1)Acknowledgment

2)Declarartion

3)Abstract

4)List of contents

5)Abbrevation

6)Introduction

7)Review of Literarture

8)Methods

9)Results

10)Conclusion and discussion

11)References

12)Appendix

**Acknowledgment**

As per today, this project has reached it's completion, denoting the conclusion of this journey. There are certain people who have been my backbone through this journey and I extend my hearty gratefulness towards their support, you have my deepest thanks. First and foremost, I thank our valuable guide Mr. R.L Meshram for their persistent and ceaseless motivation, guidance and betterment solutions which resulted in the successful development of this project. Your constant support and opportunity for our growth is highly appreciated. Warm and extended thank you for their unswerving moral support and inspiration throughout the course of this project to our respected Head of department, Dr. A.R Mahajan. Huge thanks to our epitome of inspiration, Dr. Krushnansh Mogre sir who kept our spirits up along the course. Thank you so much for helping us grow a developing and learning environment among the institute. I would elongate these words of thankfulness to the staff members, faculty and working people of our college for their significant contributions towards the completion of this project. As for the conclusive part, hearty thanks to my friends and family for their understanding and assistance which provided the energy and aid to successfully reach our goal.

**Declaration**

It is hereby certified to you that "Project name" has reached it's fulfilment as an authentic achievement in partial completion of Diploma in Information Technology At Government Polytechnic, Nagpur. Under the valuable supervision of Prof R.L Meshram, this project thesis is submitted to Department of Information Technology during the academic session 2024-25. The project focuses over betterment of Deep Learning models and our significant efforts to innovate a classification model surpassing accuracy levels of existing models by combining two efficient models. Our foremost aim and objective is to contribute in the betterment of Deep Learning : CNN and Classification Models domain, successfully achieved through the conclusion of this project.

**Abstract:**

In recent years, deep convolutional neural networks (DCNNs) have significantly advanced image classification, with the understanding that high-performing base learners contribute more effectively to ensemble models. Building upon prior research, particularly the study titled "Weighted Ensemble Model for Image Classification", this paper explores how models with higher accuracy can boost the overall performance of an ensemble. We train multiple state-of-the-art lightweight architectures, including ResNet18, DenseNet121, ShuffleNetV2, MobileNetV2, and EfficientNetB0, using a bagging ensemble technique to enhance classification accuracy. Our approach incorporates extensive data preprocessing and augmentation on an X-ray image dataset, resulting in a notable improvement in three-class accuracy compared to previous models in the literature. The results demonstrate that effective data handling not only enhances individual model performance but also amplifies the efficiency of ensemble methods. This research contributes valuable insights into optimizing ensemble techniques for image classification tasks and provides a foundation for future exploration in this domain.

**Revised Abstract for thesis:**

In recent years, deep convolutional neural networks (DCNNs) have revolutionized the field of image classification. A cornerstone of effective ensemble models is the incorporation of high-performance base learners. Building upon previous research, particularly the study titled "Weighted Ensemble Model for Image Classification," this paper delves deeper into the relationship between base learner accuracy and ensemble performance.

By training multiple state-of-the-art lightweight architectures, including ResNet18, DenseNet121, ShuffleNetV2, MobileNetV2, and EfficientNetB0, using a bagging ensemble technique, we aim to significantly enhance classification accuracy. Our approach involves meticulous data preprocessing and augmentation, tailored to an X-ray image dataset, leading to a substantial improvement in three-class accuracy compared to existing models in the literature.

The results of our experiments demonstrate that effective data handling is not only crucial for improving individual model performance but also plays a pivotal role in amplifying the efficiency of ensemble methods. This research contributes valuable insights into optimizing ensemble techniques for image classification tasks and provides a solid foundation for future explorations in this domain.

### 1. Introduction

This project focuses on enhancing the classification of lung diseases, specifically Pneumonia, No-findings and Covid-19, using ensemble deep learning models on X-ray images. By employing the bagging ensemble technique.

In ensemble learning, we use more than one model, generally called baseline models and the output of the all models are combined together to produce a single combined output. That is, we are combining the predictions/decisions of various models to produce the final prediction or decision. The bootstrapping and aggregation of the constituent base line models are done in various ways, which result in various different ensemble strategies.

Bagging is one of the primary strategies for building ensemble-based algorithms which is also known as bootstrap aggregating and is used to improve the performance of an ensemble classifier. Bagging has two main sub-processes: bootstrapping and aggregation. In bootstrapping, we divide the original dataset into sub sets (also called bagging samples), and each base learner is trained on one of these subsets or bagging samples. These bagging samples may or may not be overlapped. Now the base learners create an independent observations of same size and aggregation process combines these observations (commonly by voting for classification) to create a single observation which is better than the observation of a single model.

We worked on this project by taking reference of existing research work and simply increases accuracy for dataset, mostly by using light-weighted base learners and bagging ensemble technique.The research aims to demonstrate the effectiveness of these models in a clinical setting, ultimately contributing to automated diagnostic systems that can aid healthcare professionals in decision-making.

#### Problem Statement:

In medical field x-ray plays a very crucial role for identifying any type of disease. But it requires, lot of time, specialist availaibility for human inspection. The title “*Ensemble model for X-ray image classification”,* States that we are developing CNN based ensemble model for classification of different classes of X-ray images. Also, the rise in respiratory diseases, particularly lung-related ailments, poses a significant challenge to global healthcare systems. Accurate and timely diagnosis through imaging techniques such as X-rays is critical for effective treatment. However, manual interpretation of X-ray images can be time-consuming and subject to human error. This research addresses the problem of automating the detection and classification of lung diseases in X-ray images using deep learning techniques. By developing a robust model that accurately identifies various lung diseases, this study aims to enhance diagnostic efficiency and support healthcare professionals.

#### Research Objectives:

The specific goals of this project include:

1. Increasing Accuracy: Surpassing existing accuracy is the main goal of our project. We worked on different parameters, data augmentation, model selection, etc. to achieve higher accuracy.
2. Model Development: To develop an ensemble deep learning model that integrates multiple architectures (such as MobileNetV2, ShuffleNetV2, EfficientNet-B0, etc.) for enhanced classification performance on chest X-ray images.
3. Performance Evaluation: To evaluate the model’s performance using confusion matrices, classification reports, and metrics such as accuracy, precision, recall, and F1 score.
4. Comparison with Traditional Methods: To compare the performance of the developed model against traditional image analysis techniques, highlighting the advantages of automated approaches.
5. Real-world Application: To assess the practical implications of the model in clinical settings, aiming to reduce diagnostic errors and improve patient outcomes.

This all are some important objectives of our project. By following proper sequence of developing ensemble model for classifying labels of difficult datasets like X-rays can be achieved easily. Also, above mention objectives are achieved by us.

#### Motivation:

This research is crucial due to the increasing prevalence of lung diseases globally, exacerbated by factors such as pollution and smoking. Early detection is vital for effective treatment and better patient prognosis. By utilizing advanced deep learning techniques, this study not only aims to improve diagnostic accuracy but also to alleviate the workload of radiologists, allowing them to focus on more complex cases. The potential impact of this research extends to enhancing public health outcomes and resource allocation within healthcare systems.

Training image classification models on simple datasets, such as vegetables, is relatively straightforward. However, the task becomes significantly more challenging when dealing with complex datasets like X-ray images. Achieving high accuracy in classifying various lung diseases from X-ray images not only requires advanced techniques and robust models but also emphasizes the importance of precision in medical diagnostics.

This challenge drives the motivation behind this research, as improving accuracy in X-ray classification and increasing robustness can lead to better patient outcomes and enhance the efficiency of healthcare systems.

#### Research Questions:

The research aims to answer the following questions:

1. How effective are ensemble deep learning models compared to single-model architectures in classifying lung diseases from X-ray images?
2. What are the key factors influencing the performance of deep learning models in medical image classification?
3. What is the role of hyperparameter tuning in improving the accuracy of ensemble models for X-ray classification?
4. How does the size and quality of the X-ray dataset impact the performance of ensemble deep learning models?
5. Can the developed model achieve a level of accuracy that meets or exceeds that of traditional diagnostic methods?
6. What implications does the deployment of an automated X-ray classification system have on clinical workflows and patient outcomes?

These questions not only align with our research objectives but also deepen the investigation into key aspects of using deep learning in medical imaging.

#### Literature Review:

Existing research has explored various approaches to automated X-ray image classification. Numerous studies have demonstrated the effectiveness of convolutional neural networks (CNNs) in image analysis, with applications in detecting conditions such as pneumonia and Covid-19. However, many of these approaches rely on single-model architectures, which may limit accuracy. Recent advancements in ensemble learning have shown promise in improving classification performance by leveraging the strengths of multiple models.

Ensemble Learning: Ensemble learning combines multiple models, leveraging their strengths to improve accuracy. DCNN-based heterogeneous ensembles use diverse deep learning architectures and assign weights to models based on their individual performance. This approach enhances generalization and overall prediction accuracy.

**Existing work:**

Previous research has explored DCNN-based heterogeneous ensembles for X-ray image classification, demonstrating improved performance through model combination and weighted contributions.

This project is building upon prior research, particularly the study titled, *“*[*Weighted ensemble model for image classification | International Journal of Information Technology (springer.com)*](https://link.springer.com/article/10.1007/s41870-022-01149-8)*”*, this paper explores the effectiveness of a weighted ensemble method to improve classification accuracy, particularly for medical image datasets like X-rays. It combines multiple CNN-based classifiers and assigns higher weights to those performing better. This approach significantly enhances the final model's accuracy compared to using individual classifiers alone. The authors demonstrate the ensemble method's superiority in handling complex classification tasks, highlighting its potential in medical image analysis.

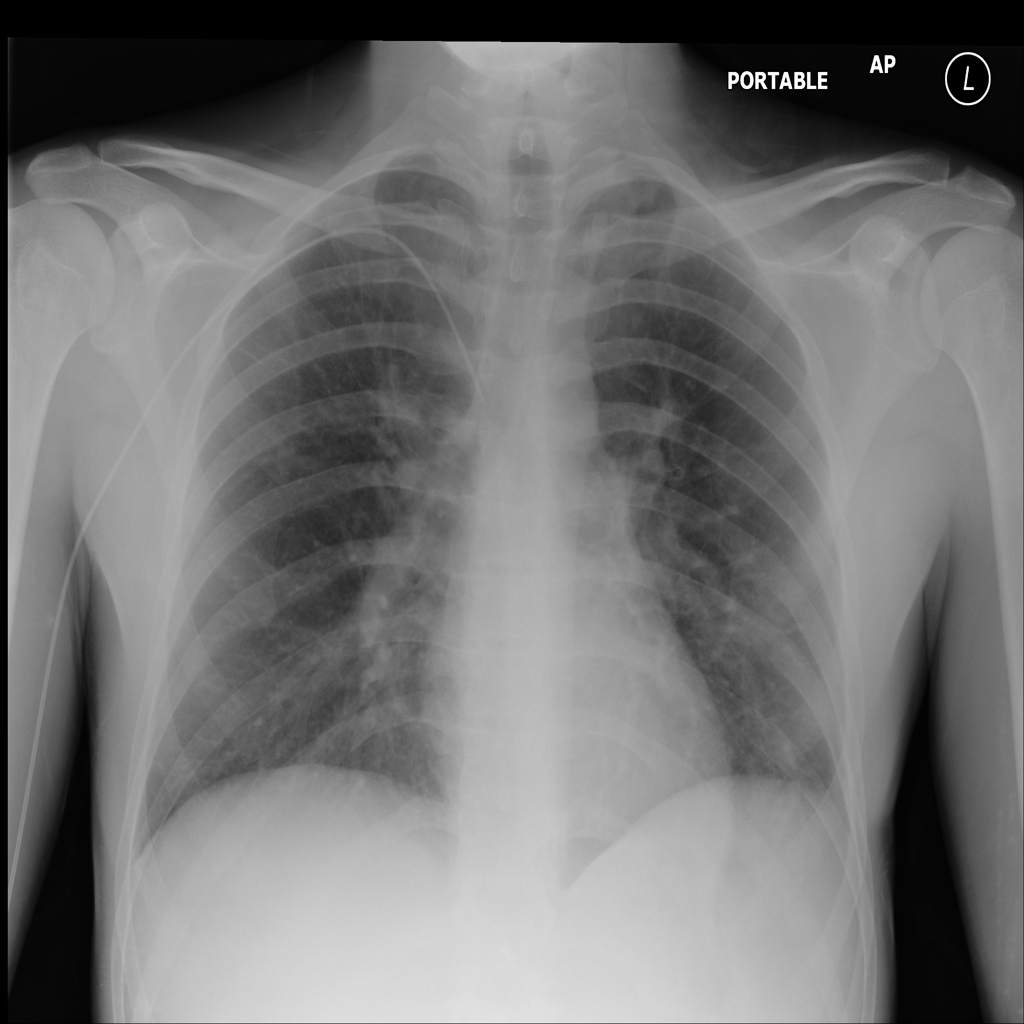
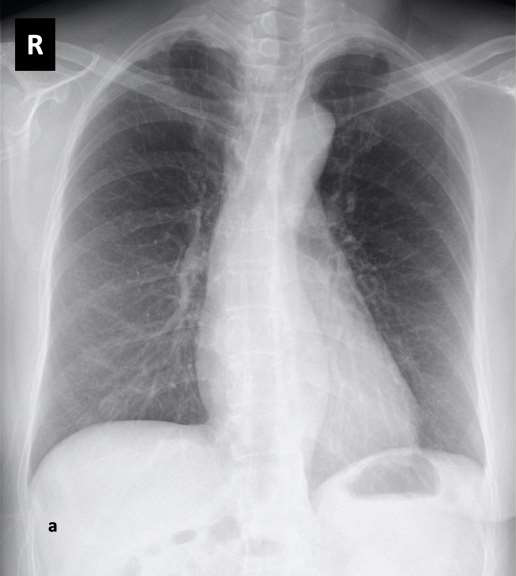
The selection of base learner may vary from one problem to another, but the essence is to select most appropriate models for the given problem. We selected five different models for classification, which are the state-of-art CNN architectures, as base learners. These base learners are ResNet101, InceptionV3, MobileNetV2, NasNet and Xception. Due to their varied structure, they possess different capability to generalize the given distribution.

They applied a weighted fusion technique to combine the outputs of these models, assigning weights based on their algorithm. The final prediction of the ensemble model was determined using a weighted sum. The model's performance was evaluated on two datasets, each containing chest X-ray images categorized as normal, Covid-19, or pneumonia. In this research, they achieved an accuracy of 94.66% on Dataset-2 using the weighted ensemble technique.\

**Methodology:**

Dataset: The data was gathered from the COVID-19 Radiography Database on Kaggle. There are now 125 X-ray images (43 female instances and 82 male instances) in the database that have been diagnosed with COVID-19. In the database, that have been found to be positive. There isn’t enough information for all of the patients in this dataset. The average age of 26 COVID-19 positive participants is around 55 years old, according to the age information provided. In addition, pictures of normal and pneumonia were available from the ChestX-ray8 database.

* Sample images from train set for each class:

Covid-19 No-findings Pneumonia

To circumvent the problem of imbalanced data, the authors that used this dataset, randomly selected 500 no-findings and 500 pneumonia class frontal chest X-ray pictures from this database. We took the dataset exactly in the same manner from above mentioned reference research paper as they did for our experiments.

In this project, we utilized a this dataset, the dataset was divided into train, val, and test sets using a 60-20-20 split ratio. This approach ensures that 60% of the data is used for training the models, allowing them to learn patterns effectively, while 20% is reserved for validation to fine-tune the model parameters and prevent overfitting. The remaining 20% is used for testing the model's performance on unseen data, ensuring its generalizability.

Models:

In our project, we utilized lightweight base learners, which offer an efficient solution for systems with limited hardware resources. This approach allows models to be trained on basic systems without the need for high-end hardware, making it both cost-effective and accessible.

Additionally, it is highly time-efficient, as model training does not require extended durations. Below, we provide a detailed explanation of the models and techniques used in this project, demonstrating the practicality and effectiveness of this method.

**CNNs (Convolutional Neural Networks) :**

Convolutional networks (LeCun, 1989), also known as convolutional neural networks, or CNNs, are a specialized kind of neural network for processing data that has a known grid-like topology. Examples include time-series data, which can be thought of as a 1-D grid taking samples at regular time intervals, and image data, which can be thought of as a 2-D grid of pixels. Convolutional networks have been tremendously successful in practical applications. The name “convolutional neural network” indicates that the network employs a mathematical operation called convolution. Convolution is a specialized kind of linear operation. Convolutional networks are simply neural networks that use convolution in place of general matrix multiplication in at least one of their layers. In this way CNNs plays crucial role in image classification tasks.

In a regular Neural Network there are three types of layers:

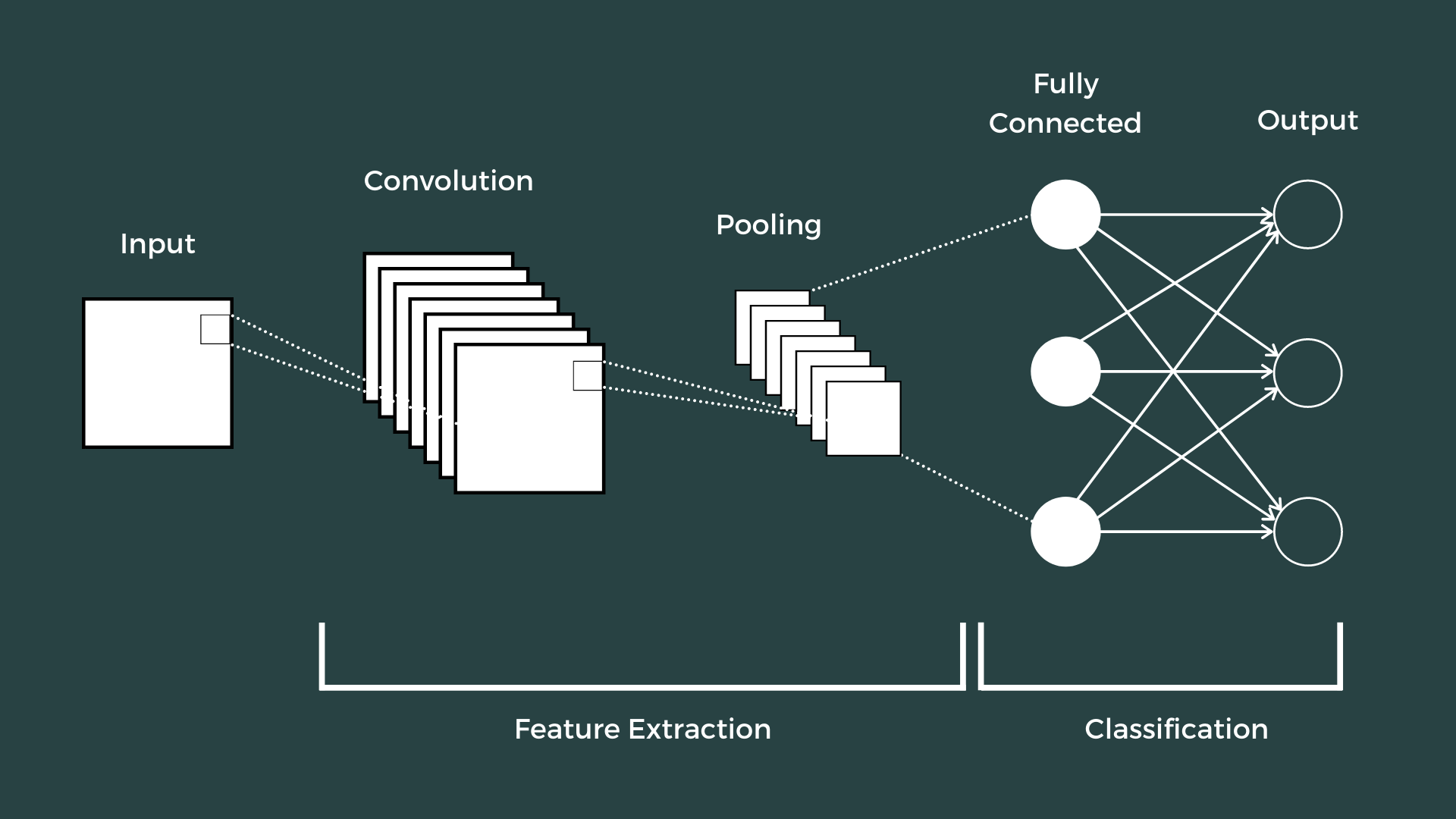
1. Input Layers: It’s the layer in which we give input to our model. The number of neurons in this layer is equal to the total number of features in our data (number of pixels in the case of an image).
2. Hidden Layer: The input from the Input layer is then fed into the hidden layer. There can be many hidden layers depending on our model and data size. Each hidden layer can have different numbers of neurons which are generally greater than the number of features. The output from each layer is computed by matrix multiplication of the output of the previous layer with learnable weights of that layer and then by the addition of learnable biases followed by activation function which makes the network nonlinear.
3. Output Layer: The output from the hidden layer is then fed into a logistic function like sigmoid or softmax which converts the output of each class into the probability score of each class.

The data is fed into the model and output from each layer is obtained from the above step is called [feedforward](https://www.geeksforgeeks.org/understanding-multi-layer-feed-forward-networks/), we then calculate the error using an error function, some common error functions are cross-entropy, square loss error, etc. The error function measures how well the network is performing. After that, we backpropagate into the model by calculating the derivatives. This step is called [Backpropagation](https://www.geeksforgeeks.org/backpropagation-in-data-mining/) which basically is used to minimize the loss.

Advantages of CNNs:

1. Good at detecting patterns and features in images, videos, and audio signals.
2. Robust to translation, rotation, and scaling invariance.
3. End-to-end training, no need for manual feature extraction.
4. Can handle large amounts of data and achieve high accuracy.

**Architecture of CNNs:**

****

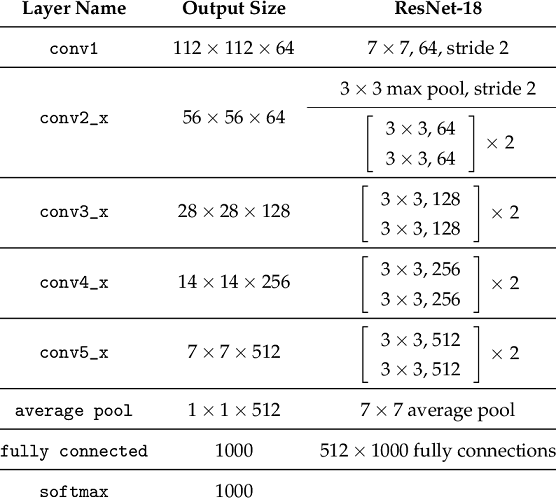
How Convolutional Layers Works?

Convolution Neural Networks or covnets are neural networks that share their parameters. Imagine you have an image. It can be represented as a cuboid having its length, width (dimension of the image), and height (i.e the channel as images generally have red, green, and blue channels).

Imagine taking a small patch of this image and running a small neural network, called a filter or kernel on it, with say, K outputs and representing them vertically. Now slide that neural network across the whole image, as a result, we will get another image with different widths, heights, and depths. Instead of just R, G, and B channels now we have more channels but lesser width and height. This operation is called Convolution. If the patch size is the same as that of the image it will be a regular neural network. Because of this small patch, we have fewer weights.

**1.ResNet18:**

ResNet-18 is a convolutional neural network that is trained on more than a million images from the ImageNet database. There are 18 layers present in its architecture. It is very useful and efficient in image classification and can classify images into 1000 object categories. The network has an image input size of 224x224.



Architecture of ResNet18

Consider the above diagram. From this diagram we can see how layers are configured in the ResNet-18 architecture. First there is a convolution layer with 7x7 kernel size and stride 2. After this there is the beginning of the skip connection. The input from here is added to the output that is achieved by 3x3 max pool layer and two convolution layers with kernel size 3x3, 64 kernels each. This was the first residual block.

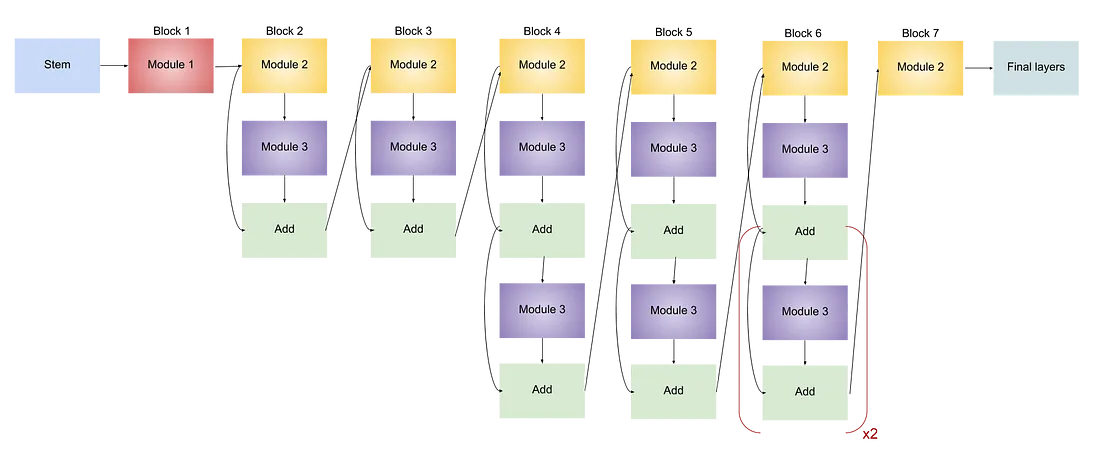
Then from here, the output of this residual block is added to the output of two convolution layers with kernel size 3x3 and 128 such filters. This constituted the second residual block. Then the third residual block involves the output of the second block through skip connection and the output of two convolution layers with filter size 3x3 and 256 such filters. The fourth and final residual block involves output of third block through skip connections and output of two convolution layers with same filter size of 3x3 and 512 such filters.

Finally, average pooling is applied on the output of the final residual block and received feature map is given to the fully connected layers followed by softmax function to receive the final output. The output of each layer is shown in the diagram and input is changed in the skip connections according to that.

**2.EfficientNetB0:**

EfficientNetB0 is part of the EfficientNet family, designed with a novel scaling method to improve accuracy while being computationally efficient. The key innovation is compound scaling, which jointly scales the depth, width, and resolution of a model using a compound coefficient. This helps the model maintain balance between these dimensions and ensures more efficient use of resources.

EfficientNetB0 uses Mobile Inverted Bottleneck Convolution (MBConv) layers combined with squeeze-and-excitation blocks, which enhance performance by focusing on important features in the data.

 Architecture of EfficientNetB0

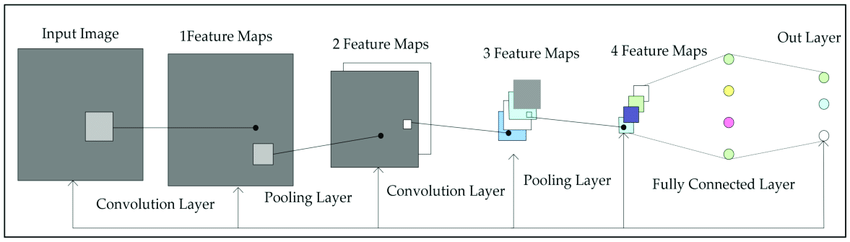
Key Features:

1. Compound Scaling: Unlike previous models, which scaled either depth, width, or resolution individually, EfficientNet scales all three uniformly to achieve better accuracy with fewer resources.
2. MBConv Layers: MobileNetV2's inverted residual blocks are used in EfficientNet, improving efficiency while reducing the number of parameters.
3. Squeeze-and-Excitation: This mechanism recalibrates the channel-wise feature responses, helping the model focus on the more critical features of an image.
4. Swish Activation Function: EfficientNet uses the swish activation function, which provides smoother gradients and improves model performance compared to ReLU.

This architecture allows EfficientNetB0 to achieve better accuracy on common datasets such as ImageNet, while using fewer parameters and FLOPs (Floating Point Operations). For instance, EfficientNetB0 achieves 77.3% accuracy on ImageNet using only 5.3 million parameters, much fewer than models like ResNet-50, which has around 25.6 million parameters.

**3. DenseNet121:**

DenseNet, short for Dense Convolutional Network, is a[deep learning](https://www.geeksforgeeks.org/deep-learning-tutorial/) architecture for [convolutional neural networks (CNNs)](https://www.geeksforgeeks.org/introduction-convolution-neural-network/) introduced by Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger in their paper titled “Densely Connected Convolutional Networks” published in 2017. DenseNet revolutionized the field of computer vision by proposing a novel connectivity pattern within CNNs, addressing challenges such as feature reuse, vanishing gradients, and parameter efficiency. Unlike traditional CNN architectures where each layer is connected only to subsequent layers, DenseNet establishes direct connections between all layers within a block. This dense connectivity enables each layer to receive feature maps from all preceding layers as inputs, fostering extensive information flow throughout the network.

**** Architecture of DenseNet121

Key Characteristics of DenseNet

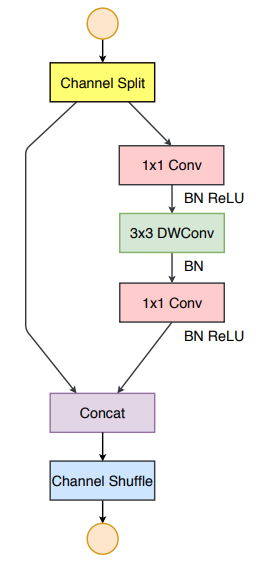
1. Alleviated Vanishing Gradient Problem: Dense connections ensure that gradients can flow directly to earlier layers, mitigating the vanishing gradient issue common in deep networks.
2. Improved Feature Propagation: Each layer has direct access to the gradients from the loss function and the original input signal, promoting better feature propagation.
3. Feature Reuse: By concatenating features from all preceding layers, DenseNet encourages feature reuse, reducing redundancy and improving efficiency.
4. Reduced Parameters: Despite its dense connections, DenseNet is parameter-efficient. It eliminates the need to relearn redundant features, resulting in fewer parameters compared to traditional networks.

DenseNet introduces a paradigm shift by connecting each layer to every other layer in a feed-forward manner. Unlike traditional CNNs, which have a single connection between consecutive layers, DenseNet ensures that each layer receives inputs from all preceding layers and passes its output to all subsequent layers. This results in a network with L(L+1)/2 direct connections for L layers, significantly enhancing information flow.

**4.ShuffleNetV2:**

ShuffleNet v2 considers direct metrics, such as speed or memory access cost, to measure the network’s computational complexity (besides FLOPs, which acts as an indirect metric). Moreover, the direct metrics are also evaluated on the target platform. ShuffleNet v2 was thus introduced in the paper, [ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design](https://arxiv.org/abs/1807.11164?ref=blog.paperspace.com), published in 2018. It was co-authored by Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun.

FLOPs is the usual metric to measure the performance of a network, in terms of its computations. However, a few studies have substantiated the fact that FLOPs do not wholly dig the underlying truths; networks having similar FLOPs differ in their speeds, this can be because of the memory access cost, degree of parallelism, target platform, etc. All these do not fall under FLOPs, and thus, are being ignored. ShuffleNet v2 overcomes such hassles by proposing four guidelines to model a network.

Prior to understanding the network architecture, the guidelines upon which the network has been built shall give a glimpse into how various other direct metrics have been considered:

1.Equal channel width minimizes the memory access cost: When the number of input channels and output channels are in the same proportion (1:1), memory access cost becomes low.

2.Excessive group convolution increases memory access cost: The group number shouldn’t be too high, otherwise the memory access cost tends to increase.

3.Network fragmentation reduces degree of parallelism: Fragmentation reduces the network’s efficiency in executing parallel computations.

4.Element-wise operations are non-negligible: Element-wise operations have small FLOPs, but can increase the memory access time.

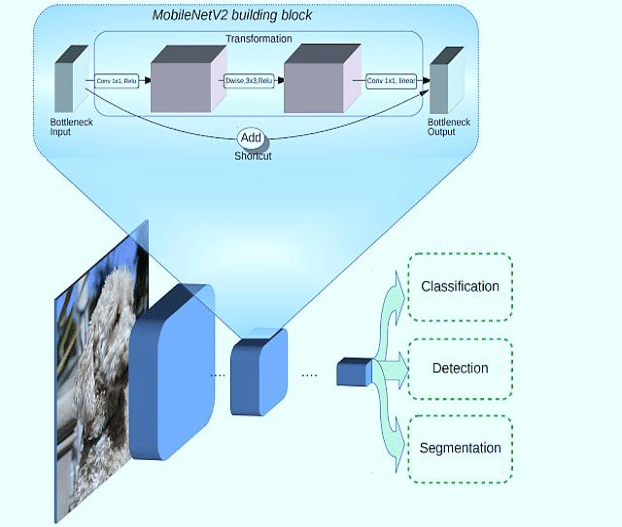
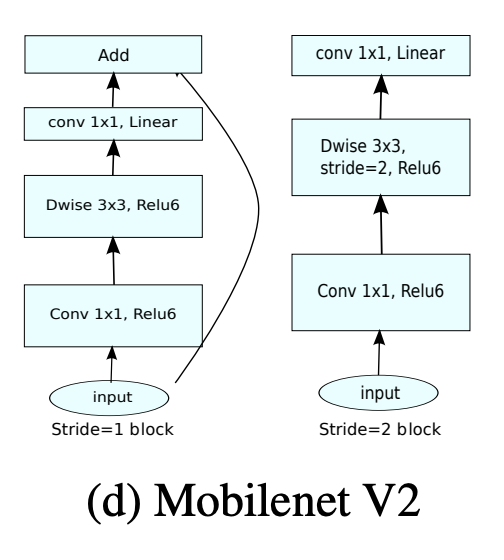
Architecture of ShuffleNetV2

All these are integrated in the ShuffleNet v2 architecture to improve the network efficiency.

The channel split operator divides the channels into two groups, where one remains as an identity (3rd guideline). The other branch has an equal number of input and output channels along the three convolutions (1st guideline). The 1x1 convolutions aren’t group-wise (2nd guideline). Element-wise operations like ReLU, Concat, depth-wise convolutions are confined to a single branch (4the guideline).

**5.MobileNetV2:**

MobileNetV2 is a [convolutional neural network](https://www.geeksforgeeks.org/introduction-convolution-neural-network/) architecture optimized for mobile and embedded vision applications. It improves upon the original MobileNet by introducing inverted residual blocks and linear bottlenecks, resulting in higher accuracy and speed while maintaining low computational costs. MobileNetV2 is widely used for tasks like [image classification](https://www.geeksforgeeks.org/cifar-10-image-classification-in-tensorflow/), [object detection, and semantic segmentation](https://www.geeksforgeeks.org/object-detection-vs-object-recognition-vs-image-segmentation/) on mobile and edge devices.



The architecture of MobileNet-v2 consists of a series of convolutional layers, followed by depthwise separable convolutions, inverted residuals, bottleneck design, linear bottlenecks, and squeeze-and-excitation (SE) blocks.

Depthwise Separable Convolution: Depthwise separable convolution is a technique used in MobileNetV2 to reduce the computational cost of convolutions. It separates the standard convolution into two separate operations: depthwise convolution and pointwise convolution. This separation significantly reduces the number of computations required, making the model more efficient.

Inverted Residuals: Inverted residuals are a key component of Mobilenetv2 architecture that helps improve the model’s accuracy. They introduce a bottleneck structure that expands the number of channels before applying depthwise separable convolutions. This expansion allows the model to capture more complex features and enhance its representation power.

Bottleneck Design: The bottleneck design in MobileNetV2 further reduces the computational cost by using 1×1 convolutions to reduce the number of channels before applying depthwise separable convolutions. This design choice helps maintain a good balance between model size and accuracy.

Linear Bottlenecks: Linear bottlenecks are introduced in MobileNet-v2 to address the issue of information loss during the bottleneck process. By using linear activations instead of non-linear activations, the model preserves more information and improves its ability to capture fine-grained details.

Experimental Setup:

In this project, we aimed to develop an ensemble model for classifying lung diseases using chest X-ray images, focusing on three classes: Covid-19, pneumonia, and normal. The experiments were conducted using the following methodology:

* Data Preprocessing: The dataset was divided into training (60%), validation (20%), and test (20%) sets. Images were resized to 224x224 pixels, and data augmentation techniques such as random horizontal flipping and rotation were applied to enhance model robustness.
* Model Architecture: We implemented several lightweight convolutional neural networks (CNNs), specifically ResNet18, EfficientNetB0, and MobileNetV2. Each model was initialized with pre-trained weights and modified to output three classes. The output layer was adapted by changing the last fully connected layer to accommodate the number of target classes.
* Training Process: The models were trained for a maximum of 15 epochs, utilizing the Adam optimizer with a learning rate of 0.0001 and weight decay of 0.001. Early stopping was implemented to prevent overfitting, with a patience of 5 epochs. The loss function used was Cross-Entropy Loss, suitable for multi-class classification tasks.
* Hyperparameter Tuning: Hyperparameters such as learning rate and batch size were fine-tuned to optimize model performance. The batch sizes were set to 8 for training and 4 for validation, ensuring efficient data loading and processing.
* Evaluation Metrics: Model performance was assessed using training and validation loss metrics. The training process was monitored, and the best model weights were saved based on the minimum validation loss. The training and validation loss were plotted to visualize the model's learning progression.
* Tools and Libraries: The experiments were conducted using Python with the PyTorch framework, leveraging libraries such as torchvision for model implementation and data handling.

After training, we evaluated each model using the test dataset, calculating accuracy and generating confusion matrices and classification reports to assess model performance comprehensively. Results were visualized with bar charts and heatmaps to illustrate the accuracy, confusion matrices, and classification metrics. This setup allowed us to leverage lightweight models effectively on limited hardware while achieving robust performance in medical image classification

.

Results:

This section presents the results of our experiments on the classification of COVID-19, Pneumonia, and Normal conditions using above mention convolutional neural networks: ResNet18, DenseNet121, ShuffleNetV2, EfficientNetB0 and MobileNetV2.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

* Findings: Presents the outcomes of your experiments (e.g., accuracy, confusion matrix).
* Analysis: Interprets the data, explaining trends, performance improvements, etc.
* Visuals: Includes graphs, tables, and charts to represent the results clearly.
* Discussion:
* Interpretation: Discusses the significance of the findings in relation to your objectives and the existing literature.
* Challenges: Addresses any difficulties faced during the research.
* Limitations: Acknowledges the constraints of your research and suggests areas for improvement.
* Conclusion:
* Summary: Recaps the main findings.
* Contributions: Highlights the contribution of your research to the field.
* Future Work: Suggests further research directions or potential advancements.
* References: Lists all the sources cited throughout the thesis in a proper citation format (e.g., APA, IEEE).
* Appendices (optional): Includes supplementary materials like code, raw data, or additional experiment details that are referenced in the thesis but too detailed for the main sections.

Resnet18 – 0.0001

MobilenetV2- 0.0001

ShuffleNet-0.0001

Desnsenet121- 0.0001

EfficientnetB0- 0.001

Adam for all