration

I hereby declare that this Master Thesis entitled "Integrating Auxilary Cross Entropy in the Inception Network for Automatic Detection and Classification of Lung Disease" is my original work. That is, it has not been submitted for the award of any academic degree, diploma, or certificate in any other university. All sources of materials that are used for this thesis have been duly acknowledged through citation.

Beyene Awlachew	_____	_____
Name of student	Signature	Date

RECOMMENDATION

I, the advisor of this thesis, hereby certify that I have read the revised version of the thesis entitled "Integrating Auxilary Cross Entropy in the Inception Network for Automatic Detection and Classification of Lung Disease" prepared under my guidance by Beyene Awlachew submitted in partial fulfillment of the requirements for the degree of Mater's of Science in Computer Science and Engineering. Therefore, I recommend the submission of a revised version of the thesis to the department following the applicable procedures.

Worku Jifara(Ph.D) _____ _____

Major Advisor Signature Date

_____ _____ _____

Co-advisor/Co-supervisor Signature Date

APPROVAL SHEET

I, the advisor of the thesis entitle "Integrating Auxilary Cross Entropy in the Inception Network for Automatic Detection and Classification of Lung Disease" and developed by Beyene Awlachew, hereby certify that the recommendation and suggestions made by the board of examiners are appropriately incorporated into the final version of the thesis.

Worku Jifara (Ph.D) _____ _____

Major Advisor/Supervisor Signature Date

We, the undersigned, members of the Board of Examiners of the thesis by Beyene Awlachew have read and evaluated the thesis entitled "**Integrating Auxiliary Cross Entropy in the Inception Network for Automatic Detection and Classification of Lung Disease**" and examined the candidate during the open defense. This is, therefore, to certify that the thesis is accepted for partial fulfillment of the requirement of the degree of Master of Science in Computer Science Engineering

_____ _____ _____

Chairperson Signature Date

_____ _____ _____

Internal Examiner Signature Date

_____ _____ _____

External Examiner Signature Date

Finally, approval and acceptance of the thesis are contingent upon submission of its final copy to the Office of Postgraduate Studies (OPGS) through the Department Graduate Council (DGC) and School Graduate Committee (SGC).

_____ _____ _____

Department Head Signature Date

_____ _____ _____

School Dean Signature Date

_____ _____ _____

Office of Postgraduate Studies, Dean Signature Date

ACKNOWLEDGMENT

In the first place, I would offer my genuine gratitude to Almighty God, who has boundless gifts and opportunities for me. Besides, I might want to thank the Virgin Mary the Mother of God.

I would like to acknowledge and give my deepest gratitude to my advisor Dr. Worku Jifara (Ph.D.) for his guidance, brilliant comments, and support throughout my work.

Then, I would like to thank Getnet Yilma (Ph.D.) for his useful comments and suggestions to better this study.

Words cannot explain how much I appreciate and adore you, my wife Simegn Tesfaye, and my beautiful babies Nathan and Beniyas, together with all of my families, bring such wonderful energy into my life. You are a real source of motivation.

Table of Contents

ACKNOWLEDGMENT iv
List of table vii
List of Figure viii
Acronym ix
Abstract: x
CHAPTER ONE 1
INTRODUCTION 1
1.1. Background of the study 1
1.2. Motivation of the Study 3
1.3. Statement of Problem 3
1.4. Research question 4
1.5. Objective of the Study 4
1.5.1. General Objective 4
1.5.2. Specific Objective 4
1.6. Scope and Limitation of Study 4
1.6.1. Scope of the Study 4
1.6.2. Limitation of the study 4
1.7. Organization Of The Thesis 5
CHAPTER TWO 6
LITERATURE REVIEW 6
2.1. Review of Literature 6
2.2. Machine Learning 6
2.3. Deep learning 6
2.4. Conventional Neural Network 6
2.5. CNN's foundational element 7
2.6. CNN Architecture 9
2.7. Related works 12
2.8. Summary of related works 14
CHAPTER THREE 16
RESEARCH METHODOLOGY 16
3.1. Chapter Overview 16
3.2. Datasets 16

3.3. Preprocessing 16
3.4. Data Augmentation 17
3.5. Developments Tools 18
3.5.1. Hardware tool 18
3.5.2. Software tool 18
CHAPTER FOUR 19
PROPOSED ARCHITECTURE 19
4.1. Chapter Overview 19
4.2. Problem formulation 19
4.3. Baseline InceptionV3 19
4.4. Auxiliary entropy loss optimized InceptionV3 backbone network 20
4.4.1. Auxiliary entropy loss optimized InceptionV3 single backbone network 21
4.4.2. Auxiliary entropy loss optimized InceptionV3 dual backbone backbone network 22
4.5. Code segments for implementation of the proposed method 23
4.6. Code segments dual backbone data loader 23
4.7. Code segments single backbone data loader 24
4.8. Code segments dual backbone training 24
4.8.1. Code segments single backbone training 25
CHAPTER FIVE 26
EXPERIMENT RESULT AND DISCUSSION 26
5.1 Experiment configuration 26
5.2 Baseline InceptionV3 result 26
5.3 Baseline ResNet50 result 27
5.4 Baseline VGG19 result 28
5.5 Proposed entropy optimized InceptionV3 single backbone result 30
5.6 Proposed entropy optimized InceptionV3 dual backbone result 31
CHAPTER SIX 34
CONCLUSION AND RECOMMENDATION 34
6.1 Conclusion 34
6.2 Recommendation: 34
References 35

List of table

Table 1: summary of related works 15
Table 2. Lung disease dataset 16
Table 3. Hardware tools 18
Table 4. Aggregate Summary of the results (in %) 32

List of Figure

Figure 1. Baseline InceptionV3 backbone network 20
Figure 2. Auxiliary entropy loss optimized InceptionV3 single backbone network 21
Figure 3. Auxiliary entropy loss optimized dual backbone InceptionV3 network 22
Figure 4. Inception V3 baseline Backbone performance report 26
Figure 5. Inception V3 baseline backbone train/validation accuracy and loss curve. 27
Figure 6. ResNet50 classification performance 28
Figure 7. ResNet50 train/validation accuracy and loss curve 28
Figure 8. VGG19 classification performance 29
Figure 9. VGG19 train/validation accuracy and loss curve 29
Figure 10. Cross entropy optimized InceptionV3 backbone classification performance 30
Figure 11. Cross Entropy Optimized Inception V3 Backbone train/validation accuracy and loss curve. 31
Figure 12. Cross entropy optimized InceptionV3 dual backbone classification performance 31
Figure 13. Cross entropy optimized Inception V3 dual backbone train/validation accuracy and loss curve 32

Acronym

AIDS Acquired Immune Deficiency Syndrome

API Application Programming Interface

CAD Computer Aided Detection

CNNs Convolutional Neural Networks

CPU Central Processing unit

CT Computed Tomography

CXRs Chest X-rays

DL Deep Learning

GPU Graphics processing unit

HIV Human Immunodeficiency Virus

RAM Random Access Memory

TB Tuberculosis

WHO World Health Organization

Abstract:

Automated detection and classification of lung diseases from medical imaging data play a pivotal role in early diagnosis and treatment planning. In this study, we propose the integration of auxiliary cross entropy in the InceptionV3 network to enhance the accuracy and efficiency of lung disease detection and classification. By augmenting the InceptionV3 architecture with auxiliary losses, our aim is to improve the network's ability to learn discriminative features for differentiating between various lung disease patterns. Through the utilization of deep learning techniques and auxiliary supervision, we seek to develop a robust system capable of accurately identifying and classifying different lung diseases from x-ray images. Based on various models experiments, the results displayed both cross entropy optimized Inception V3 methods have outperformed other baselines models by a minimum of 1% accuracy for example in Inception V3 baseline, by a maximum of 16% in VGG19 baseline model. Therefore. We can conclude that the proposed cross entropy optimized Inception V3 strong enough to identify lung disease. Therefore, the experimental results demonstrate the efficacy of our approach in enhancing the performance of the InceptionV3 network for automatic detection and classification of lung diseases, paving the way for more precise and reliable medical image analysis in the field of pulmonary healthcare.

Keyword: deep learning, Lung disease, auxiliary cross-entropy loss, Inception V3, Dual Backbone

CHAPTER ONE

INTRODUCTION

1.1. Background of the study

Respiratory disorders also referred to as lung diseases; affect the airways and other lung components (Bousquet, J.,2007). Pneumonia, TB, and Coronavirus Disease 2019 (COVID-19) are a few lung diseases. According to the Forum of International Respiratory Societies (European Respiratory Society, Sheffield, UK, 2017), each year millions more people die from pneumonia, 1.6 million from lung cancer, 1.4 million from tuberculosis, and 334 million people suffer from asthma worldwide. The COVID-19 pandemic has caused millions of illnesses worldwide and put pressure on healthcare systems. (World Health Organization)(Rahaman, M.M.; Li, C.; Yao, Y.; Kulwa, F.; Rahman, M.A.; Wang, Q.; Qi, S.; Kong, F.; Zhu, X.; Zhao, X. 2020). It is evident that one of the main causes of death and disability worldwide is lung Disease. Improving long-term survival rates and improving the likelihood of recovery depend heavily on early identification (Yahiaoui, A.; Er, O.; Yumusak, 2017)(Hu, Z.; Tang, J.; Wang, Z.; Zhang, K.; Zhang, L.; Sun, Q, 2018). Traditionally, tests for skin, blood, sputum samples, chest X-rays, and computed tomography (CT) scans can be used to identify lung illness (Am. J. Respir. Crit. Care Med. 2000) (Setio, A.A.A.; Traverso, A.; de Bel, T.; Berens, M.S.; van den Bogaard, C.; Cerello, P.; Chen, H.; Dou, Q.; Fantacci, M.E.; Geurts, B.; et al. 2017). However, the manual interpretation of these images by healthcare professionals is time-consuming and prone to human error. Recent research has shown that deep learning holds great promise for using medical imaging to diagnose diseases, particularly lung conditions.

The field of "deep learning" in machine learning examines algorithms that are inspired by the composition and functions of the human brain. Medical visual pattern identification, measurement, and classification are made easier by recent developments in machine learning, particularly deep learning (Shen, D.; Wu, G.; Suk, H.I. 2017). Unlike hand-designed features drawn from domain-specific expertise, deep learning can learn features from data alone, which has allowed for these breakthroughs. Deep learning is quickly becoming the state of the art, leading to better results in a variety of medical applications. These advancements therefore make it easier for medical experts to recognize and classify specific medical illnesses (Wu, C.; Luo, C.; Xiong, N.; Zhang, W.; Kim, T.H., 2018).

Mycobacterium tuberculosis is the bacteria that cause tuberculosis (TB). Although the TB bacteria typically affect the lungs, they can also affect the kidney, spine, and brain, among

other parts of the body. Not every person who has the TB bacteria gets sick. Consequently, there are two TB-related conditions: latent tuberculosis infection and illness (www.cdc.gov/tb).

Latent TB Infection: The body can harbor the tuberculosis-causing bacteria even when no symptoms are present. This is known as a latent tuberculosis infection. Most people who breathe in the bacteria that cause tuberculosis and get infected are able to fight the bugs and stop it from growing. TB infection patients who are latent have a positive response on a TB skin test or blood test, usually show no symptoms, don't feel sick, and can't spread the TB bacteria to others (www.cdc.gov).

Often, latent infections with tuberculosis do not progress to tuberculosis disease. The tuberculosis (TB) bacteria in these people are dormant and do not produce symptoms for a lifetime. On the other hand, the bacteria multiply and cause tuberculosis in other people, especially in those with compromised immune systems (www.cdc.gov/tb).

TB Disease: When the immune system is unable to halt the growth of TB bacteria, the bacteria become active. TB illness is what happens when the TB bacteria are active, or multiplying within your body. Those who have tuberculosis are ill. They may also be able to spread the bacterium to persons they spend time with every day (www.cdc.gov/tb).

Latent tuberculosis infections frequently do not result in tuberculosis illness. Within weeks of contracting an infection, some patients get tuberculosis before their immune system has a chance to combat the TB germs. Years later, when their immune systems weaken for other reasons, additional people might become ill (www.cdc.gov/tb).

For people whose immune systems are weak, especially those with HIV infection, the risk of developing TB disease is much higher than for people with normal immune systems.

The TB skin test (TST) and TB blood tests are the two types of tests used to identify TB germs in the body. A positive TB skin test or blood test only indicates a TB bacterium infection in the subject. It is unable to determine if the patient has developed latent tuberculosis infection (LTBI) or tuberculosis illness. To determine whether a person has tuberculosis (TB) disease, more testing is required, such as a chest x-ray and a sample of sputum (www.cdc.gov/tb).

The majority of people who contract tuberculosis are from impoverished nations with limited healthcare resources and medical infrastructure. Over half of the deaths were due to late detection, which caused patients to miss out on the best treatment options. The death rate of tuberculosis can be considerably decreased, according to research, if the disease is diagnosed

and treated early. As a result, early detection of tuberculosis is crucial for illness prevention, disease transmission mitigation, and death rate reduction.

The most well-known method for detecting tuberculosis is computed tomography (CT). Because of the radiation dose, cost, availability, and ability to reveal previously undetected pathologic alterations, chest X-rays (CXRs) are used to confirm the earliest diagnosis of tuberculosis (TB). TB, on the other hand, is a more complicated disease with multiple manifestations, including consolidation (Adler D, Richards WF, 1953), effusion (Vorster MJ, Allwood BW, Diacon AH, Koegelenberg CF, 2015), fibrosis (Chung MJ, Goo JM, Im JG,2004), infiltration (Mishin V, Nazarova NV, Kononen AS, Miakishev TV,2006), mass (), nodule (Kant S, Kushwaha R, Verma SK,2007:8), and pleural thickening (Gil V, Soler JJ, Cordero PJ.,1994).

1.2. **Motivation of the Study**

The wide range of pathology makes lung disease detection more difficult, lowering the accuracy of doctors' judgment. Furthermore, without professional training and extensive experience, distinguishing It's challenging to distinguish between soft tissues with comparable textures and lung problems. Due to a lack of advanced medical infrastructure and funding for healthcare, there are very few professional radiologists trained in resource-poor and marginalized areas. Additionally, the quality of the CXRs generated is low, which negatively affects both the diagnostic accuracy and time delay for lung disease detection. Moreover, the substantial workload associated with interpreting CXRs may lead to weariness, which makes it increasingly harder for human professionals to perform their duties reliably..

As a result, methods that can help to reduce the time delay in Lung disease diagnosis while maintaining high quality and a low economic cost are required to improve treatment while also limiting disease reproduction at an early stage.

1.3. **Statement of Problem**

The automated detection and identification of lung diseases from medical imaging play a pivotal role in improving diagnostic accuracy, enabling early intervention, and enhancing patient outcomes. Traditional methods of disease detection often rely on manual interpretation of radiographic images, which can be time-consuming and prone to subjectivity. Deep learning techniques, such as convolutional neural networks (CNNs), have shown promise in automating the analysis of medical images, offering the potential for faster and more accurate diagnosis.

In this context, the problem addressed in this study is the need to enhance the performance of existing deep learning models, specifically the Inception V3 architecture, for automatically detecting and identifying lung diseases from chest X-ray images. While CNNs have demonstrated success in image classification tasks, integrating additional information sources, such as entropy, further improve the network's ability to extract relevant features and patterns related to various lung pathologies.

1.4. Research question

RQ1. To what extent entropy improve inception v3 performance to detect and classify lung disease?

RQ2. Which design entropy based feature selection method in the inception network for lung disease detection and classification?

1.5. Objective of the Study

1.5.1. General Objective

The general objective of integrating Entropy in the Inception Network for Automatic Detection and Classification of lung disease.

1.5.2. Specific Objective

The study address the following specific objectives in order to achieve the overall goal:

- To develop an entropy-based inception network for TB detection and classification.
- To evaluate the performance of the proposed method using a large dataset of chest X-ray images.
- To compare the performance of the proposed method with existing methods for TB detection and classification.

The suggested approach was contrasted with both conventional machine learning techniques and deep learning-based techniques that are currently in use for the identification and categorization of lung diseases. The comparison using different performance indicators, including F1-score, accuracy, precision, and recall.

1.6. Scope and Limitation of Study

1.6.1. Scope of the Study

The focus of this study is to create an Inception v3 model that uses entropy to detect and classify lung disease more accurately.

1.6.2. Limitation of the study

The limitation of the study includes:

- The dataset include only (Bacterial Pneumonia, Corona Virus Disease, Normal, Tuberculosis, Viral Pneumonia)

- This study does not incorporate imaging for external lung tuberculosis.

1.7. **Organization Of The Thesis**

This thesis' remaining sections are organized as follows:

Chapter 2: Review relevant publications and background literature

Chapter 3: Talk about research techniques like gathering datasets, preparing data, and using development tools.

Chapter 4: Talk about the training pseudocode, the Auxiliary Entropy Loss Optimized InceptionV3 backbone, and the suggested model design.

Chapter 5: Talk about the suggested model experiment's outcome and analysis Describe the results of the suggested solution, provide a comparison between the new and previous solutions, and interpret the results.

Chapter 6: all of the recommendations and conclusions

CHAPTER TWO

LITERATURE REVIEW

2.1. Review of Literature

The description of lung disease, the data preparation method for increasing the amount of training data, the CNN architecture's fundamental components, the examined Inception v3 model, and associated work are the main topics of this chapter. Additionally, the application of transfer learning to improve model performance is covered.

2.2. Machine Learning

The goal of machine learning, a branch of artificial intelligence, is to teach robots how to learn without much help from humans. The first computer learning application was a checkers game created by Arthur Samuel. As it played more, the application enhanced IBM's computer. After that, a number of machine learning technologies gained and lost popularity. Computer vision is a type of machine learning (ML) technology that enables machines to see, comprehend, and interpret what they see (Gollapudi, 2019). A domain expert's help is required when using machine learning techniques to identify the majority of the relevant characteristics in order to decrease data complexity and highlight trends for learning algorithms.

2.3. Deep learning

By examining computer methods, deep learning—a subset of machine learning—focuses on improving oneself. Deep learning reduces the need for extensive feature extraction and domain knowledge by gradually learning high-level features from data. Artificial neural networks (ANN), which are designed to resemble human thought and learning processes, are used in deep learning. Deep Learning (DL) has made significant strides in computer vision, especially with the CNN (Convolutional Neural Network) technique. LeNet (Lecun et al., 1998) was the first CNN designed by Yann LeCun to handle character recognition problems. Large datasets like ImageNet made it possible for AlexNet (Krizhevsky Alex et al., 2012) to develop an advanced CNN that could accomplish computer vision tasks that had previously been beyond the capabilities of computers.

2.4. Conventional Neural Network

CNN is a novel approach intended for both feature extraction and classification, according to (Khan et al., 2020a). CNN's convolution and pooling layers were employed to extract significant visual information. wherein the collected features from every prior neuron were fed into fully connected layers for mapping into the classification layer (Khan et al., 2020a). High-level feature extractors that use the lower-layer low-level features to learn high-level semantic features have been found to use deep learning methods. None of the affine transformations

have produced features that are invariant when extracted using the deep CNN. According to (Ren et al., 2016), one of the deep neural networks created for image identification, detection, and classification is the conventional neural network. As a result, a layer's neuron was linked to a tiny portion of the layer's predecessor layer. Additionally, CNN has been used with element-wise matrix multiplication and submission, or convolution. CNN is made up of numerous interconnected parts, including convolutional layers, pooling layers, activation functions, fully connected layers, Los functions, regularizations, and optimizations, as (Khan et al., 2020a) made evident.

2.5. **CNN's foundational element**

A. Classification

When it comes to various computer vision tasks like image reconstruction, object identification, and classification, CNN architecture performs remarkably well. Because these structures can extract remarkable traits, they are commonly used in the medical area for the classification of diseases. Because of its solid architectural context, using CNN's built-in architectures is also shown to be advantageous in a number of applications. The built-in architectures that are most frequently utilized include Inception, AlexNet, DenseNet, and ResNet models. However, because these models are trained on the ImageNet dataset, which contains general images, the use of these built-in architectures in the medical area is still restricted due to the domain shift (S. Z. Y. Zaidi, M. U. Akram, A. Jameel, and N. S. Alghamdi,2021).

B. Convolutional layer

CNN's fundamental building component is the convolution layer. Numerous kernels make up the convolution layer. Each neuron functions as a separate kernel. Convolution can be used by various kernels and filters to apply operations to images, including edge detection, blur, and sharpening. The convolution kernel divides images into tiny blocks and extracts features from each block. By multiplying each image's elements by the corresponding elements of the receptive region, the kernel can interact with images using a predetermined set of weights (Bouvrie, Jake.,2006).

C. Pooling layer

When a picture is too big, the pooling layer seeks to minimize the danger of overfitting by reducing the number of parameters. Additionally, it reduces memory utilization and computational stress. Sub-sampling and down-sampling are common terms used to describe

spatial pooling. Each map has its dimensions shrunk yet the important information remain the same. A variety of pooling formulations are used by CNN, such as overlapping, spatial pyramid pooling, L2-norm pooling, max pooling, sum pooling, average pooling, and more (Boureau, Y. L., et al., 2010); Wang, Tao, et al., 2012.); and K. He, X. Zhang, S. Ren, and J. Sun, 2015).

D. Activation function

An activation function, also known as a layer, is a node at the end of neural networks that serves as a decision-making function for understanding complex patterns. Selecting an efficient activation function can hasten the learning process. TanH and Sigmoid functions have been employed as the activation function in recent decades. ReLU is utilized in almost all CNN designs and is presently the most popular activation function worldwide. The vanishing gradient problem can be resolved with the use of ReLU and its modified versions (Nwankpa, Chigozie, et al., 2018).Since no neuron activates at the same time, the activation function of the ReLU is highly computationally efficient. In practice, ReLU is converging six times faster than TanH and Sigmoid.

E. Batch normalization

By shifting hidden values around (covariance shift), batch normalization increases the amount of unit values and speeds up deep learning processes. It is also considerably simpler to learn more independently from each network layer thanks to batch normalization. In batch normalization, each layer's characteristics are independently normalized with a mean of zero and a variance of one (Laurent, César, et al., 2016).

F. Padding

To maintain the original image size during a convolution, on the other hand, a zero-padding padding operation or zero-padding the original image's borders have been applied. The input volume's spatial dimensions have been shrunk too rapidly without any cushioning.

G. Stride Number

The number of pixels skipped when the convolutional filter or kernels slid over the input vector receptive fields was denoted by the stride number. Sections of the input vector with the same dimension as the convolutional kernel are known as receptive fields. The feature map or convolved features were generated by the element-wise dot products of the receptive fields and the kernels.

H. Fully Connected Layer

Eventually, a completely connected layer called the Flatten Layer was created, which was used to calculate the probability of each class's final output. Prior to applying the classifier layer (often Softmax), it was applied at the conclusion of the CNN model. The feature maps that were extracted in the final convolution or pooling layer were converted into a one-dimensional (1D) array of numbers using the fully connected layer.

A soft-max function was employed as the activation function in the last fully connected layer of a multiclass classification issue to normalize the output values of the final fully connected layer into target probabilities.

Different researchers have devised various optimization strategies that go beyond the activation layers. These optimization strategies were applied to reduce losses and improve model accuracy. Adaptive and non-adaptive optimization algorithms are the two types of optimization algorithms that are typically used. As shown by the stochastic gradient descent, Momentum, and Nester Momentum algorithms, the learning rate for non-adaptive optimization methods was constant for all features and required human selection. But as the AdaGrad, Adadelta, and Adam algorithms make evident, the learning rate in adaptive optimization algorithms was dependent on the presence of features. Consequently, in order to address the issue of manual learning rate schedules in non-adaptive optimization algorithms, adaptive optimization methods have been chosen.

I. Dropout

Hinton (2012) proposed the regularization technique known as dropout. that during training disregards neurons selected at random. Dropout makes it possible to learn more intricate features and about doubles the amount of iterations needed to converge.

2.6. CNN Architecture

Convolutional neural network (CNN) designs have developed over a number of years, becoming increasingly important in terms of accuracy and speed. Because of this, various academics have created various architectural designs at various points in time.

A. AlexNet

In order to improve the model's capacity for learning, Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton—Krizhevsky's doctoral advisor—worked together to develop and propose the AlexNet CNN. This CNN consists of 11 layers total, with each layer being created by

applying various optimization parameters. Between the input and output layers were three completely linked layers and three pooling levels, for a total of five convolution layers. The input layer, which defined the input dimensions initially, had an image size of 227 × 227 x 3, and the output layer served as the last layer in charge of classification (Khan et al., 2020b; Khan et al., 2020a).

ReLU was used as a non-linear activation function in the AlexNet architecture. However, for expedited architectural design, (Khan et al., 2020a) presented an efficient CNN architecture design named VGGNet. It was the 2014 ILSVRC's runner-up design. The primary contribution of the design was emphasizing how crucial network depth is to improving CNN's recognition and classification accuracy.

B. LeNet-5

LeNet-5 was one of the most well-known designs. Yann LeCun and associates introduced it in 1998 as a way to categorize handwritten numbers (Khan et al., 2020b). Three convolutional layers, two average pooling layers, one fully connected layer, and one output layer make up LeNet5's total of seven layers. Prior to a pooling process, nonlinearity was incorporated using the sigmoid function. RBF was integrated into the output layer to categorize ten digits. LeNet was the original CNN design that automatically extracted features from raw pixels while having fewer parameters.

However, the deep CNN LeNet5 performed poorly for all image classes and was only able to classify handwritten digits (Khan et al., 2020b).

C. VGGNet

The two convolutional layers that make up the VGGNet architecture both made use of the ReLU activation function. Additionally, a ReLU activation function was used to merge the completely connected layers and the single max-pooling layer. The SoftMax layer, which was used for classification, was the model's last layer. Compared to the AlexNet architecture, the VGGNet architectural design was more intricate. The technique has utilized a sequence of one or more convolutional layers with an equivalent number of filters, measuring 3x3 with an astride of 1x1, and the same amount of padding. As a result, the output size of every convolutional layer was equal to the input size for every filter. A rectified linear activation function and a max-pooling layer with a 2x2 size and astride of the same dimension are the next two layers in the VGGNet architecture. As an array of numbers (or vectors), the output feature maps of the last pooling layer were usually flattened and connected to one or more

fully connected layers, also referred to as dense layers, in which each input was linked to each output by a learnable weight (Khan et al., 2020a).

The primary problem with VGGNet, however, was that it required 138 million trainable parameters, making it computationally prohibitively expensive and challenging to use on systems with limited resources.

D. ResNet

In 2016, four researchers from Microsoft Research introduced ResNet (Residual Network) (He et al., 2016). The network structure can be further simplified by lowering the network parameters, and the gradient disappearance problem caused by a deep neural network's greater depth can be successfully resolved by connecting the remaining element to the element's output via a brief connection. This type of convolutional neural network (CNN) combines the input from the preceding layer with the output of the current layer. Performance is enhanced and the network's ability to learn is facilitated by this skip connection.

Semantic segmentation, object detection, and image classification are just a few of the applications where the ResNet architecture has proven useful. ResNets are also made up of layers; these networks' depth can be changed to represent space at any level. The large receptive fields that capture more information about each pixel in an image, the separation of the localization and classification stages, the effective encoding schemes with basic arithmetic operations, the computational efficiency at higher levels, and the increased accuracy as features are extracted further into the network are some of the factors that contribute to the success of the model. Convolutional neural network (CNN) architecture has advanced significantly since 1989. These improvements cover a wide range of topics, including regularization methods, structural improvements, and parameter optimization. Notably, the addition of new building blocks and the restructuring of processing units are primarily responsible for the increase in CNN performance. The improvements in depth and spatial usage are the main goals of CNN architecture. Based on the kinds of architectural alterations used, CNN architectures can be divided into seven groups. Spatial usage, depth, multi-path, breadth, feature-map exploitation, channel boosting, and attention-based CNNs are some of these categories.(Khan et al., 2020a).

E. Inceptionv3

Google created Inceptionv3, a convolutional neural network (CNN) with an astounding 50 layers deep. It was painstakingly built and trained, and the research publication "Going Deeper

with Convolutions" (Szegedy et al., 2015) has comprehensive details regarding its design and training procedure. With ImageNet weights, the pre-trained version of Inceptionv3 is capable of classifying a large number of objects, supporting up to 1000 distinct categories. Unlike the VGG19 network, Inceptionv3 can handle inputs of larger images—299 x299 pixels, to be exact. VGG19 placed second in the 2014 ImageNet competition, with Inception taking first place (Orhan G, 2020).

Inception-v3 is a convolutional neural network architecture from the Inception family that makes several improvements including using Label Smoothing, Factorized 7 x 7 convolutions, and the use of an auxiliary classifier to propagate label information lower down the network (along with the use of batch normalization for layers in the side head) (Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. 2016).

2.7. **Related works**

To address two types of problems in medical imaging, DeepCXRay was proposed as an automatic disease diagnosis tool. We used a unique CNN architecture with InceptionV3 to automatically extract features from X-ray pictures in order to achieve this goal. We employed an ImageNet-based model since no dataset was big enough to train the model. We have created a loss function known as CW-CEL in order to tackle the issue of imbalanced datasets. State-of-the-art results were obtained by combining the cross-entropy terms with the positive and negative weights. Experimental findings indicate that the suggested loss function performs significantly better when a dataset is significantly imbalanced (Xu, X., Guo, Q., Guo, J., & Yi, Z.,2018).

A computer-aided technique utilizing convolutional convolutional neural networks (CCNNs) to identify various types of tuberculosis lesions on radiographs. Resnet, GoogleNet, and Alexnet are used to transfer learning. We employ CNNs like UNET for segmentation, and VGG-16 pertained CNNs for feature extraction and classification. Data augmentation techniques are employed to extract more data from restricted datasets (Agnus James, Harisree P, Adarsh S, Vishnu S, 2021).

In order to train the TB images and increase detection accuracy, this research combines EfficientNet-B4, ResNet50, and ResNet-18 in conjunction with the DL approach and Unsharp Masking (UM) and High-Frequency Emphasis Filtering (HEF) image improvement. The results of the experiments demonstrated how competitive the suggested idea's accuracy is. Furthermore, we thoroughly compared the results with earlier studies in terms of the AUC and accuracy, and the suggested idea produced better results. TB detection via an image

enhancement Thus, the tested pre-trained network will be able to learn a better model if the TB photos are pre-processed using the DL technique. To demonstrate a more notable impact of enhancement on DL models, future research will assess additional picture enhancing methods. Also examined will be the medical expert's thorough subjective assessment and preference (K. Munadi, K. Muchtar, N. Maulina, and B. Pradhan, 2020).

Liu et al. (Liu C, Cao Y, Alcantara M, Liu B, Brunette M, Peinado J and Curioso W, 2018) designed a revised AlexNet and GoogleNet CNN models to detect TB manifestations in X-ray images with an accuracy of 85.68%.

An ensemble of three architectures, namely AlexNet, GoogleNet, and ResNet, is proposed by Hooda et al. (Hooda R, Mittal A and Sofat S ,2019) for TB detection. The ensemble architecture obtains the accuracy of 88.24%. (Ban et al.,2020). The use of deep learning in computer-aided tuberculosis diagnosis (CTD) was introduced by authors. With 11200 CXR pictures, the authors created a brand-new benchmark TB dataset (TBX11K). The authors suggested concurrent X-ray picture categorization and TB region detection using deep learning methods.

(Hwang s.Kim H-E, Jeong J and Kim H-J, 2016) proposed the first deep convolution neural net (CNN) based method for TB detection using the modified Alexnet network and transfer learning

This paper (Devnath L, Luo S, Summons P and Wang D, 2018) has proposed a TB detection method based on 14 layers CNNs architecture and data augmentation technique. This CNN model classifies CXR images of public datasets (MC, CH, IN) into TB positive and negative classes with an accuracy of 87.29%.

Ghorakavi (Ghorakavi and Srivatsav R ,2019) proposed a deep neural network ResNet18 with the data augmentation process used to increase the accuracy.

(S. J. Heo et al.,2017) D-CNN is proposed, which is a combination of Image CNN (I-CNN) and demographic factors such as age, height, weight, and gender. D-CNNs can predict TB from X-rays of the chest, and demographic variables can enhance the performance of tuberculosis diagnosis. D-CNN has accuracy more than I -CNN. Lakhani and Sundaram (Lakhani P and Sundaram B, 2017) proposed that the AlexNet and GoogleNet architectures are combined for TB detection.

(Nguyen et al.,2019) Shown that for TB detection using transfer learning the weights of ImageNet is insufficient, and the use of appropriate pre-training data is necessary.

(Sivaramakrishnan et al,2018) are compared five pre-trained DL models for TB detection from CXRs. Authors found that the performance of pretrained DL models is better than the customized model. CAD software, CAD4TB is commercially available software for TB detection.

The diagnostic performance of CAD software for the identification of PTB was thoroughly examined by the authors in (Pande T, Cohen C, Pai M, and Ahmad Khan F, 2016) in comparison to a microbiological reference test. Deep learning techniques served as the foundation for the development of the CAD4TBv6 program. The sensitivity of CAD4TBv6 software is high, with an area under curve ranging from 0.71 to 0.84.

Three Deep Learning (DL) based algorithms were examined by the authors in Qin Z Z, Sander M S, Rai B, Titahong C N, Sudrungrot S, Laah S N, Adhikari L M, Carter E J, Puri L, Codlin A J, and Creswell J, 2019 () to identify TB-associated anomalies in chest X-ray images. The three DL systems were examined by the authors in comparison. These are Lunit INSIGHT (V 4.7.2) for Chest X-ray picture developed by Lunit (South Korea), qXR (v.2) developed by Qure.ai, and CAD4TB (V.6) developed by Lunit. CXR image files are read in the DICOM format by Lunit and CAD4TB. For TB detection, the CAD4TB output falls within the anomaly score range of 0 to 100. Cavitation, nodule, pneumothorax, and other lung abnormalities are recognized by qXR and Lunit from the CXR image. Three Deep Learning-based software programs, CAD4TB, qXR, and Lunit, fared better at distinguishing between patients with tuberculosis and those who did not than skilled human readers. Summary of related works

Table 1: summary of related works

Authors	Title	Methods	Gaps
Bogomasov, Kirill, et al.	Feature and Deep Learning Based Approaches for Automatic Report Generation and Severity Scoring of Lung Tuberculosis from CT Images	They directly use U-net and VGG19 as classification networks CNN models	They get an accuracy of 0.6923 and it is too low for clinical use.
Lopez-Garnier, Santiago, Patricia Sheen, and Mirko Zimic,	Automatic diagnostics of tuberculosis using convolutional neural networks analysis of MODS digital images	train and evaluate CNN for automatic interpretation of MODS cultures digital images	It is hard for hospitals in remote areas to get patients' MODS images, so their method is not suitable for poor areas
Pasa, F., et al.	Efficient deep network architectures for fast chest x-ray tuberculosis screening and visualization	They use a CNN model which consists of 5 convolutional blocks, followed by a global average pooling layer and a fully-connected softmax layer with two outputs	Their CNN model is truly simple, but they get low accuracy
Panicker, Rani Oomman	Automatic detection of tuberculosis bacilli from microscopic sputum smear images using deep learning methods	By image binarization and subsequent classification of detected regions using a CNN	Different TB patients have different symptoms, someone may not cough and have no phlegm. What's more, they get 78.4% precision and it is not enough for TB detection
Xiong, Yan, et al.	Kant, Sonaal, and Muktabh Mayank Srivastava,	Towards automated tuberculosis detection using deep learning	

CHAPTER THREE

RESEARCH METHODOLOGY

3.1. Chapter Overview

This chapter describes the study methodology in detail, covering baseline work, evaluation measures, development tools, augmentation techniques, preprocessing procedures, and datasets used.

3.2. Datasets

This is extracted from publicly available datasets from kaggle[i]. The dataset contains five class x-ray images. The class labels such as Bacterial Pneumonia, Corona Virus Disease, Normal, Tuberculosis, Viral Pneumonia. The full detail is also shown in Table 2.

Table 2. Lung disease dataset

Arthritis Types	Train	Val(10%)	Test(15%)
Bacterial Pneumonia	1205	121	175
Corona Virus Disease	1218	122	175
Normal	1207	121	175
Tuberculosis	1220	122	176
Viral Pneumonia	1204	120	175

3.3. Preprocessing

An essential step in the suggested model to enhance the efficiency of lung disease detection and classification was picture preprocessing. By improving their quality, the input photos are now better suited for training the model. It includes a number of processes, including data augmentation, normalization, and resizing.

To guarantee that the model receives inputs of a consistent size, it was necessary to resize the input photos to a standard size (Fadnavis, 2014). This streamlines and enhances the training procedure. Normalization improves consistency in the training data by helping to standardize the pixel values. It is crucial to carry out specific preprocessing actions in order to get the pictures ready for training and testing in the suggested. In order to ensure that every image has the same height and breadth, the first step in this process is to resize each image to a consistent size of 128X128. This is significant since the model is made to accept inputs of a particular size.

The photos' pixel values should be normalized as another crucial preparation step. In other words, the pixel values must be converted to the range [0, 255]. Standardizing the input data can aid in enhancing the model's performance. Spatial filtering techniques that are capable of

removing Gaussian noise are also used for Gaussian noise reduction. These preprocessing procedures are also essential for enhancing the suggested model's performance and accuracy in identifying lung diseases.

3.4. **Data Augmentation**

By randomly altering the original photos, a technique known as data augmentation produces more training samples (Shorten & Khoshgoftaar, 2019a). We need a huge data collection for deep learning since it helps to enhance the quantity and diversity of the training data, making the model more resilient and capable of diagnosing cattle diseases under varied settings. We therefore use data augmentation. To expand the quantity of training data points in a dataset, the process of creating new data from an existing training sample is known as data augmentation. Increasing the number of data points is essential. Additionally, it is employed to extract complex characteristics from data without running the risk of overfitting. As a data-space solution, data augmentation addresses the issue of limited data. The term "data augmentation" describes a range of methods for increasing the quantity and caliber of training datasets in order to build more potent deep learning models. To add more images to our data set, different data augmentation techniques were applied to the original photos in this thesis. Numerous augmentation techniques exist, including shifting, brightness and contrast adjustments, noise injection, color modification, rotation, cropping, and noise reduction.

Data augmentation helps improve the efficacy and outputs of deep learning models by generating more and varied cases to train datasets. Before and after the data is given into our model (Image-Data Generator augmentation), as well as during the training phase, data augmentation can be performed. In this thesis, data augmentation is carried out during network training using Keras libraries. Every image that was fed into the network during training comes from the original image. By specifying a set of random modifications to be performed to the images that the Image-Data-Generator instance reads, this may be done in Keras. We have also utilized an augmenter from the imgaug library.

3.5. Developments Tools

3.5.1. Hardware tool

The hardware tools used for the research are listed in the table below.

Table 3. Hardware tools

Number	Device Name	Usage for the research
1	CPU	To train the model
2	GPU	To train the model
3	Hardisk	To store the datasets

3.5.2. Software tool

Software tools used to write the code, debug it, visualize the results, and document the research process for reporting purposes.

Package management and deployment are made easier with the help of Anaconda, a R and Python programming language distribution. Because it only requires a single installation to deliver a wide range of tools required in machine learning and data analysis, Anaconda is incredibly popular. The package manager for Anaconda, conda, facilitates the installation of libraries. It also has a useful interface with pip that makes it possible to install libraries that are not available through the anaconda package manager. It can be used on Windows, macOS, or Linux because it is also cross-platform.

Jupyter notebook: An online program that is available to the public that combines computer code, documentation, results from calculations, and multimedia into one cohesive document. It is used for code editing and result viewing, and it is integrated with the Anaconda environment.

PyTorch is an open-source machine learning library based on the torch library, which is used in computer vision and natural language processing applications. Pytorch is almost easier to understand since it shares a syntax with the Python programming language.Tensorflow is an artificial intelligence platform that can be used to create models and implement them in client contexts, much like other real-world applications. It facilitates deep learning, machine learning, and adaptable numerical computing**.**

CHAPTER FOUR

PROPOSED ARCHITECTURE

4.1. Chapter Overview

This research is focused on detection and classification of Lung diseases by integrating Convolutional Neural Networks, with top accuracy image classification(Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. 2016]. Auxiliary cross entropy classifier improved network training convergence via pushing gradients deeper. However, the impact of auxiliary entropy loss on classification performance metrics such as accuracy can be improved via directly adding the auxiliary cross entropy loss to the original cross entropy loss function. Since the auxiliary is applied at each iteration to predict auxiliary classification labels, the actual cross entropy loss class prediction can be improved at each epoch. Therefore, including auxiliary cross entropy into the inception model could improve accuracy more.

4.2. Problem formulation

We formulate the proposed cross entropy integrated auxiliary classifier in the InceptionV3 backbone (Gao et al., 2021)(Gao et al., 2021) taking x-ray images as training sample images with height, width and channel R^{HxWxC}, $X_i = \{x_i, y_i\}_{i=1}^{N}$, where $x_i$represent x-ray image i and y_i is the corresponding target label for the x-ray image. InceptionV3 model function is denoted as $f_\theta(X_i, y_i)$. θ is learning parameter, X_i training sample images, y_i the class labels.

$$y = f_\theta(X_i, y_i)\text{Concat}\begin{pmatrix}\text{ReLU}\left(\text{Conv1x1}\left(\text{BN}(X_i)\right)\right), \text{ReLU}\left(\text{Conv3x3}\left(\text{BN}(X_i)\right)\right), \\ \text{ReLU}\left(\text{Conv5x5}\left(\text{BN}(X_i)\right)\right), \text{MaxPooling}(X_i)\end{pmatrix} \quad (1)$$

$$y = f_\theta(X_i, y_i) = \text{Dropout}(f_\theta(X_i, y_i)) \quad (2)$$

4.3. Baseline InceptionV3

InceptionV3 developed by Google, is designed for image classification and object recognition tasks. Improved from previous versions of InceptionV1, InceptionV3 has introduced improvements such as use of different filter sizes (1x1, 3x3, 5x5) to capture multiple scale features and reduce the number of parameters. Further InceptionV3 incorporated architectural enhancements batch normalization, factorized 7x7 convolutions, and auxiliary cross entropy classifiers. Batch normalization stabilizes and accelerates the training process by normalizing the inputs to each layer, while factorized convolutions split large filters into smaller ones, reducing computational cost. The inclusion of auxiliary classifiers at intermediate layers provides additional supervision signals during training, encouraging the learning of more

discriminative features (Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. 2016). InceptionV3 consists of multiple stacked Inception modules followed by global average pooling and a fully connected layer for classification. With approximately 48 layers and more than 23 million parameters. The detailed InceptionV3 backbone architecture in given in Figure 1.

The cross entropy classification $actual_{loss}$ for the baseline InceptionV3 model is given by

$$actual_{loss} = -\sum(y_{true} \times \log(y_{pred})) \qquad (3)$$

Where, y_{true} is the actual class label, y_{pred} is the predicted class label from the learned features.

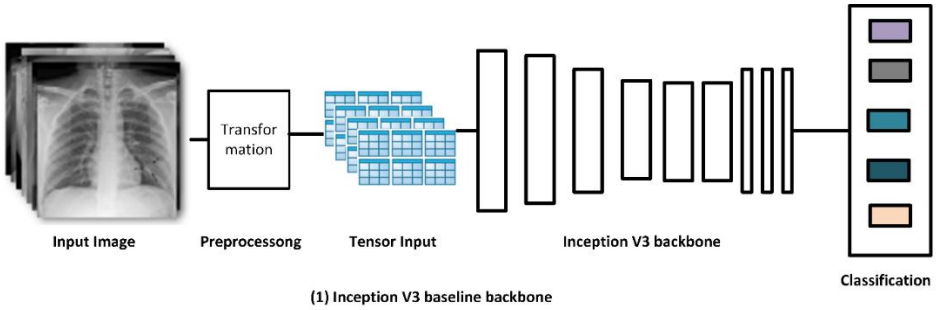


Figure 1. Baseline InceptionV3 backbone network

Inception v3 is built from several stacked inception modules, whereas the next version of inception v4 constructed using both inception modules and residual units from ResNet. So we specifically aim to enhance only inception modules via auxiliary classification loss. Further more, the fact that inception v4 tiny performance difference compared to inception v3 and inception v3 extensive evaluation on medical image classification motivated us to use inception v3 (Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A., 2017).

4.4. **Auxiliary entropy loss optimized InceptionV3 backbone network**

Using InceptionV3 backbone cross entropy to classify lung disease is a normal procedure. However, integrating auxiliary loss to screen out high classification probabilities boosts the inception v3 backbone classifier while reducing vanishing gradient and improving convergence. Auxiliary entropy loss tries to filter, high value features to pass to the fully connected layer optimizing the performance of the proposed model.

Therefore, the proposed InceptionV3 model has two losses. The first loss is auxiliary cross entropy loss which has optimization function via auxiliary class prediction and the second cross entropy loss has classification function.

To evaluate the inception v3 backbone performance we presented two design options. The first is single inception v3 backbone with auxiliary cross entropy loss as optimization. The second is dual InceptionV3 backbone with auxiliary cross entropy loss as optimization.

4.4.1. Auxiliary entropy loss optimized InceptionV3 single backbone network

In the single backbone classifier one batch of input is given to the backbone network, the learned feature before the fully connected layer is given to the auxiliary cross entropy loss to filter high value target predictions in each batch shown in Figure 2. There can be more class prediction than the target lung disease classes which are 5. Auxiliary predictions assist the InceptionV3 backbone either to penalize the backbone network when there is low prediction or boost classification performance of the Inception backbone when there is high auxiliary prediction. The optimized feature after several iterations is passed to the fully connected layer for lung disease classification purpose. From equation (3) we have the original cross entropy loss, now the auxiliary cross entropy loss is given by

$$loss_{aux\_i} = -\sum_{i=1}^{b}(y_{true\_i} \times \log(y_{pred\_i})) \qquad (4)$$

The $loss_{aux\_i}$ is the auxiliary predictions in each iteration of batch b. The sum of all the auxiliary prediction of each iteration i to b will give us the batch prediction denoted $loss_{aux}$.

$$loss_{aux} = \sum_{i=1}^{b} loss_{aux\_i} \qquad (4)$$

The total loss $Total_{loss}$ is given by $Total_{loss} = loss_{actual} + \alpha \times loss_{aux}$ (5)

Where α is a weighting factor that determines the relative importance of the auxiliary classifier loss compared to the actual classifier loss. It is typically set to a small value such as 1 to balance the contributions of the two losses.

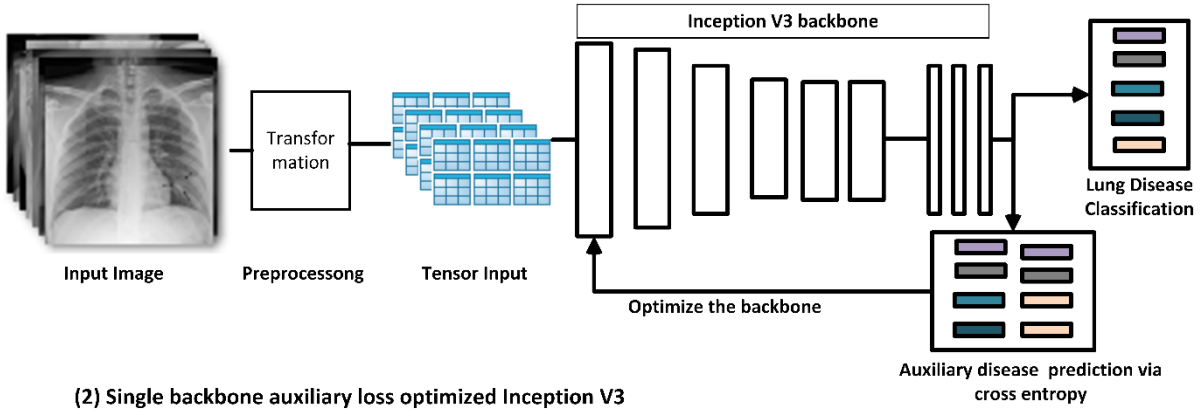


Figure 2. Auxiliary entropy loss optimized InceptionV3 single backbone network

4.4.2. Auxiliary entropy loss optimized InceptionV3 dual backbone network

In the dual or Siamese backbone InceptionV3 backbone has two InceptionV3 branches with the same model structure. The Siamese backbone takes two inputs. One is original training sample and the other branch takes in strongly augmented input images as input. The two backbone features y_1 and y_2 combined together via concatenation y_{cat} as one and is given to the auxiliary cross entropy loss $loss_{aux\_i}$ to optimize lung disease classification performance. The weakly augmented input backbone network feature is directly passed to the classification layer loss $loss_{actual}$ for lung disease classification purpose. The dual or Siamese backbone InceptionV3 backbone is described in Figure 3.

$$y_{cat} = cat(y_1 + y_2) \quad (6)$$

$$y = f_\theta(X_i, y_i) = Dropout(f_\theta(y_{cat\_i}, y_i)) \quad (7)$$

$$loss_{aux} = -\sum_{i=1}^{b}(y \times \log(y_{pred})) \quad (8)$$

$$Total_{loss} = loss_{actual} + \alpha \times loss_{aux} \quad (9)$$

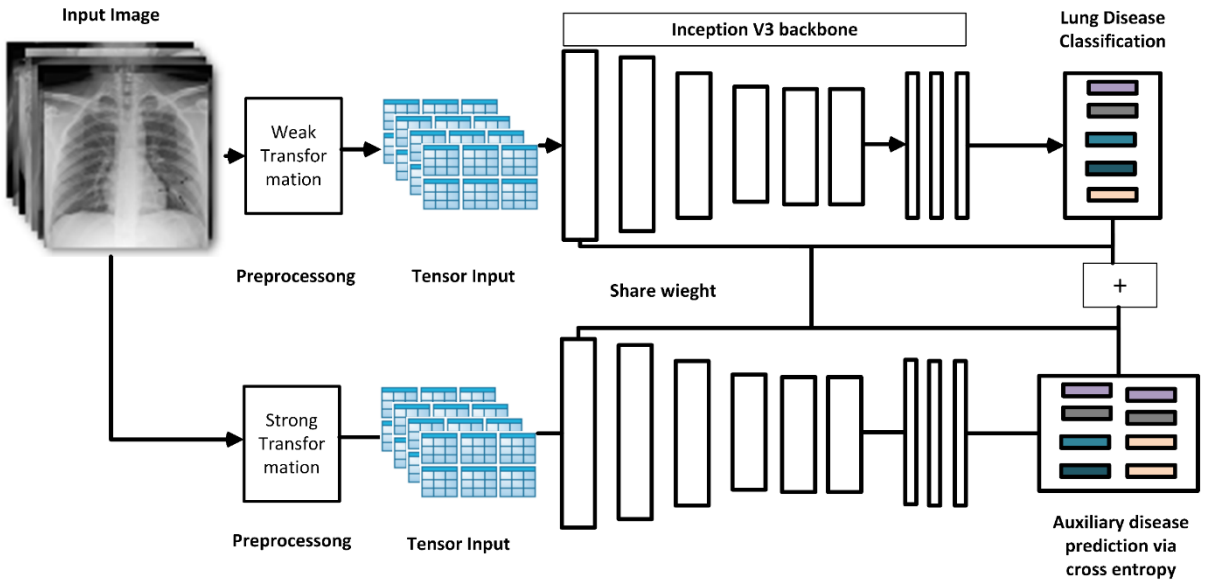


Figure 3. Auxiliary entropy loss optimized dual backbone InceptionV3 network

4.5. **Code segments for implementation of the proposed method**

4.6. **Code segments dual backbone data loader**

The following code segments shows two data loaders for the Inception dual backbone network. The loader creates weak augmentation and strong [more distorted] augmentation method.

```
class DualTransformDataset(datasets.ImageFolder):
    def __init__(self, root, transform=None, strong_transform=None):
        super(DualTransformDataset, self).__init__(root, transform=transform)
        self.strong_transform = strong_transform

    def __getitem__(self, index):
        path, target = self.samples[index]
        image = self.loader(path)
        if self.transform is not None:
            weak_image = self.transform(image)
        if self.strong_transform is not None:
            strong_image = self.strong_transform(image)
        return (weak_image, strong_image), target

# Define transformations
def check_min_size(size):···

weak_transform = T.Compose([
    # T.Lambda(check_min_size((120, 120))),
    T.RandomCrop(112),
    T.ToTensor(),
    T.Normalize(mean, std),
])

strong_transform = T.Compose([
    # T.Lambda(check_min_size((120, 120))),
    T.RandomCrop(112),
    T.RandomHorizontalFlip(),
    T.ColorJitter(brightness=0.5, contrast=0.5, saturation=0.5, hue=0.1),
    T.RandomAffine(degrees=15, translate=(0.1, 0.1), scale=(0.8, 1.2), shear=10),
    T.ToTensor(),
    T.Normalize(mean, std),
])
```

4.7. **Code segments single backbone data loader**

```python
def create_dataloader(data_dir, batch_size, transform, shuffle=True):
    # Create a dataset using ImageFolder
    dataset = datasets.ImageFolder(root=data_dir, transform=transform)

    # Create a DataLoader
    dataloader = DataLoader(dataset, batch_size=batch_size, shuffle=shuffle, num_workers=4)
    return dataloader

def get_trainloader(args):
    train_transform = T.Compose([
        # T.Lambda(check_min_size((120, 120))),  # Check and pad image size if needed
        T.RandomResizedCrop(112),
        T.RandomHorizontalFlip(),
        T.ToTensor(),
        T.Normalize(mean, std),
    ])
    return create_dataloader(args.train_dir, batch_size=args.batch_size, transform=train_transform)
```

4.8. **Code segments dual backbone training**

```python
def train_model(model, train_loader, val_loader, test_loader, criterion, optimizer, scheduler, epochs, device):
    for batch in tqdm(train_loader, total=len(train_loader), desc='Training'):
        (images_weak, images_strong), labels = batch # torch.Size([96, 3, 112, 112]),  torch.Size([96])

        labels = labels.to(device)
        inputs = torch.cat((images_weak, images_strong)).to(device)
        optimizer.zero_grad()

        logits = model(inputs) # logits: torch.Size([96, 5])
        logits_w, logits_s = logits.chunk(2) # logits_w: torch.Size([48, 5]), logits_s: torch.Size([48, 5])
        loss = criterion(logits_w, labels)
        aux_pl = torch.softmax(logits_s, dim=-1) # pl: pseudo_label
        # Filter the max probabilities for each class and their corresponding index as targets
        max_probs, targets = torch.max(aux_pl, dim=1)
        """
        # In case of balanced class dataset, use confident classes as a mask to calibrate the aux_ce_loss
        mask = max_probs.ge(threshold).float()
        aux_ce_loss = (F.cross_entropy(aux_predictions, targets, reduction='none') * mask).mean()
        """
        aux_ce_loss = (F.cross_entropy(logits_s, targets, reduction='none') * max_probs).mean()
        total_loss = loss + 1.0*aux_ce_loss # multiply by param mu=1.0 to control the auxiliary loss
        aux_losses.append(aux_ce_loss.item())
        train_losses.append(total_loss.item())

        total_loss.backward()
        optimizer.step()
        scheduler.step()

    avg_aux_loss = torch.tensor(aux_losses).mean().item()
    avg_train_loss = torch.tensor(train_losses).mean().item()
    avg_train_accuracy = torch.tensor(train_accuracies).mean().item()
    print(f'Epoch {epoch + 1}: Train Loss: {avg_train_loss:.4f}, Auxiliary Loss: {avg_aux_loss:.4f}, Train Accuracy:
```

4.8.1. Code segments single backbone training

```
def train_model(model, train_loader, val_loader, test_loader, criterion, optimizer, scheduler, epochs, device):
    for epoch in range(epochs):
        model.train()
        train_losses = []
        aux_losses = []
        train_accuracies = []
        for batch in tqdm(train_loader, total=len(train_loader), desc='Training'):
            images, labels = batch
            images, labels = images.to(device), labels.to(device)
            optimizer.zero_grad()
            predictions, aux_predictions = model(images) # torch.Size([64, 5])
            loss = criterion(predictions, labels)
            # Calculate main task accuracy
            acc = accuracy(predictions, labels) # [0]
            train_accuracies.append(acc)
            aux_pred_pseudo_labels = torch.softmax(aux_predictions, dim=-1)
            # Filter the max probabilities for each class and their corresponding index as targets
            max_probs, targets = torch.max(aux_pred_pseudo_labels, dim=1)
            """
            # In case of balanced class dataset, use confident classes as a mask to calibrate the aux_ce_loss
            mask = max_probs.ge(threshold).float()
            aux_ce_loss = (F.cross_entropy(aux_predictions, targets, reduction='none') * mask).mean()
            """
            aux_ce_loss = (F.cross_entropy(aux_predictions, targets, reduction='none') * max_probs).mean()
            total_loss = loss + 1.0*aux_ce_loss # multiply by param mu=1.0 to control the auxiliary loss
            aux_losses.append(aux_ce_loss.item())
            train_losses.append(total_loss.item())
            total_loss.backward()
            optimizer.step()
            scheduler.step()
        avg_aux_loss = torch.tensor(aux_losses).mean().item()
        avg_train_loss = torch.tensor(train_losses).mean().item()
        avg_train_accuracy = torch.tensor(train_accuracies).mean().item()
        print(f'Epoch {epoch + 1}: Train Loss: {avg_train_loss:.4f}, Auxiliary Loss: {avg_aux_loss:.4f}, Train Accuracy:
```

CHAPTER FIVE

EXPERIMENT RESULT AND DISCUSSION

5.1 Experiment configuration

This section discusses experimental configurations, extensive experiments and result discussions. To explore the performance of the proposed attention augmented methods and other baselines on various evaluation metrics, we applied common configuration settings. We used 128×128 image size as input with an 80:15 train test split ratio and 10% of the train as validation. Pytorch platform packages are used for deep learning experiment. $lr = 0.01$, batch=64, epoch=50 momentum=0.9, weight decay= 0.001, optimizer=SGD, loss=categorical cross-entropy.

5.2 Baseline InceptionV3 result

The baseline InceptionV3 experiment classification performance result is presented in Figure 4. (a) the classification report result the accuracy is 94% Baseline InceptionV3 backbone results. As shown from the result the Baseline Inception V3 has shown good classification result giving high confidence in lung disease classification. (b) the confusion matrix result where there are a total of 54 confusions out of 876 test samples which is good performance. Therefore, InceptionV3 baseline model has the ability to classify lung disease with high accuracy and low confusion.

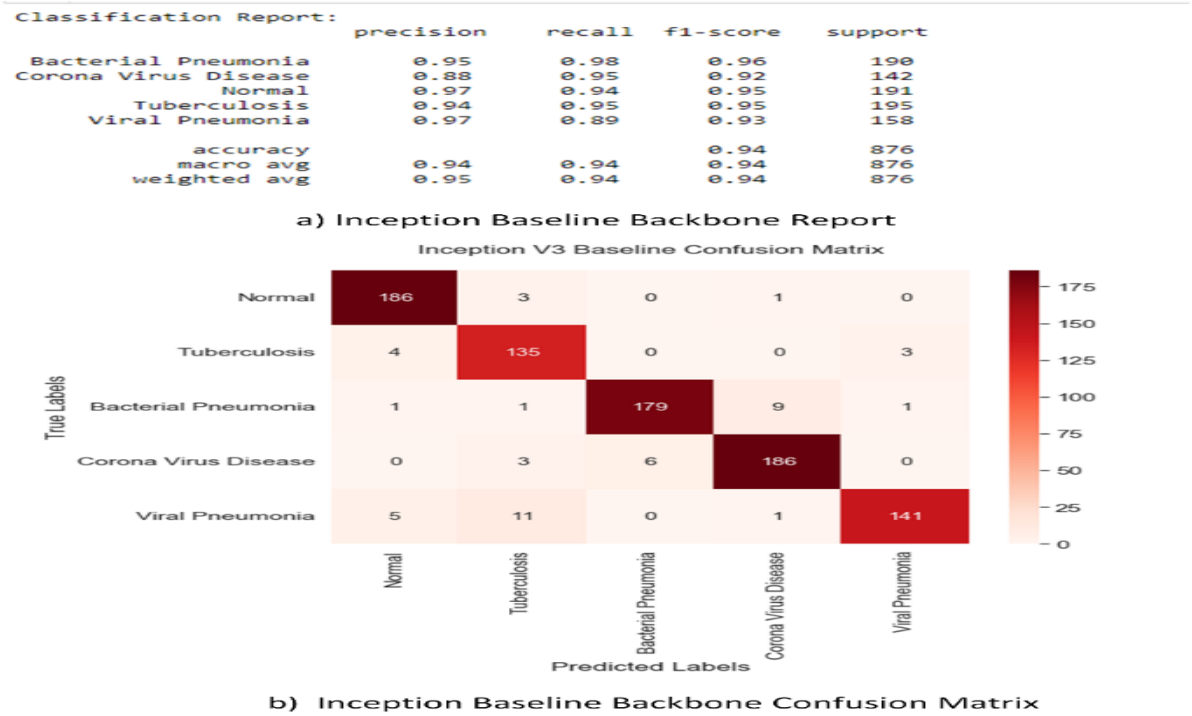


Figure 4. Inception V3 baseline Backbone performance report

From Figure 5 the (a) train validation accuracy curve, we can see both the curves have been increasing up to epoch <=6 and started to remain constant up to the last epoch 50. This is one of the indication of over fitting in which case the model has stopped learning too early and the model has able to memorize the training dataset. Similar behavior is observed in the loss curve.

a) Inception Baseline Backbone train validation accuracy Curve

b) Inception Baseline Backbone train validation loss Curve

Figure 5. Inception V3 baseline backbone train/validation accuracy and loss curve.

5.3 Baseline ResNet50 result

ResNet50 backbone experiment classification performance result is presented in Figure 6. (a) the classification report result the accuracy is 88% ResNet50 backbone results. As shown from the result the Baseline ResNet50 has shown lower classification performance result for lung disease classification. (b) the confusion matrix result where there are a total of 102 confusions out of 876 test samples which is lower performance compared to the baseline InceptionV3 result. Figure 5 (b), the confusion matrix demonstrates 57 images of corona virus disease are wrongly classified as viral pneumonia and 24 images of viral pneumonia confused as corona virus disease. This indicates ResNet50's inability to differentiate between corona virus and viral pneumonia. However, both the train and val accuracy and loss curves given in Figure 6, shows the ResNet50 backbone learning curves. The accuracy curve consistently increased and loss curves consistently decreased.

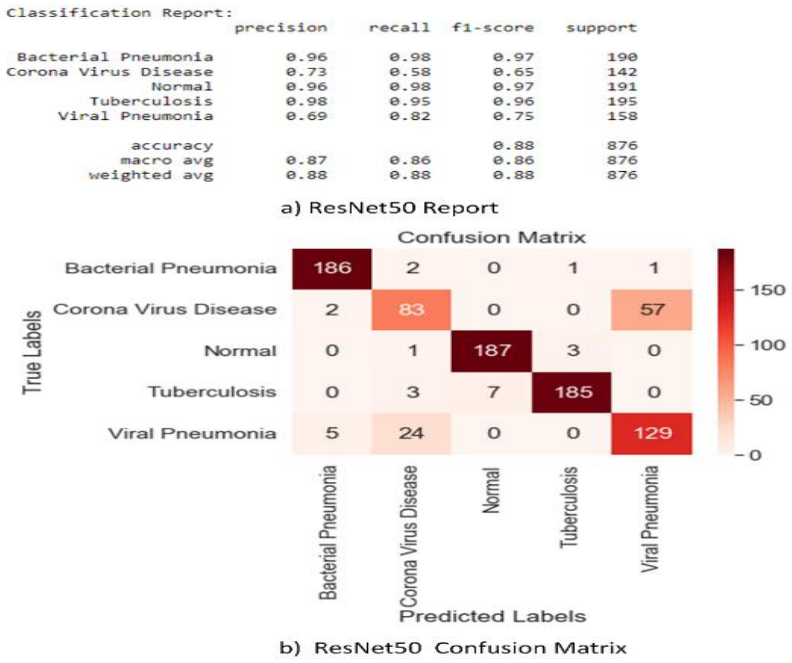


Figure 6. ResNet50 classification performance

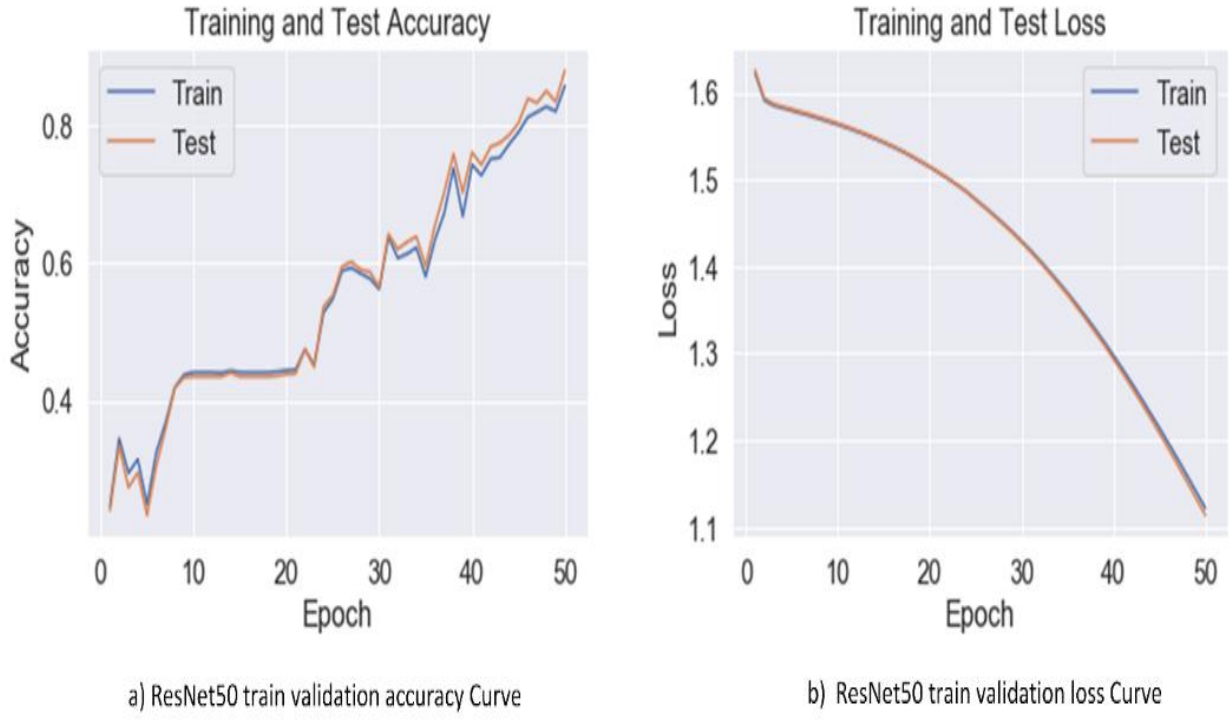


Figure 7. ResNet50 train/validation accuracy and loss curve

5.4 Baseline VGG19 result

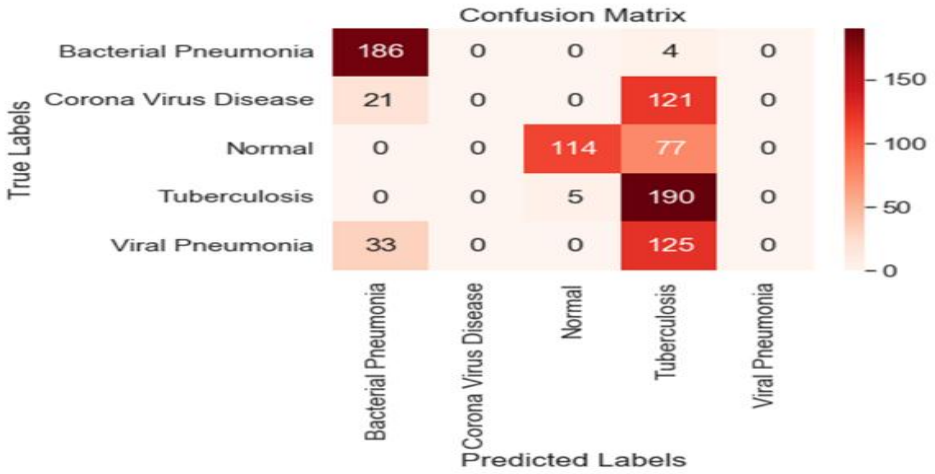
VGG19 backbone experiment classification performance result is presented in Figure 8. (a) the classification report result shows accuracy of 81%. As shown from the result the Baseline VGG19 has shown lower classification performance result for lung disease classification. (b) the confusion matrix result where there are a total of 380 confusions out of 876 test samples

which is lower performance compared to the baseline InceptionV3 and ResNet50. Figure 8(b) the VGG19 has completely failed to recognize corona virus disease and most of the images are confused as tuberculosis disease. VGG19 reported the lowest performance compared to other models. The train and validation accuracy and loss curves given in Figure 9, shows the VGG19 backbone learning curves. The accuracy curve consistently increased and loss curves consistently decreased showing no sign of over fitting.

Classification Report:

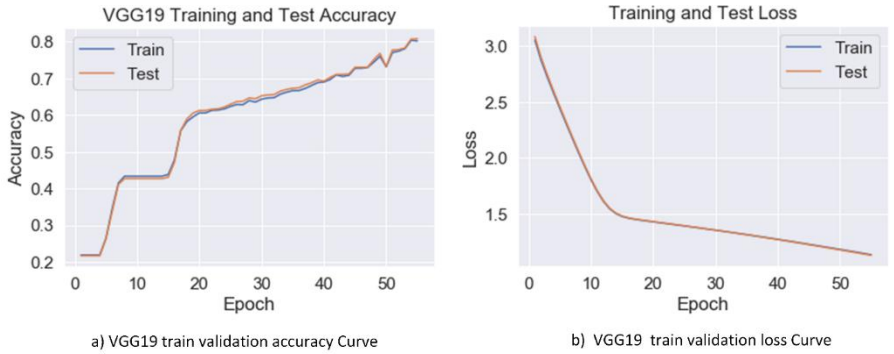
	precision	recall	f1-score	support
Bacterial Pneumonia	0.71	0.99	0.82	190
Corona Virus Disease	0.91	0.59	0.72	142
Normal	0.71	0.98	0.82	191
Tuberculosis	0.98	0.95	0.96	195
Viral Pneumonia	0.97	0.40	0.57	158
accuracy			0.81	876
macro avg	0.86	0.78	0.78	876
weighted avg	0.85	0.81	0.79	876

a) VGG19 Report



b) VGG19 Confusion Matrix

Figure 8. VGG19 classification performance



a) VGG19 train validation accuracy Curve

b) VGG19 train validation loss Curve

Figure 9. VGG19 train/validation accuracy and loss curve

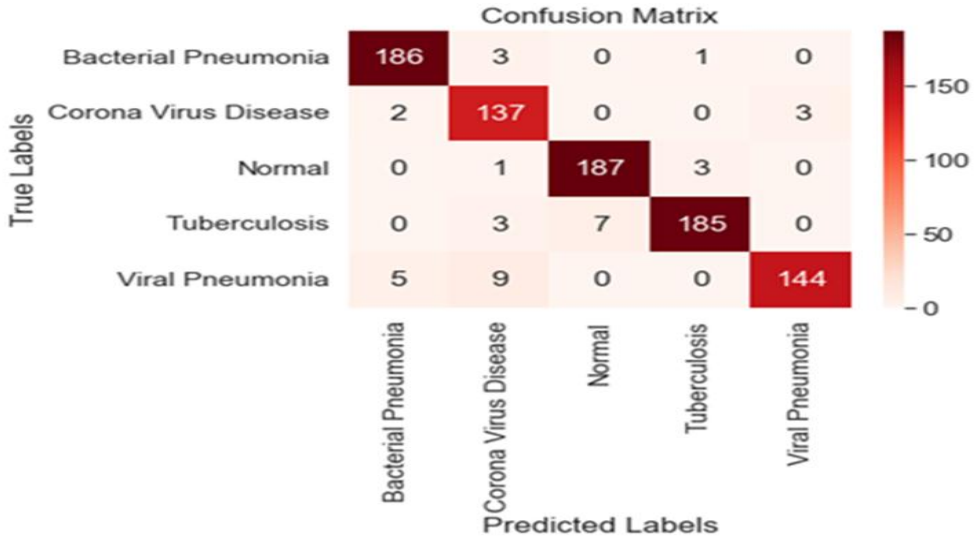
5.5 Proposed entropy optimized InceptionV3 single backbone result

The entropy optimized InceptionV3 single backbone experiment classification performance result is presented in Figure 10. (a) The classification result of the entropy optimized InceptionV3 single backbone accuracy is 96%. As shown from the result the entropy optimized InceptionV3 has shown high classification result giving high confidence in lung disease classification compared to the baseline InceptionV3. (b) the confusion matrix result where there are a total of 37 confusions out of 876 test samples which is high performance. Therefore, entropy optimized InceptionV3 model has the ability to classify lung disease with high accuracy and low confusion compared to other baselines. The proposed cross entropy optimized Inception V3 model has very few false positives compared to other baselines.

Classification Report:

	precision	recall	f1-score	support
Bacterial Pneumonia	0.96	0.98	0.97	190
Corona Virus Disease	0.90	0.96	0.93	142
Normal	0.96	0.98	0.97	191
Tuberculosis	0.98	0.95	0.96	195
Viral Pneumonia	0.98	0.91	0.94	158
accuracy			0.96	876
macro avg	0.96	0.96	0.96	876
weighted avg	0.96	0.96	0.96	876

a) Cross Entropy loss Optimized Inception Backbone Report



b) Cross Entropy Optimized Inception Backbone Confusion Matrix

Figure 10. Cross entropy optimized InceptionV3 backbone classification performance

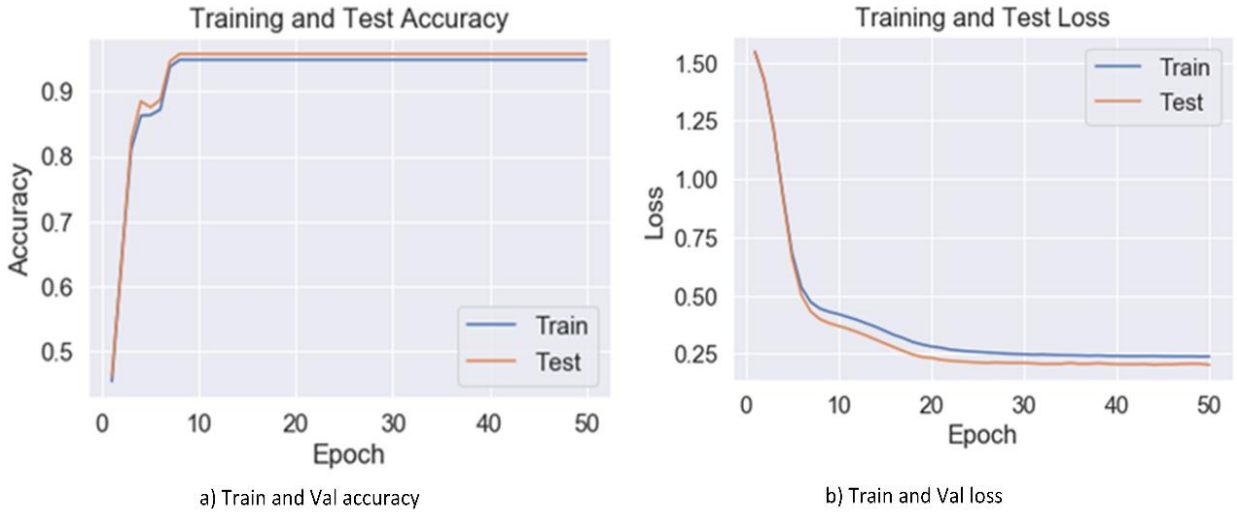


Figure 11. Cross Entropy Optimized Inception V3 Backbone train/validation accuracy and loss curve.

The train and validation accuracy curve shown in Figure 11 is the Cross Entropy Optimized Inception V3 where there is a consistent increase and the loss curve a consistent decrease.

5.6 Proposed entropy optimized InceptionV3 dual backbone result

The entropy optimized InceptionV3 dual backbone experiment classification performance result is presented in Figure 12. (a) the classification result of the entropy optimized InceptionV3 dual backbone accuracy is 97%. As shown from the result the entropy optimized dual InceptionV3 backbone has shown high classification result compared to the single backbone giving high confidence in lung disease classification. The confusion matrix result where there are a total of 40 confusions out of 876 test samples is also an indication of the high performance. Therefore, entropy optimized dual InceptionV3 model has the ability to classify lung disease with high accuracy and low confusion compared to other baselines.

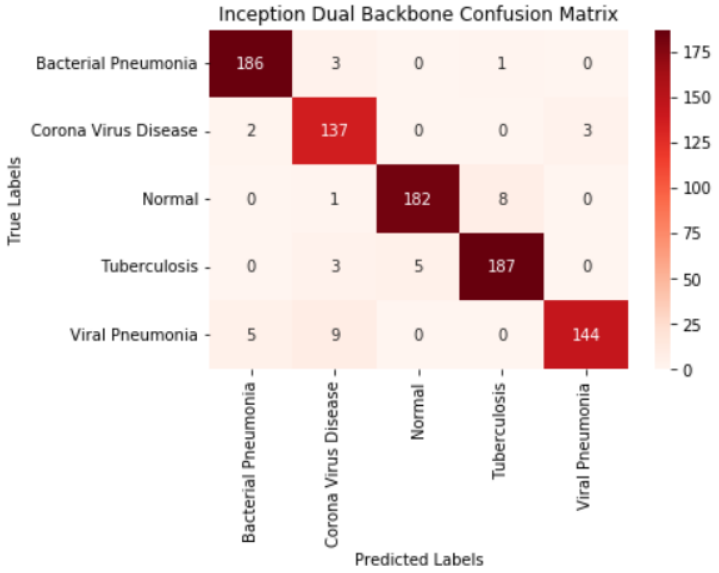


Figure 12. Cross entropy optimized InceptionV3 dual backbone classification performance

The train and validation accuracy curve shown in Figure 13 is the Cross entropy optimized InceptionV3 where there is a consistent increase and the loss curve a consistent decrease. To see the effect of increasing the epochs we trained for 50 epochs and 100 epochs and the curves shows no significant change. After epoch 50 the classification performance almost remains constant. Therefore, training the proposed method 50 epoch is sufficient to the maximum epoch the model needs to be trained.

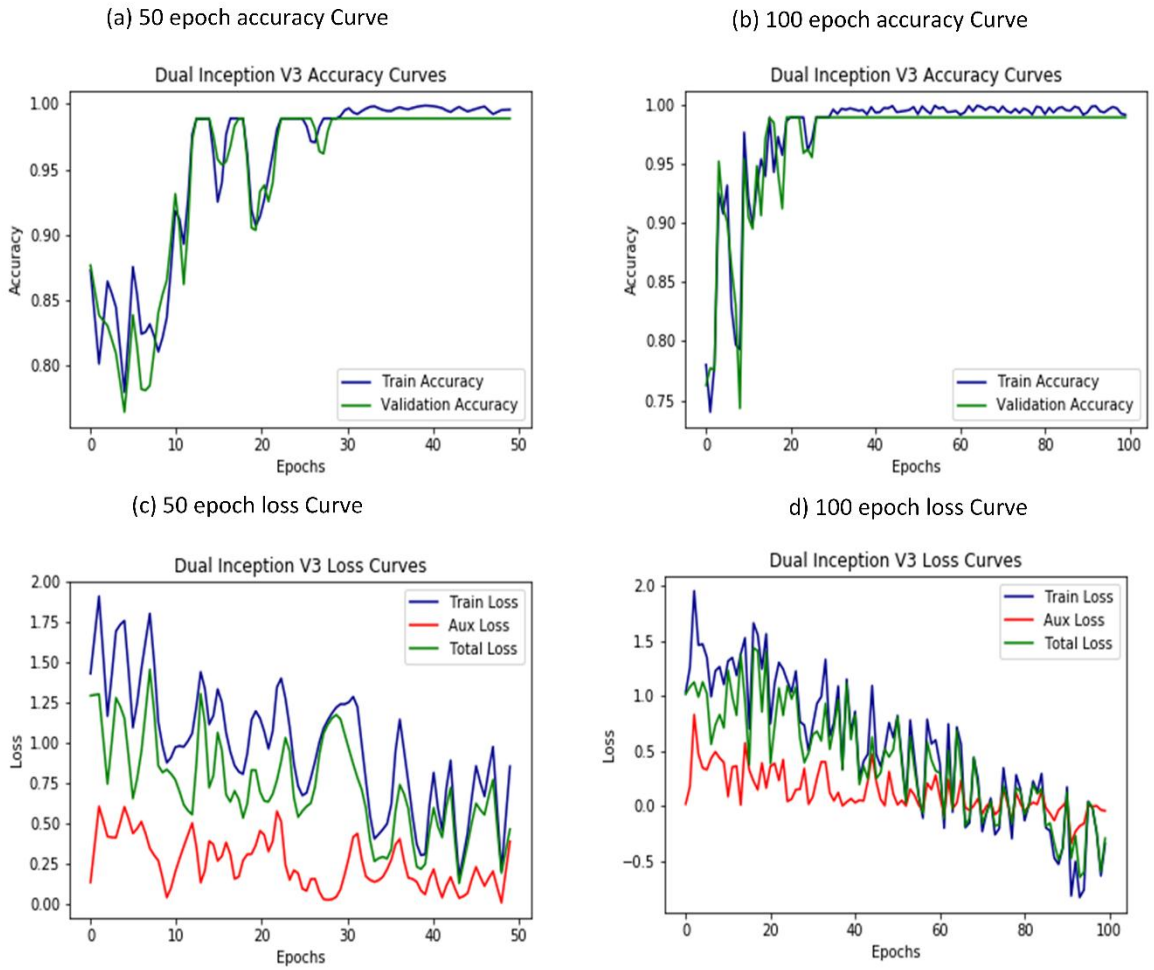


Figure 13. Cross entropy optimized Inception V3 dual backbone train/validation accuracy and loss curve

Table 4. Aggregate Summary of the results (in %)

Model	Precision	Recall	F1-score	Accuracy
VGG19	86	78	78	81
ResNet50	87	86	86	88
Inception V3 baseline	94	94	94	94
Optimized Inception V3(Ours)	96	96	96	96
Dual backbone InceptionV3(Ours)	98	97	97	97

As shown on Table 4, the aggregate summary of the results displayed both cross entropy optimized Inception V3 methods have outperformed other baselines models by a minimum of 1% accuracy for example in Inception V3 baseline, by a maximum of 16% in VGG19 baseline model. Therefore. We can conclude that the proposed cross entropy optimized Inception V3 strong enough to identify lung disease.

CHAPTER SIX

CONCLUSION AND RECOMMENDATION

6.1 Conclusion

The utilization of auxiliary cross entropy within the InceptionV3 network has significantly bolstered the accuracy and efficiency of detecting and classifying lung diseases from medical imaging data. Through the integration of this novel approach, the network has been empowered to discern crucial features essential for distinguishing between diverse lung disease patterns. The results from multiple model experiments reveal that the cross entropy optimized InceptionV3 models have consistently outperformed baseline models by a substantial margin, demonstrating their superior efficacy in identifying lung diseases accurately.

6.2 Recommendation:

The success of the cross entropy optimized InceptionV3 model in enhancing lung disease detection and classification underscores the potential for its widespread adoption in medical image analysis. This innovative approach showcases a robust system capable of precise identification and classification of various lung diseases from x-ray images. With its demonstrated efficacy in improving the performance of the InceptionV3 network, this method paves the way for more accurate and reliable automated systems in pulmonary healthcare. It is recommended to further explore and validate the proposed approach in real-world clinical settings to ascertain its full potential in revolutionizing early diagnosis and treatment planning for lung diseases.

References

Abbas, A.; Abdelsamea, M.M. Learning Transformations for Automated Classification of Manifestation of Tuberculosis using Convolutional Neural Network. In Proceedings of the 2018 13th International Conference on Computer Engineering andSystems (ICCES), Cairo, Egypt, 18–19 December 2018; IEEE: New York, NY, USA, 2018; pp. 122–126.

Adler D, Richards WF. 1953 Consolidation in primary pulmonary tuberculosis. Thorax. Sep;8(3):223.

Agnus James, Harisree P, Adarsh S, Vishnu S, 2021, Computer-Aided Detection of Tuberculosis using Convolutional Neural Networks, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) NCREIS – 2021 (Volume 09 – Issue 13)

Al-Ajlan, A.; Allali, A.E. CNN—MGP: Convolutional Neural Networks for Metagenomics Gene Prediction.

Am. J. Respir. Crit. Care Med. ***2000****, 161, 1376–1395.*

American Thoracic Society. Diagnostic Standards and Classification of Tuberculosis in Adults and Children.

Andika, L.A.; Pratiwi, H.; Sulistijowati Handajani, S. Convolutional neural network modeling for classification of pulmonary tuberculosis disease. J. Phys. Conf. Ser. ***2020****, 1490.*

Antony, B.; Nizar Banu, P.K. Lung tuberculosis detection using x-ray images. Int. J. Appl. Eng. Res. ***2017****, 12, 15196–15201.*

Ardila, D.; Kiraly, A.P.; Bharadwaj, S.; Choi, B.; Reicher, J.J.; Peng, L.; Tse, D.; Etemadi, M.; Ye, W.; Corrado, G.; et al. End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. Nat. Med. ***2019****, 25, 954–961.*

Ayan, E.; Ünver, H.M. Diagnosis of Pneumonia from Chest X-Ray Images using Deep Learning. Sci. Meet. Electr.-Electron. Biomed. Eng. Comput. Sci. ***2019****, 1–5.*

Boureau, Y. L., et al. "Icml2010B. Pdf. Icml." (2010): 111-118.

Bousquet, J. Global Surveillance, Prevention and Control of Chronic Respiratory Diseases; World Health Organization: Geneva, Switzerland, 2007; pp. 12–36.

Bouvrie, Jake. "Notes on convolutional neural networks." (2006).

Cao, X.; Wipf, D.; Wen, F.; Duan, G.; Sun, J. A practical transfer learning algorithm for face verification. In Proceedings of the IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013; pp. 3208–3215.

Cao, Y.; Liu, C.; Liu, B.; Brunette, M.J.; Zhang, N.; Sun, T.; Zhang, P.; Peinado, J.; Garavito, E.S.; Garcia, L.L.; et al. Improving Tuberculosis Diagnostics Using Deep Learning and Mobile Health Technologies among Resource-Poor and Marginalized Communities. In Proceedings of the 2016 IEEE 1st International Conference on Connected Health: Applications, Systems and Engineering Technologies, CHASE, Washington, DC, USA, 27–29 June 2016 ; pp. 274–281.

CDC: Centers for Disease Control and Prevention (www.cdc.gov)

*Chauhan, A.; Chauhan, D.; Rout, C. Role of gist and PHOG features in computer-aided diagnosis of tuberculosis without segmentation. PLoS ONE **2014**, 9, e112980.*

*Chouhan, V.; Singh, S.K.; Khamparia, A.; Gupta, D.; Albuquerque, V.H.C.D. A Novel Transfer Learning Based Approach for Pneumonia Detection in Chest X-ray Images. Appl. Sci. **2020**, 10, 559.*

Chung MJ, Goo JM, Im JG. 2004. Pulmonary tuberculosis in patients with idiopathic pulmonary fibrosis. European journal of radiology. 1; 52(2):175-9.

*Coudray, N.; Ocampo, P.S.; Sakellaropoulos, T.; Narula, N.; Snuderl, M.; Fenyö, D.; Moreira, A.L.; Razavian, N.; Tsirigos, A. Classification and mutation prediction from non–small cell lung cancer histopathology images using deep learning. Nat. Med. **2018**, 24, 1559–1567.*

Chung MJ, Goo JM, Im JG. 2004. Pulmonary tuberculosis in patients with idiopathic pulmonary fibrosis. European journal of radiology. 1; 52(2):175-9.

*Development and Validation of a Deep Learning—based Automatic Detection Algorithm for Active Pulmonary Tuberculosis on Chest Radiographs. Clin. Infect. Dis. **2019**, 69, 739–747.*

Devnath L, Luo S, Summons P and Wang D 2018 Tuberculosis (Tb) Classification in Chest Radiographs using Deep Convolutional Neural networks. 6 68–74

*Dietterich, T.G. Ensemble Methods in Machine Learning. Int. Workshop Mult. Classif. Syst. **2000**, 1–15._1.*

Divina, F.; Gilson, A.; Goméz-Vela, F.; Torres, M.G.; Torres, J.F. Stacking Ensemble Learning for Short-Term Electricity Consumption Forecasting. Energies ***2018****, 11, 949.*

Domingos, P. A Few Useful Things to Know About Machine Learning. Commun. ACM ***2012****, 55, 78–87.*

Forum of International Respiratory Societies. The Global Impact of Respiratory Disease, 2nd ed.; European Respiratory Society, Sheffield, UK, 2017; pp. 5–42.Front. Med. ***2019****, 14, 450–469.*

Gao, X.W.; James-reynolds, C.; Currie, E. Analysis of tuberculosis severity levels from CT pulmonary images based on enhanced residual deep learning architecture. Neurocomputing ***2019****, 392, 233–244.*

Gao, Z., Xu, Y., Sun, C., Wang, X., Guo, Y., Qiu, S., & Ma, K. (2021). A systematic review of asymptomatic infections with COVID-19. Journal of Microbiology, Immunology and Infection, 54(1), 12-16.

Ghorakavi and Srivatsav R 2019 TBNet:Pulmonary Tuberculosis Diagnosing System using Deep Neural Networks arXiv Prepr. arXiv1902.08897.

Gil V, Soler JJ, Cordero PJ. 1994 .Pleural Thickening in Patients With Pleural Tuberculosis. Chest. 1;105(4):1296.

Gordienko, Y.; Gang, P.; Hui, J.; Zeng, W.; Kochura, Y.; Alienin, O.; Rokovyi, O.; Stirenko, S. Deep Learning with Lung Segmentation and Bone Shadow Exclusion Techniques for Chest X-Ray Analysis of Lung Cancer. Adv. Intell. Syst. Comput. ***2019****, 638–647.*

Gozes, O.; Frid, M.; Greenspan, H.; Patrick, D. Rapid AI Development Cycle for the Coronavirus (COVID-19) Pandemic: Initial Results for Automated Detection & Patient Monitoring using Deep Learning CT Image Analysis Article. arXiv ***2020****, arXiv:2003.05037.*

Gurcan, M.N.; Boucheron, L.E.; Can, A.; Madabhushi, A.; Rajpoot, N.M.; Yener, B. Histopathological Image Analysis: A Review. IEEE Rev. Biomed. Eng. ***2009****, 2, 147–171.*

H. Boussaid and I. Kokkinos, Jun. 2014 ‘‘Fast and exact: ADMM-based discriminative shape segmentation with loopy part models,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR),, pp. 4058–4065.

Heo S J, Kim Y, Yun S, Lim S S, Kim J, Nam C M, Park E C, Jung I and Yoon J H 2019 Deep learning algorithms with demographic information help to detect tuberculosis in chest radiographs in annual workers' health examination data Int. J. Environ. Res. Public Health 16

Heo, S.J.; Kim, Y.; Yun, S.; Lim, S.S.; Kim, J.; Nam, C.M.; Park, E.C.; Jung, I.; Yoon, J.H. Deep Learning Algorithms with Demographic Information Help to Detect Tuberculosis in Chest Radiographs in Annual Workers' Health Examination Data. Int. J. Environ. Res. Public Health ***2019****, 16, 250.*

Herman, G.T. Fundamentals of Computerized Tomography; Springer: London, UK, 2009; Volume 224.

Hinton, G. E. "Improving neural networks by preventing co-adaptation of feature detectors." arXiv preprint arXiv:1207.0580 (2012).

Hinton, G.E.; Osindero, S. A fast learning algorithm for deep belief nets. Neural Comput. ***2006****, 18, 1527–1554.*

Hooda R, Mittal A and Sofat S 2019 Automated TB classification using ensemble of deep architectures Multimed. Tools Appl. 78 31515–32

Hu, Z.; Tang, J.; Wang, Z.; Zhang, K.; Zhang, L.; Sun, Q. Deep learning for image-based cancer detection and diagnosis-A survey. Pattern Recognit. ***2018****, 83, 134–149.*

Hwang S, Kim H-E, Jeong J and Kim H-J 2016 A novel approach for tuberculosis screening based on deep convolutional neural networks Medical Imaging 2016: Computer-Aided Diagnosis vol 9785 p 97852W

Hwang, E.J.; Park, S.; Jin, K.N.; Kim, J.I.; Choi, S.Y.; Lee, J.H.; Goo, J.M.; Aum, J.; Yim, J.J.; Park, C.M.

Hwang, S.; Kim, H.E.; Jeong, J.; Kim, H.J. A Novel Approach for Tuberculosis Screening Based on Deep Convolutional Neural Networks. Med. Imaging ***2016****, 9785, 1–8.*

Interdiscip. Sci. Comput. Life Sci. ***2019****, 11, 628–635.*

Islam, M.T.; Aowal, M.A.; Minhaz, A.T.; Ashraf, K. Abnormality Detection and Localization in Chest X-Rays using Deep Convolutional Neural Networks. arXiv ***2017****, arXiv:1705.09850v3.*

Jaeger, S.; Karargyris, A.; Candemir, S.; Folio, L.; Siegelman, J.; Callaghan, F.; Xue, Z.; Palaniappan, K.; Singh, R.K.; Antani, S.; et al. Automatic tuberculosis screening using chest radiographs. IEEE Trans. Med. Imaging ***2014****, 33, 233–245.*

K. He, X. Zhang, S. Ren and J. Sun, "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 9, pp. 1904-1916, 1 Sept. 2015,

K. Munadi, K. Muchtar, N. Maulina and B. Pradhan, "Image Enhancement for Tuberculosis Detection Using Deep Learning," in IEEE Access, vol. 8, pp. 217897-217907, 2020, doi: 10.1109/ACCESS.2020.3041867

Kabari, L.G.; Onwuka, U. Comparison of Bagging and Voting Ensemble Machine Learning Algorithm as a Classifier. Int. J. Adv. Res. Comput. Sci. Softw. Eng. ***2019****, 9, 1–6.*

Kant, S.; Srivastava, M.M. Towards Automated Tuberculosis detection using Deep Learning. In Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence (SSCI), Bengaluru, India, 18–21 November 2018; pp. 1250–1253.

Karnkawinpong, T.; Limpiyakorn, Y. Classification of pulmonary tuberculosis lesion with convolutional neural networks. J. Phys. Conf. Ser. ***2018****, 1195.*

Ker, J.; Wang, L. Deep Learning Applications in Medical Image Analysis. IEEE Access ***2018****, 6, 9375–9389.*

Kieu, S.T.H.; Hijazi, M.H.A.; Bade, A.; Yaakob, R.; Jeffree, S. Ensemble deep learning for tuberculosis detection using chest X-Ray and canny edge detected images. IAES Int. J. Artif. Intell. ***2019****, 8, 429–435.*

Kieu, T.; Ho, K.; Gwak, J.; Prakash, O. Utilizing Pretrained Deep Learning Models for Automated Pulmonary Tuberculosis Detection Using Chest Radiography. Intell. Inf. Database Syst. ***2019****, 4, 395–403.*

Krizhevsky, A.; Sutskeve, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. Adv. Neural Inf. Process. Syst. ***2012****.*

Kwan-Hoong, N.; Madan M, R. X ray imaging goes digital. Br. Med J. ***2006****, 333, 765–766.*

Lakhani P and Sundaram B 2017 Deep Learning at Chest Radiography : Automated Classification of Pulmonary Tuberculosis by Using Convolutional neural networks Radiology 284 574–82

*Lakhani, P.; Sundaram, B. Deep Learning at Chest Radiography: Automated Classification of Pulmonary Tuberculosis by Using Convolutional Neural Networks. Radiology **2017**, 284, 574–582.*

Lanbouri, Z.; Achchab, S. A hybrid Deep belief network approach for Financial distress prediction. In Proceedings of the 2015 10th International Conference on Intelligent Systems: Theories and Applications (SITA), Rabat, Morocco, 20–21 October 2015; pp. 1–6.

Laurent, César, et al. "Batch normalized recurrent neural networks." 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2016.

*Lincoln, W.P.; Skrzypekt, J. Synergy of Clustering Multiple Back Propagation Networks. Adv. Neural Inf. Process. Syst. **1990**, 2, 650–659.*

Liu C, Cao Y, Alcantara M, Liu B, Brunette M, Peinado J and Curioso W 2018 TX-CNN: Detecting tuberculosis in chest X-ray images using convolutional neural network Proc. - Int. Conf. Image Process. ICIP 2314–8

Liu Y, Wu Y-H, Ban Y, Wang H and Ming-Ming Cheng 2020 Rethinking Computer-aided Tuberculosis Diagnosis IEEE/CVF Conference on Computer Vision and Pattern Recognition pp 2646–55

Liu, J.; Liu, Y.; Wang, C.; Li, A.; Meng, B. An Original Neural Network for Pulmonary Tuberculosis Diagnosis in Radiographs. In Lecture Notes in Computer Science, Proceedings of the International Conference on Artificial Neural Networks, Rhodes, Greece, 4–7 October 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 158–166._16.

*Lopes, U.K.; Valiati, J.F. Pre-trained convolutional neural networks as feature extractors for tuberculosis detection. Comput. Biol. Med. **2017**, 89, 135–143.*

López, Y.P.; Filho, C.F.F.C.; Aguilera, L.M.R.; Costa, M.G.F. Automatic classification of light field smear microscopy patches using Convolutional Neural Networks for identifying Mycobacterium Tuberculosis. In Proceedings of the 2017 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON), Pucon, Chile, 18–20 October 2017 .

Ma, J.; Song, Y.; Tian, X.; Hua, Y.; Zhang, R.; Wu, J. Survey on deep learning for pulmonary medical imaging.

Melendez, J.; Sánchez, C.I.; Philipsen, R.H.; Maduskar, P.; Dawson, R.; Theron, G.; Dheda, K.; Van Ginneken, B. An automated tuberculosis screening strategy combining X-ray-based computer-aided detection and clinical information. Sci. Rep. ***2016****, 6, 1–8*

Mikołajczyk, A.; Grochowski, M. Data augmentation for improving deep learning in image classification problem. In Proceedings of the 2018 International Interdisciplinary PhD Workshop, Swinoujscie, Poland, 9–12 May 2018; pp. 117–122.

Mishin V, Nazarova NV, Kononen AS, Miakishev TV, Sadovski AI. 2006. Infiltrative pulmonary tuberculosis: course and efficiency of treatment. Problemy tuberkuleza i boleznei Kant S, Kushwaha R, Verma SK. 2007;8. Bilateral nodular pulmonary tuberculosis simulating metastatic lung cancer. The Internet Journal of Pulmonary Medicine.

Mithra, K.S.; Emmanuel, W.R.S. Automated identification of mycobacterium bacillus from sputum images for tuberculosis diagnosis. Signal Image Video Process. ***2019****.*

Murali Krishna Puttagunta and S. Ravi. 2021 Detection of Tuberculosis based on Deep Learning based methods. Journal of Physics J. Phys.: Conf. Ser. 1767 012004

Murphy, K.; Habib, S.S.; Zaidi, S.M.A.; Khowaja, S.; Khan, A.; Melendez, J.; Scholten, E.T.; Amad, F.; Schalekamp, S.; Verhagen, M.; et al. Computer aided detection of tuberculosis on chest radiographs: An evaluation of the CAD4TB v6 system. Sci. Rep. ***2019****, 10, 1–11.*

Nguyen Q H, Nguyen B P, Dao S D, Unnikrishnan B, Dhingra R, Ravichandran S R, Satpathy S, Raja P N and Chua M C H 2019 Deep Learning Models for Tuberculosis Detection from Chest X-ray Images 2019 26th International Conference on Telecommunications (ICT)(IEEE) pp 381–5

Nguyen, Q.H.; Nguyen, B.P.; Dao, S.D.; Unnikrishnan, B.; Dhingra, R.; Ravichandran, S.R.; Satpathy, S.; Raja, P.N.; Chua, M.C.H. Deep Learning Models for Tuberculosis Detection from Chest X-ray Images. In Proceedings of the 2019 26th International Conference on Telecommunications (ICT), Hanoi, Vietnam, 8–10 April 2019; pp. 381–385.

Nogueira, K.; Penatti, O.A.; dos Santos, J.A. Towards better exploiting convolutional neural networks for remote sensing scene classification. Pattern Recognit. ***2017****, 61, 539–556.*

Nwankpa, Chigozie, et al. "Activation functions: Comparison of trends in practice and research for deep learning." arXiv preprint arXiv:1811.03378 (2018).

O'Mahony, N.; Campbell, S.; Carvalho, A.; Harapanahalli, S.; Hernandez, G.V.; Krpalkova, L.; Riordan, D.; Walsh, J. Deep Learning vs . Traditional Computer Vision. Adv. Intell. Syst. Comput. ***2020****, 128–144.*

O'Shea, K.; Nash, R. An Introduction to Convolutional Neural Networks. arXiv ***2015****, arXiv:1511.08458v2.*

Oomman, R.; Kalmady, K.S.; Rajan, J.; Sabu, M.K. Automatic detection of tuberculosis bacilli from microscopic sputum smear images using deep learning methods. Integr. Med. Res. ***2018****, 38, 691–699.*

Pan, S.J.; Yang, Q. A Survey on Transfer Learning. IEEE Trans. Knowl. Data Eng. ***2010****, 22, 1345–1359.*

Pande T, Cohen C, Pai M and Ahmad Khan F 2016 Computer-aided detection of pulmonary tuberculosis on digital chest radiographs: A systematic review Int. J. Tuberc. Lung Dis. 201226–30

Pasa, F.; Golkov, V.; Pfeiffer, F.; Cremers, D.; Pfeiffer, D. Efficient Deep Network Architectures for Fast Chest X-Ray Tuberculosis Screening and Visualization. Sci. Rep. ***2019****, 9, 2–10.*

Qin Z Z, Sander M S, Rai B, Titahong C N, Sudrungrot S, Laah S N, Adhikari L M, Carter E J, Puri L, Codlin A J and Creswell J 2019 Using artificial intelligence to read chest radiographs for tuberculosis detection : A multi-site evaluation of the diagnostic accuracy of three deep learning systems Sci. Rep. 9 1–1

Rahaman, M.M.; Li, C.; Yao, Y.; Kulwa, F.; Rahman, M.A.; Wang, Q.; Qi, S.; Kong, F.; Zhu, X.; Zhao, X. Identification of COVID-19 samples from chest X-Ray images using deep learning: A comparison of transfer learning approaches. J. X-Ray Sci. Technol. ***2020****, 28, 821–839.*

Rajaraman, S.; Candemir, S.; Xue, Z.; Alderson, P.O.; Kohli, M.; Abuya, J.; Thoma, G.R.; Antani, S.; Member, S. A novel stacked generalization of models for improved TB detection in chest radiographs. In Proceedings of the 2018 40th Annual International Conference the IEEE Engineering in Medicine and Biology Society (EMBC), Honolulu, HI, USA, 17–21 July 2018; pp. 718–721.

Rao, P.; Pereira, N.A.; Srinivasan, R. Convolutional neural networks for lung cancer screening in computed tomography (CT) scans. In Proceedings of the 2016 2nd International

Conference on Contemporary Computing and Informatics, IC3I 2016, Noida, India, 14–17 December 2016 ; pp. 489–493.

S. Hermann, ‘‘Evaluation of scan-line optimization for 3D medical image registration,’’ in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2014, pp. 3073–3080.

S. Jaeger et al., ‘‘Automatic tuberculosis screening using chest radiographs,’’ IEEE Trans. Med. Imag., vol. 33, no. 2, pp. 233–245, Feb. 2014.

S. Kant and M. M. Srivastava. 2018 "Towards Automated Tuberculosis detection using Deep Learning," IEEE Symposium Series on Computational Intelligence (SSCI), 2018, pp. 1250-1253,

S. Z. Y. Zaidi, M. U. Akram, A. Jameel and N. S. Alghamdi, "Lung Segmentation-Based Pulmonary Disease Classification Using Deep Neural Networks," in IEEE Access, vol. 9, pp. 125202-125214, 2021,

Salman, F.M.; Abu-naser, S.S.; Alajrami, E.; Abu-nasser, B.S.; Ashqar, B.A.M. COVID-19 Detection using Artificial Intelligence. Int. J. Acad. Eng. Res. ***2020****, 4, 18–25.*

Samuel, R.D.J.; Kanna, B.R. Tuberculosis (TB) detection system using deep neural networks. Neural Comput. Appl. ***2019****, 31, 1533–1545.*

Setio, A.A.A.; Traverso, A.; de Bel, T.; Berens, M.S.; van den Bogaard, C.; Cerello, P.; Chen, H.; Dou, Q.; Fantacci, M.E.; Geurts, B.; et al. Validation, comparison, and combination of algorithms for automatic detection of pulmonary nodules in computed tomography images: The LUNA16 challenge. Med. Image Anal. ***2017****, 42, 1–13.*

Shah, M.I.; Mishra, S.; Yadav, V.K.; Chauhan, A.; Sarkar, M.; Sharma, S.K.; Rout, C. Ziehl–Neelsen sputum smear microscopy image database: A resource to facilitate automated bacilli detection for tuberculosis diagnosis. J. Med. Imaging ***2017****, 4, 027503.*

Shen, D.; Wu, G.; Suk, H.I. Deep Learning in Medical Image Analysis. Annu. Rev. Biomed. Eng. ***2017****, 19, 221–248.*

Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. J. Big Data ***2019****,*

Sivaramakrishnan R, Antani S, Candemir S, Xue Z, Thoma G, Alderson P, Abuya J and Kohli M 2018 Comparing deep learning models for population screening using chest radiography SPIE 10575, Medical Imaging 2018: Computer-Aided Diagnosis, p 49

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2818-2826).

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017, February). Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the AAAI conference on artificial intelligence (Vol. 31, No. 1).

Song, Q.Z.; Zhao, L.; Luo, X.K.; Dou, X.C. Using Deep Learning for Classification of Lung Nodules on Computed Tomography Images. J. Healthc. Eng. ***2017****, 2017.*

Stirenko, S.; Kochura, Y.; Alienin, O. Chest X-Ray Analysis of Tuberculosis by Deep Learning with Segmentation and Augmentation. In Proceedings of the 2018 IEEE 38th International Conference on Electronics andNanotechnology (ELNANO), Kiev, Ukraine, 24–26 April 2018; pp. 422–428.

Tajbakhsh, N.; Shin, J.Y.; Gurudu, S.R.; Hurst, R.T.; Kendall, C.B.; Gotway, M.B.; Liang, J. Convolutional Neural Networks for Medical Image Analysis: Full Training or Fine Tuning? IEEE Trans. Med. Imaging ***2016****, 35, 1299–1312.*

U. Avni, H. Greenspan, E. Konen, M. Sharon, and J. Goldberger, ''X-ray categorization and retrieval on the organ and pathology level, using patchbased visual words,'' IEEE Trans. Med. Imag., vol. 30, no. 3, pp. 733–746, Mar. 2011.

Ul Abideen, Z.; Ghafoor, M.; Munir, K.; Saqib, M.; Ullah, A.; Zia, T.; Tariq, S.A.; Ahmed, G.; Zahra, A. Uncertainty assisted robust tuberculosis identification with bayesian convolutional neural networks. IEEE Access ***2020****, 8, 22812–22825.*

Vajda, S.; Karargyris, A.; Jaeger, S.; Santosh, K.C.; Candemir, S.; Xue, Z.; Antani, S.; Thoma, G. Feature Selection for Automatic Tuberculosis Screening in Frontal Chest Radiographs. J. Med Syst. ***2018****, 42.*

Vorster MJ, Allwood BW, Diacon AH, Koegelenberg CF.2015. Tuberculous pleural effusions: advances and controversies. Journal of thoracic disease.7 (6):981.

Vorster MJ, Allwood BW, Diacon AH, Koegelenberg CF.2015. Tuberculous pleural effusions: advances and controversies. Journal of thoracic disease.7 (6):981.

Wang, C.; Chen, D.; Hao, L.; Liu, X.; Zeng, Y.; Chen, J.; Zhang, G. Pulmonary Image Classification Based on Inception-v3 Transfer Learning Model. IEEE Access ***2019****, 7, 146533–146541.*

World Health Organization, 2018. Global tuberculosis report 2018. Geneva: World Health Organization.

Wang, Tao, et al. "End-to-end text recognition with convolutional neural networks." Proceedings of the 21st international conference on pattern recognition (ICPR2012). IEEE, 2012.

Webb, A. Introduction To Biomedical Imaging; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2003.

World Health Organisation. Global Health TB Report; World Health Organisation: Geneva, Switzerland, 2018;

World Health Organization, 2018. Global tuberculosis report 2018. Geneva: World Health Organization.

World Health Organization. Coronavirus Disease 2019 (COVID-19) Situation Report; Technical Report March; World Health Organization: Geneva, Switzerland, 2020.

Wu, C.; Luo, C.; Xiong, N.; Zhang, W.; Kim, T.H. A Greedy Deep Learning Method for Medical Disease Analysis. IEEE Access ***2018****, 6, 20021–20030.*

Xu, X., Guo, Q., Guo, J., & Yi, Z. (2018). DeepCXray: Automatically diagnosing diseases on chest x-rays using deep neural networks. IEEE Access, 6, 66972–66982.

Yahiaoui, A.; Er, O.; Yumusak, N. A new method of automatic recognition for tuberculosis disease diagnosis using support vector machines. Biomed. Res. ***2017****, 28, 4208–4212.*