National College of Ireland

Analysing the Impact of Deep Learning and Data Augmentation on Medical Image Classification

MSc Research Project
Msc Data Analytics

Ayush Kumar
Student ID: 21208590

School of Computing
National College of Ireland

Supervisor: Dr.Abid Yaqoob

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Ayush Kumar
Student ID:	21208590
Programme:	Msc Data Analytics
Year:	2023
Module:	MSc Research Project
Supervisor:	Dr.Abid Yaqoob
Submission Due Date:	14/12/2024
Project Title:	Analysing the Impact of Deep Learning and Data Augmentation on Medical Image Classification
Word Count:	6796
Page Count:	20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	31st January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Analysing the Impact of Deep Learning and Data Augmentation on Medical Image Classification

Ayush Kumar
21208590

Abstract

Computer vision techniques are implemented and utilized in different fields. In large laboratories, medical image classification plays a pivotal role in the diagnosis of diseases. In this research, computational power is harnessed to identify patients suffering from pneumonia using images of frontal chest X-rays as dataset for deep learning. Pneumonia is a condition in which lung air sacks are infected by bacteria or viruses. Pneumonia can be diagnosed in any age group but it is most common in young children and elderly people. Deep learning convolution neural network(CNN) models implemented in research experiment are ResNet, VGG, EfficientNet, GoogleNet and DenseNet for medical image classification. In the research experiment novel approach is implemented to conduct a comprehensive investigation of the influence of different deep learning and data augmentation techniques on all five distinct CNN models. The research experiment is conducted in two different phases, phase one models are implemented without data augmentation to systematically test multiple hyperparameters and in phase two, the optimal test case scenario of phase one is utilized for the implementation of data augmentation techniques. The assessment of these models in research experiment involves evaluation metrics like accuracy, recall, precision and loss. The analysis of these performance metrics of all distinct models is conducted with the intention of determining the optimal scenario.

1 Introduction

Pneumonia is a highly infectious disease causing severe damage to the lungs, in this inflammatory condition the patient's respiratory system does not function properly. The infectious virus has the capacity to damage the tissue present lungs, usually, pneumonia affects one of lungs, but in some fatal cases both the lungs are infected by bacteria. Pneumonia is caused by infectious bacteria, viruses, fungi or other pathogens, that affect the air sacks called alveoli. As the result of affected alveoli, it becomes difficult for the patient to breathe as lungs are filled a bodily fluid or purulent material(pus). This causes the depleting level of oxygen in the blood as pus creates issues in the transaction of oxygen, which causes a patient to suffer chills, fever and cough with pus. Pneumonia can either be acute or chronic, this is completely dependent on the duration of the infection and condition. The best or standard process of diagnosis of pneumonia can be performed by analysing the x-rays (World Health Organization and others (2001)). This disease is common in infants and elderly people but can be diagnosed in smokers, people exposed

to impure air like miners and patients hospitalized. Pneumonia is an airborne disease, if diagnosed and not treated in early stages then it could be fatal.

1.1 Background and Motivation

Recently, around 740,180 children died suffering from pneumonia accounting for 14% of all deaths of children under the age of five years World Health Organization (2023). The initiative of W.H.O. provides information about precautionary steps like immunization and adequate nutrition.

Medical image classification is an active research field as researchers around the globe work on improved approaches for image classification using computer vision. Deep learning algorithms provide an extensive model or foundation stone for medical image classification. In continued research in the classification of images, the researcher's main objective is to provide a highly accurate and robust model that can diagnose diseases of patients using MRI, CT-scan, x-rays and other medical reports. In recent research of Showkat and Qureshi (2022), the authors proposed a transfer learning-based ResNet model for the detection of infected tissue present in the lungs due to COVID-19 using frontal chest x-rays. The authors also conclude from their research that initial training data plays a vital role in the development of a robust model for medical image classification as the accuracy of the model is 95.65 percent and the precision of 92.74 percent.

1.2 Research Question and Objectives

How will deep learning techniques and data augmentation techniques affect the performance of deep learning models for medical image classification?

In this research, the experiment is designed as multiple test hyperparameters on five pre-trained models with same data and layers to analyse the different performances of deep learning model. Using the optimal hyperparameters, data augmentation is applied for an enhanced comparative study of effect of data augmentation on deep learning model. In the evaluation, a discussion on the impact of data augmentation techniques and hyperparameter tuning is completed for more insights.

1.3 Research Contributions

Hyperparameter tuning and data augmentation techniques are crucial in deep learning models, to find optimal model performance and to make the model robust and reliable. In medical image classification, hyperparameter tuning and data augmentations are used to improve the model's ability to generalize. In the step of data preparation/ preprocessing, hyperparameters and augmentation techniques are applied for the training model. In hyperparameter tuning a well-chosen batch size can affect the model stability and coverage of the dataset for efficient usage of computational power while preventing overfitting. Data augmentation helps the model to understand and classify images effectively as medical images of patients a diverse because of multiple factors like lighting, position, resolution of image and other factors that make data augmentation techniques essential for the medical image classification model. The research conducted by Monshi et al. (2021), In this research authors have used deep learning model in diagnosis of COVID in patients

with the help of chest x-rays. Researchers explained the importance of hyperparameter tuning and data augmentation tuning for enhancement of their results, researchers were able to achieve an accuracy of 95.82% in the detection of patients suffering from COVID.

2 Related Work

2.1 Medical Image Classification

Deep learning algorithms help in medical image segmentation and classification. In this era, large amounts of medical data include MRIs, electronic health records, x-rays and other medical reports. In the research of Cai et al. (2020) authors have classified and segmented large amounts of medical data with the help of deep learning algorithms. The data include diabetic retinopathy, gastric cancer pathology segmentation, pulmonary nodule disease(lung disease) and diagnosis of early-stage of Alzheimer's disease. Authors have implemented multiple models in the analytical study, as in the research experiment Mask-RCNN gets a high precision score in hippocampus segmentation, meanwhile authors also pointed out that Unet-3D performed far better than traditional Unet. This research provides a comprehensive analysis of model implementation on medical data with a singular foundation.

In the research of Kumar et al. (2018) a research experiment was conducted, in which a brain tumor was classified from MRI images. In this research dataset was quite small as three hundred fifty-four images were present. In this research experiment authors proposed a hybrid support vector machine(SVM) with a particle swarm optimization algorithm. While implementing the experiment the proposed mode performed impressive as the accuracy attain by the proposed model was ninety-five percent and traditional SVM model was able to achieve sixty-eight percent. This research experiment provides with understanding of layers and their usage for enhanced results.

The researchers in Chauhan et al. (2021) proposed an automated deep learning model to diagnose COVID-19 using less computational time. In this research experiment fine tuning and optimization of DenseNet architecture is executed. In order to optimize deep learning model authors suggest the implementation of data augmentation techniques and normalization of the training dataset. Other techniques used in the research experiment consist of early stopping, multiple optimizer, loss function and LR scheduler for achieving better performing model. The authors utilize multiple optimizer out of which the Adamax optimizer with cross-entropy loss function outperformed with an accuracy of 98.45 %. This research paper plays a vital role in understanding of application of loss functions and optimizers in implementation of deep learning models.

2.2 Lung Diseases Detection

CAD is widely accepted and utilized in large hospitals and laboratories all around the globe. In 2019, researchers published a research for using deep convolutional neural network(DCNN) to classify lung cancer in patient's CT scan. The authors Mohanapriya et al. (2019) proposed a classifier in the detection of tumors and further classify the tumor as either malignant or benign. In the architecture of CNN model six layers and softmax classifier. For evaluation precision, recall, accuracy and dicescore are considered.

The experimental architecture performed adequately. Finally in conclusion discuss the implementation of DCNN for other human parts like the breast, liver and brain. This research provides with a crucial process of implementation of layers for model development.

Artificial Neural Network(ANN) is extensively utilized for image segmentation, In 2015 researchers Adetiba and Olugbara (2015) utilized the basic rule ANN for prediction of lung cancer in patients using x-ray images. This research extensively compares the ANN and Support Vector Machine(SVM) ensembles for lung cancer prediction The experiment was performed using genomic feature present in the Orient Gradient Histogram(HOG). the authors also used Voss DNA encoding for mapping nucleotide sequences. ANN ensemble with HOG outperformed SVM ensemble. The authors conclude that HOG genomic features have the potential of automatic screening and detect lung cancer in its early stages.

In December 2019 COVID-19 virus was reported in Wuhan, China. The researcher provided an approach to automated diagnosis of COVID-19 using EfficientNet (Marques et al. (2020)). In this research experiment authors proposed has been validated using 10-fold cross-validation. Research experiments consist of binary classification and multi-class classification. the datset used in this research is not robust, hence author suggested the utilization of ALbumentation and ImageAugmentor model. In conclusion, the proposed EfficientNet architecture was able to classify 99.62% of binary and 96.70% of multi-class classification.

2.3 Pneumonia Detection

Medical image classification programs help large laboratories in diagnosis of diseases. In the research of Yadav and Jadhav (2019) authors have implemented multiple approaches for deep learning model inorder to have comprehensive analysis of research experiment. The authors have used frontal chest x-ray for detection of patients suffering from pneumonia, by implementing VGG16 and InceptionV3 for diagnosis. In this research experiment, authors also applied data augmentation to enhance the performance of the models. The authors concluded that the traditional CNN- based transfer learning approach performed best. This research helps in understanding of application of layers for robust models and gives a future scope of using more powerful CNN-based models Like ResNet.

In 2019, researchers provided an efficient approach to CNN deep learning model for scratch(Stephen et al. (2019)). In this research experiment, the authors proposed a new architecture of scratch CNN model which does not rely on traditional ways of transfer learning or traditional architecture. In experiment authors provided the model with different case scenarios, in which the size of training gradually increased from 100 to 300. This research provides the explanation of how the size of the dataset affect the accuracy of models.

In same year of 2019, a comparative study was performed, in this research of Sharma and Guleria (2023) the experiment focuses on the detection of pneumonia patients using chest X-rays. Here, VGG16 was applied with other neural networks like Random Forest, K-Near Neighbor(KNN), Niave Bayes and SVM. The experiment was conducted on two different CXR image datasets for diverse results. The proposed model in research outperformed all the other comparison models used for pneumonia detection. Not only

the accuracy was increased, sensitivity and precision of the proposed model also increased.

In the research of Varshni et al. (2019) the authors proposed a densely connected neural network(DenseNet-169) model. In this proposed model architecture was divided into three stages as pre-processing stage, feature extraction stage and classification stage. In classification stage Random Forest, Support Vector Machine(SVM) were used. pre-tained models such as VGG16, VGG19, ResNet-50, DenseNet-121 and DenseNet-121 were implemented for comperative analysis. The proposed model of combining CNN-based feature extraction and supervised classifier outperformed the existing models. This research provides with insightful information but it lacks the application of data augmentation for extensive training of deep learning algorithms.

2.4 Data Augmentation on deep learning

In the research of Bisogni et al. (2022) the authors used deep learning model in classifying the human facial expression with three different datasets. Firstly the authors extracted the facial recognition box that uses features extracted from the recognition box used for classifying the expression of a person. In this research, the authors have fine-tuned hyperparameters and used multiple data augmentation techniques. In conclusion, the researchers provide an explanation of how data augmentation techniques help in improving the performance of models and with the help of hyperparameter tuning, the models improve the ability to recognize the facial expression better.

In the year 2019,the researcher explained that deep learning models are highly dependent on big training data so that models train properly. The researcher provided with a survey on application of data augmentation on Convolutional Neural Network(CNN) model(Shorten and Khoshgoftaar (2019)) with small dataset. The researchers provide an extensive insight into different types of data augmentation techniques. In this research experiment the authors have used basic augmentation techniques like flipping, erasing, random zoom and others. For comparative analysis of results authors have also used Generative Adversarial Network (GAN) based augmentation. The datasets used in research experiments are diverse in fields. In conclusion proposed with the statement that data augmentation can not overcome all the biases present in small datasets while providing data augmentation solutions to the problem of overfitting in deep learning models with small datasets.

An electrocardiogram(ECG) records the electrical signals from the heart to check the condition. In 2021, a research(Anwar and Zakir (2021)) was conducted that discussed the diverse effect of data augmentation techniques on EfficientNetB3 deep learning model with learning rate of 0.0001 and two optimizers(Adam and AdamW). The authors in the research experiment applied basic augmentation techniques such as random cropping, flipping with a probability of 0.7, distortion with the scale of 0.1, image contrast, and gamma applied in order to increase the number of data images. While concluding researcher explained that deep learning models are data-hungry and require big data for better performance. To provide different perspective image augmentation techniques applied, this increases the size of data but destroys the hidden pattern present in data

causing a decrease in performance. The researchers proposed that with the help of data augmentation techniques performance of a model enhanced to an extent after that peak threshold point the performance of the model suffered.

After analysis of multiple research papers, it provides better understanding of data augmentation and other deep learning techniques on model implementation. The research earlier lacks some intriguing points like comparative analysis with the same foundational code and input data. In multiple researches hyperparameter tuning is specific to the particular model. Researchers are providing extensive information on the application of data augmentation, but very less research experiments conduct a comparative analysis of model performance.

3 Methodology

In medical image classification, the flow of information needs to be systematic to extract meaningful patterns and insights for effective understanding of the process. The fundamental Knowledge Discovery in Databases(KDD) methodology is adopted in this research experiment. KDD provides guidance to the framework in comprehensive and structured flow of information/data while working on this research experiment while keeping in mind the objectives and steps were used in order as presented in Figure 1. Implementing KDD in the research field of science is more straightforward as researchers have a thorough understanding of data(Fayyad et al. (1996)). Incorporation of KDD in the research experiment of medical image classification ensures in-depth knowledge of meaningful insights, which furnish optimal results for classification.

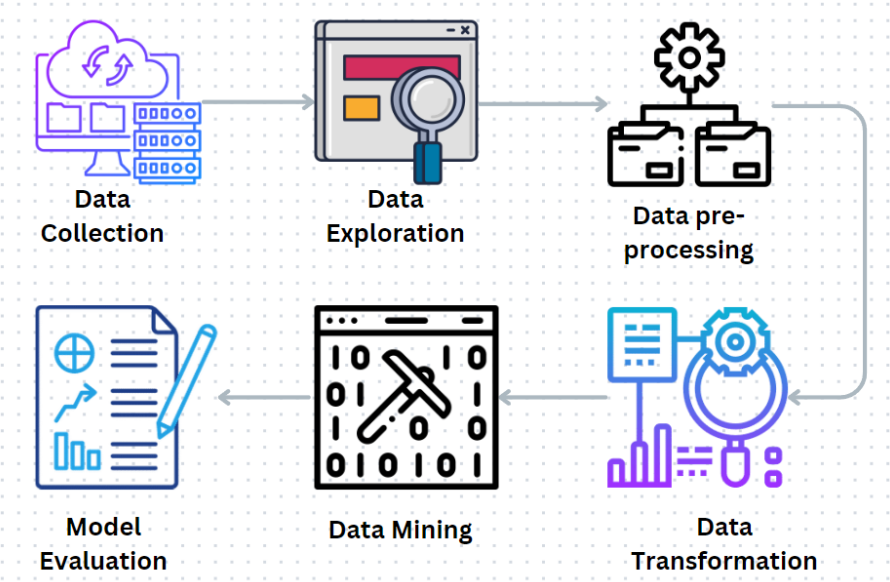

Figure 1: Knowledge Discovery in Databases(KDD)

3.1 Data Acquisition

The dataset utilized in the research experiment consists of chest X-rays. The radiographic images of chest X-rays are of one to five-year-old children from Guangzhou Women and Children's Medical Center in Guangzhou, China. The dataset containing medical images were professionally labeled by a group of scientists (Kermany et al. (2018)). The dataset consists of 5,863 x-ray images of patients which are categorised in two sections of 1,583 Normal images and 4,273 images of Pneumonia patients as shown in Figure 2. The dataset consists of three folders of train, test and val, each folder consists of subfolders of normal and pneumonia x-rays. The complete database of 5,856 x-ray images is uploaded by Paul Mooney on the Kaggle website as a zip file of 2.3GB. [1]

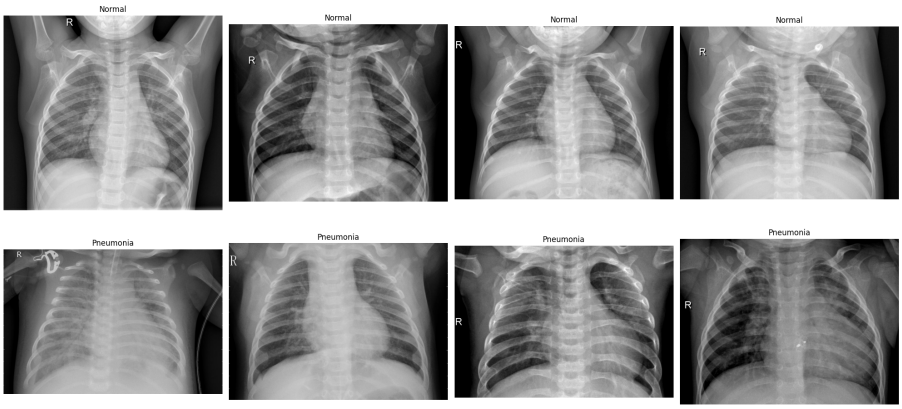

Figure 2: Normal and Pneumonia x-rays

3.2 Exploratory Data Analysis

In the exploration of data of medical image classification, various crucial steps are taken for comprehensive understanding and optimal utilization of the image dataset. A thorough examination of pixel intensity distribution, class distribution and reading images for better understanding, this information from data exploration helps in model building. As mentioned earlier the dataset is categorized in two pneumonia and normal, all the x-rays in both of the categories are counted as shown in Table 1.

Table 1: Image Dataset.

Folder	**Category**	**Number of images**
Train	pneumonia	3875
	Normal	1341
Test	pneumonia	390
	Normal	234
Validation	pneumonia	8
	Normal	8
Total		5856

[1] `https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia`

In Figure 3, a sample image of normal x-rays and pneumonia is visualized and analysed. In this process the basic pixel-level information is extracted from both the x-rays. The statistical analysis of image pixels provides the basic difference between a normal chest X-ray and an X-ray of a patient suffering from pneumonia.

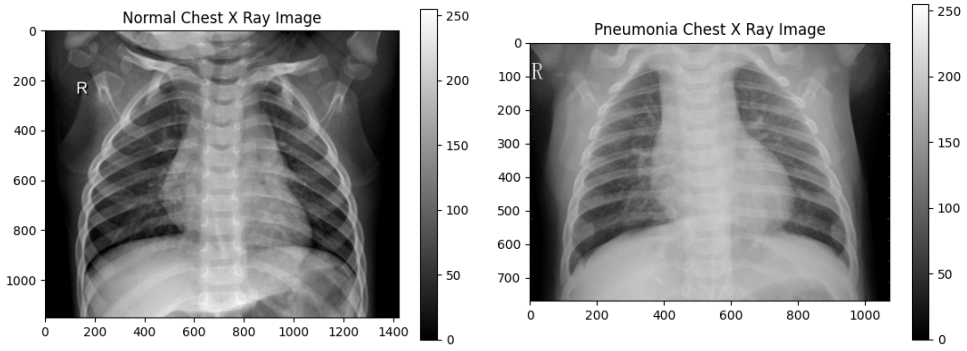

Figure 3: Pixel Distribution in Normal and Pneumonia X-rays

The dimensions of X-ray images are 1152 pixels and 1422 pixels in height with a singular colored channel. The mean value of pixel for a normal X-ray is 100.6506 and the standard deviation is 59.8083 on the other hand for a pneumonia patient's X-ray, mean value is 127.3646 and the standard deviation is 57.4101. The pixel intensity distribution is shown in Figure 4. This histogram provides a visual representation of the distribution of pixel intensities in the chest x-ray images. The histogram enlightens the information about the number of pixels in image falling within different intensity ranges.

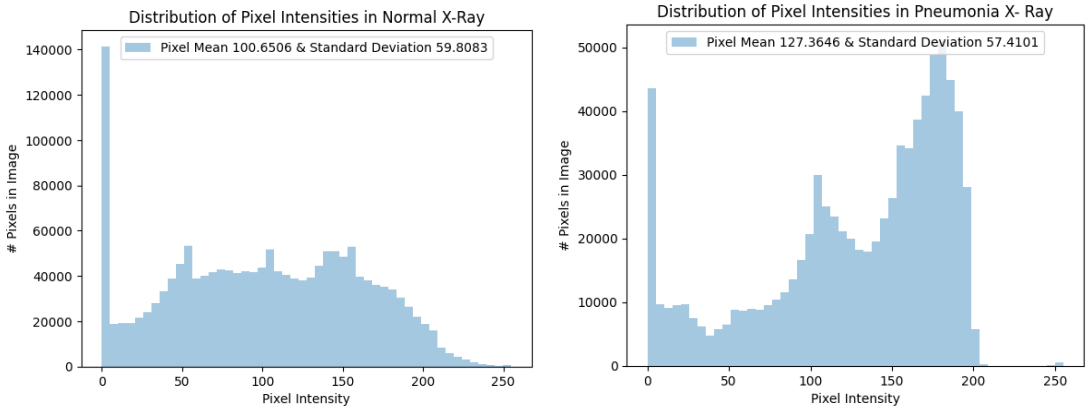

Figure 4: Histogram about Distribution of Pixel Intensity

Analysing the class weight distribution is a crucial step, it helps in understanding the dataset whether it is biased or not. As shown in Figure 5, the total number of images in our dataset of pneumonia is 74 percent in comparison to 26 percent of normal cases. In this case of weight distribution the model might exhibit the tendency to prioritize the majority class in training i.e. pneumonia class.

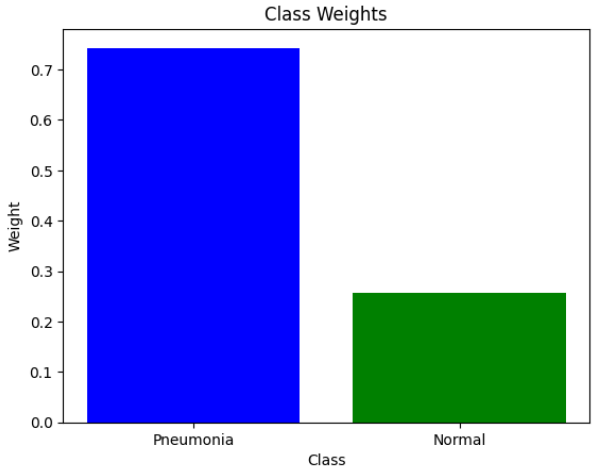

Figure 5: Class Weight Distribution

3.3 Data Pre-processing and Transformation

Data pre-processing and transformation play a crucial role in the implementation of the model. The data is transformed and then split into segments of train, test and validation. As mentioned earlier, the dataset is already segmented into three folders of train, test and val. Using the "ImageDataGenerator" function present in the Keras library, transformed data in batches of images while resizing the images into 180 x 180 for efficient loading into deep learning model.

3.4 Model Implementation

In this research experiment, five pre-trained models ResNet50, VGG16, EfficientNetB0, InceptionV3 and DenseNet121 are implemented. These pre-trained models are imported from the Keras library, all these models are implemented with additional layers. The experiment is designed in two main phases one implementation without data augmentation and the other with data augmentation. Multiple hyperparameter tuning is executed as changes in batch sizes, steps per epochs and different values of augmentation parameters.

3.5 Model Evaluation

In the analysis of research experiments, a traditional approach of considering multiple evaluation metrics is analysed with keeping the objective of research in mind. In order to assess the performance and efficiency of model accuracy, recall, precision and binary cross-entropy loss values are considered. Analysis of these metrics provides insights into the performance and robustness of deep learning models.

Accuracy: The accuracy of the model is the overall correctness of the model. It is the ratio of prediction and total values in the dataset. For medical image classification accuracy of the model is not the main attribute to suggest the performance of the model as usually the dataset provided is biased

$$Accuracy = \frac{True\ Positives + True\ Negatives}{Total\ Population} X100 \tag{1}$$

Precision: Precision is the ratio of correctly positive prediction with respect to total positive prediction.

$$Precision = \frac{True\ Positives}{True\ Positives\ +\ False\ Positives} X100 \tag{2}$$

Recall or Sensitivity: Recall is correctly predicted positive with respect to true positives and false negatives.

$$Recall = \frac{True\ Positives}{True\ Positives\ +\ False\ Negatives} X100 \tag{3}$$

Dice Score: In medical image classification, dice score or F1 score plays a crucial role in testing the performance of the model. It is an overlap of predicted and true positives.

$$Dice\_Score = \frac{2 \cdot (\text{Predicted} \cap \text{True Positives})}{\text{Total Predicted Positives} + \text{Total True Positives}} \tag{4}$$

4 Design Specification

This research experiment analyzes the impact of deep learning techniques and data augmentation techniques on deep learning models for medical image classification. The experiment is done in two different phases. In Figure 6, the flow of the experiment is depicted.

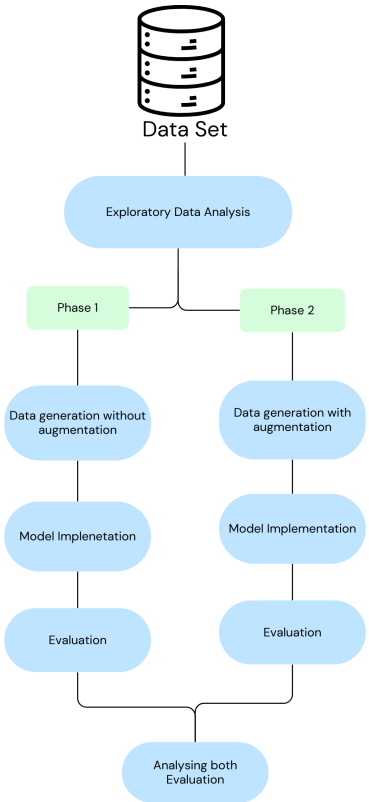

Figure 6: Design Flow of Experiment

Phase 1: the data collected from the local drive and loaded to dataframe. Then data exploration is executed in which class weights are checked, pixel intensity and statistical

analysis of the image are performed. As mentioned earlier in the section on data acquisition the dataset is segmented into 3 different folders test, train and validation. Hence with the help of *ImageDataGenrator* function data is pre-processed. Here no parameters of data augmentation techniques are given to function. In order to get ready data for model implementation the batch size value i.e. 4,8 and 16 is provided. Other hyperparameters like shuffle, class mode, target size and seed values are given to *flow_from_directory* function. This function automatically infers the class labels based on sub-directories. These batches of images are provided to the model as training data, testing data and validation data.

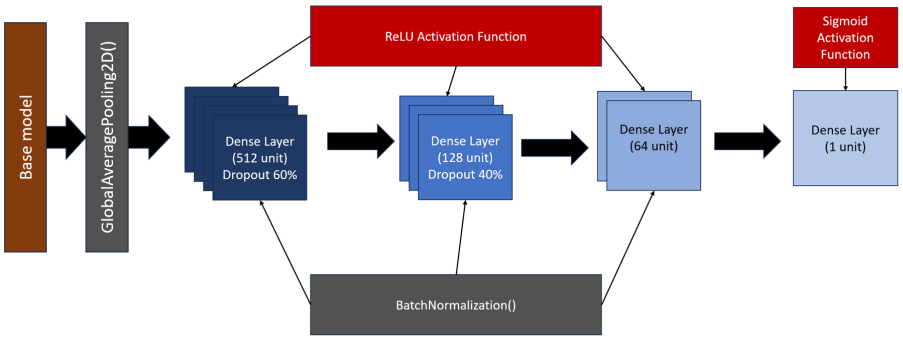

Figure 7: Additional Layers

Input data for training, testing and validating is set to be 180x180 pixels with three color channels(RGB). All five pre-trained CNN models are provided with the same batches of input data. The pre-trained CNN model is used as the base model further layers are applied on the base model providing the output layer with a single unit and *sigmoid* activation function as shown in Figure 7, it is most suitable for binary classification. The new model is implemented with *Adam* optimizer and loss is set to *binary_crossentropy*. Later steps per epoch are provided to model in *fit* function. All five pre-trained models are implemented with the same parameters for comparative analysis of the performance of the deep learning model. In phase one the hyperparameter tuning is implemented in order to improve the performance of models. The changes are done to the values of batch size and *steps_per_epoch*. These test scenarios help in understanding the impact of deep learning techniques on model performance.

Phase 2: In this phase, the Python program follows the same foundation of loading, EDA, model implementation and result evaluation. The difference occurs in data pre-processing as in this phase *ImageDataGenerator* function is provided with data augmentation techniques as parameters. The data augmentation techniques used in the experiment are *random rotation, width shift, zoom and sample wise normalization*. These techniques transform the input data of all five pre-trained CNN models. The performance of all the models analyzed and evaluated at last. In phase two hyperparameter test scenario consists of changing the values of data augmentation techniques. To check whether it affects the performance of the model or not.

Systematically the results provided by both of the phases are further analysed to check the impact of data augmentation on the performance of the model for medical image classification.

5 Implementation

5.1 Environment Setup

In the research experiment, the computational power requirement was high to implement deep learning model. Hence, a new GPU environment is created for faster execution with the help of the terminal present in Anaconda terminal. Major libraries installed in the environment are tensorflow==2.10, CudaToolkit=11.2 and Cudnn=8.1 with other libraries like numpy, pandas, seabon and matplotlib. The hardware specifications are present in table 2.

Table 2: Hardware Specification

Hardware	**Specification**
Processor	Intel(R) Core i7-7700HQ @ 2.8GHZ
RAM	16 GB
ROM	512 GB SSD
GPU	NVIDIA GTX 1050
Operating System	Windows 11 (64 Bits)

5.2 Programming Language and Platform

In this research experiment, python programming language is utilized in the implementation of deep learning models. Python programming language provides a huge sum of libraries and packages which are crucial for data analysis and deep learning tasks. For scripting python language, Jupyter Notebook is used as an interactive development environment(IDE) supported by Anaconda Navigator, it integrates with a web browser(Google Chrome) and code executes in cells which is helpful in debugging or finding errors in program code.

5.3 Model Implementation

In this research experiment, Five pre-trained models are used for comparative analysis of their performance. A traditional working CNN model is shown in Figure 8 [2].

[2]image source: https://www.upgrad.com/blog/basic-cnn-architecture/

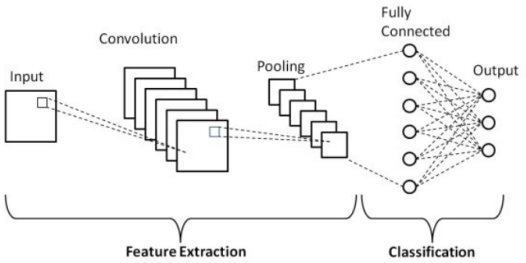

Figure 8: Traditional Convolutional Neural Network(CNN)

In this research experiment, all five CNN architectures are used as base models and over them additional layers are applied as shown in Figure 7.

5.3.1 ResNet50

In 215, ResNet(Residual Network) is introduced by researchers in their research paper(He et al. (2016)). ResNet uses residual blocks in architecture which uses skip connection or identity mappings as links in two blocks for the flow of information. This skip connection supports the training of an extremely deep network. This allows the model to understand residuals while differentiating between the input layer and output layer. ResNet50 is a variation of ResNet architecture, the number 50 indicates the total number of layers in the network. Traditional Block of the residual block consists of two convolutional layers with batch normalization and ReLU activation function. It is widely implemented in computer vision tasks such as objective detection, image classification or segmentation.

5.3.2 EfficentNetB0

EfficeintNetB0 is the smallest variant of EfficientNet architecture, it is introduced in 2019 by Tan and Le (2019). It is designed in a manner to balance computational efficiency with competitive performance. The key characteristics of EfficientNetB0 are compound scaling, Efficient building blocks, resolution with depth and width. The EfficientNet application depends on specific requirements as it demands less computational resources.

5.3.3 VGG16

VGG16 stands for Visual Geometry Group 16, It was introduced in 2014 by Simonyan and Zisserman (2014) at the University of Oxford. VGG16 has 16 weight layers as 13 convolutional layers and 3 connected layers. It has a simple and uniform structure with the filter size of 3x3. It may not be efficient with computational resources but provides straightforward yet effective architecture for various image classification problems.

5.3.4 InceptionV3

InceptionV3 was introduced in 2016 by Szegedy et al. (2016), it comes from the family of GooLeNet which was developed at Google. It provides parallel convolutional operation of different filter sizes. It is well known for the efficient use of computational resources and the ability to capture diverse features and patterns through inception modules.

5.3.5 DenseNet121

In 2017, DenseNet121 was introduced by Huang et al. (2017), it provides the concept of dense connectivity where early layers are connected to every other layer in feed-forward manner. The densely connected blocks enhance feature reuse, this improves parameter efficiency. The dense connectivity enables the model to capture patterns and reuse them to improve gradient flow during training.

6 Evaluation

this research experiment is segmented into phases. In phase one of the experiment, the models are not provided with augmented data. In the stage of data transformation no data augmentation is applied to data images. In phase two while pre-processing data, data augmentation techniques are applied for data transformation,

In order to check the effect of *Steps_per_epoch* on model performance, in one execution steps are decided as 100 and in other execution steps are decided in traditional fashion where the number of steps = total sample/ batch size.

6.1 Batch Size 4

6.1.1 Steps= Total sample / Batch Size

Here, Batch size while pre-processing is decided as 4. The models are implemented with input data with size of 4. The Figure 9, shows the training accuracy of Python program with batch size 4 and no augmentation techniques applied to it. The Figure 10, shows the evaluation of the training accuracy of the model with input data of batch size 4 and data augmentation techniques are applied to transformed data.

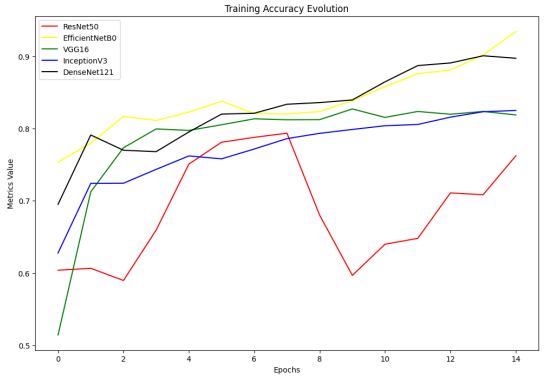

Figure 9: Training Accuracy Evolution

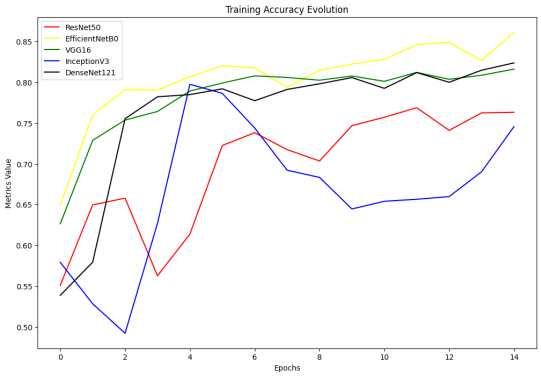

Figure 10: Training Accuracy Evolution with Data Augmentation

The results produced in the first experiment are shown in Table 3. [3]

[3]Accuracy, Precision, Recall are in percentage

Table 3: Evaluation Metrics of batch size 4 and steps are total sample/ batch size

Model	Training Accuracy	Testing Accuracy	Precision	Recall	Dice Score
ResNet50	88.03681135177612	79.487181	78.854626	91.794872	0.848341
ResNet50(With DA)	80.4064154624939	62.980771	63.184083	97.692305	0.767372
EfficientNetB0	81.6717803478241	89.423078	89.705884	93.846154	0.917293
EfficientNetB0(With DA)	90.4524564743042	81.891024	80.573952	93.589741	0.865955
VGG16	93.30905079841614	74.679488	72.745097	95.128202	0.824444
VGG16(With DA)	93.88419985771179	79.487181	76.518220	96.923077	0.855204
InceptionV3	92.63803958892822	83.012819	80.735928	95.641023	0.875587
InceptionV3(With DA)	79.52454090118408	63.942307	63.591433	98.974359	0.774323
DenseNet121	93.1173324584961	83.012819	80.603451	95.897436	0.875878
DenseNet121(With DA)	92.82975196838379	76.282054	73.359072	97.435898	0.837004

6.1.2 Step decided as 100

While experimenting with the steps = total sample/ batch size, the program took around 4 to 5 hours to implement as in each epoch the model is trained on complete data. Hence decided to decrease the number of steps for faster results. The model is trained where *Steps_per_epoch* is decided as 100. The Figure 11 shows the training accuracy evolution of the model without augmented data and Figure 12 shows the evolution of model accuracy in which data augmentation is applied.

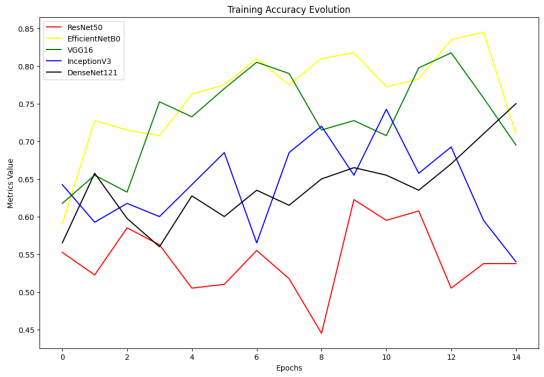

Figure 11: Training Accuracy Evolution

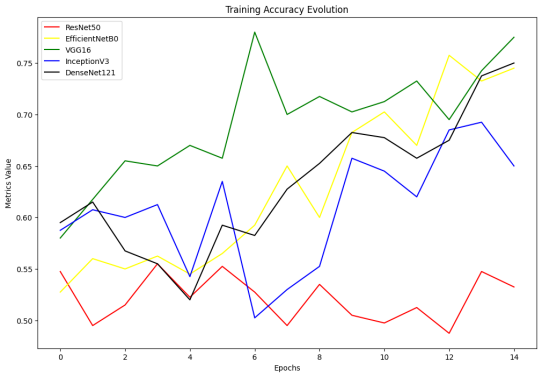

Figure 12: Training Accuracy Evolution with augmented data

All the evaluation metrics are present in Table 4 of this research experiment.

Table 4: Evaluation Metrics

Model	Training Accuracy	Testing Accuracy	Precision	Recall	Dice Score
ResNet50	39.55138027667999	48.717949	73.648649	27.948719	0.405204
ResNet50(With DA)	58.358895778656006	58.173078	71.009773	55.897439	0.625538
EfficientNetB0	30.8090478181839	39.903846	80.000001	5.128205	0.096386
EfficientNetB0(With DA)	35.486963391304016	43.429488	83.636361	11.794872	0.206742
VGG16	75.90107321739197	62.820512	62.700963	100.000000	0.770751
VGG16(With DA)	85.19938588142395	70.192307	70.647776	89.487177	0.789593
InceptionV3	71.85583114624023	70.512819	69.360900	94.615382	0.800434
InceptionV3(With DA)	53.50843667984009	55.929488	83.625734	36.666667	0.509804
DenseNet121	73.58129024505615	68.750000	74.074072	76.923078	0.754717
DenseNet121(With DA)	83.89570713043213	80.448717	88.728327	78.717947	0.834239

6.2 Batch Size 8

6.2.1 Steps= Total sample / Batch Size

Here, Batch size while pre-processing is decided as 8 because the model performance on batch size 4 was not adequate. The models are implemented with input data and testing data as size 8. The Figure 13, shows the training accuracy of Python program with batch size 4 and no augmentation techniques applied to it. The Figure 14, shows the evaluation of the training accuracy of the model with input data of batch size 4 and data augmentation techniques are applied to transformed data.

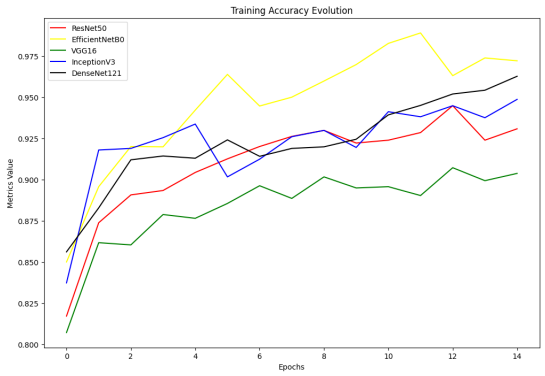

Figure 13: Training Accuracy Evolution

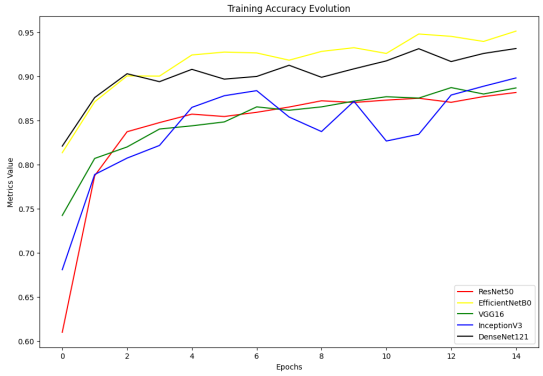

Figure 14: Training Accuracy Evolution with augmented data

the results for batch size 8 are shown in Table 5.

Table 5: Evaluation Metrics

Model	Training Accuracy	Testing Accuracy	Precision	Recall	Dice Score
ResNet50	94.95782256126404	79.807693	76.720649	97.179484	0.857466
ResNet50(With DA)	92.5230085849762	85.096157	83.521444	94.871795	0.888355
EfficientNetB0	95.3987717628479	90.384614	88.732392	96.923077	0.926471
EfficientNetB0(With DA)	74.29064512252808	62.500000	62.500000	100.000000	0.769231
VGG16	94.40184235572815	67.628205	65.878379	100.000000	0.794297
VGG16(With DA)	95.6480085849762	80.128205	76.181102	99.230766	0.861915
InceptionV3	98.42791557312012	78.205127	74.517375	98.974359	0.850220
InceptionV3(With DA)	91.33435487747192	72.756410	69.855595	99.230766	0.819915
DenseNet121	97.39263653755188	87.500000	83.620691	99.487180	0.908665
DenseNet121(With DA)	95.70552110671997	83.653843	80.000001	98.461539	0.882759

6.2.2 Step decided as 100

While experimenting with steps= total sample/batch size, the program took around 3 to 4 hours to implement as in each epoch the model is trained on complete data. Hence decided to decrease the number of steps for faster results. The model is trained where *Steps_per_epoch* is decided as 100. The Figure 15 shows the training accuracy evolution of the model without augmented data and Figure 16 shows the evolution of model accuracy in which data augmentation is applied.

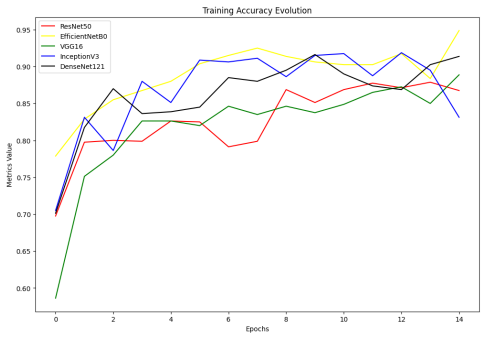

Figure 15: Training Accuracy Evolution

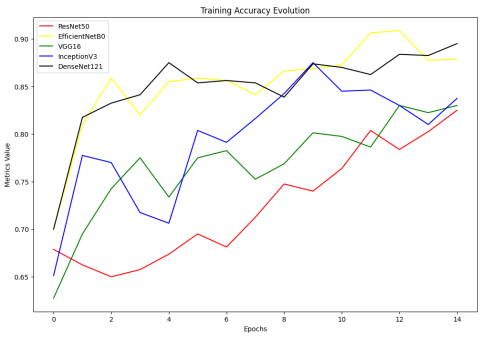

Figure 16: Training Accuracy Evolution with augmented data

All the evaluation metrics are present in Table 6 of this research experiment.

Table 6: Evaluation Metrics

Model	Training Accuracy	Testing Accuracy	Precision	Recall	Dice Score
ResNet50	86.77147030830383	83.974361	85.714287	89.230770	0.874372
ResNet50(With DA)	81.09662532806396	73.397434	72.672063	92.051280	0.812217
EfficientNetB0	97.85276055335999	85.096157	81.528664	98.461539	0.891986
EfficientNetB0(With DA)	74.29064512252808	62.500000	62.500000	100.000000	0.769231
VGG16	93.67331266403198	70.833331	68.439716	98.974359	0.809224
VGG16(With DA)	81.36503100395203	63.942307	63.502455	99.487180	0.775225
InceptionV3	34.54754650592804	40.224358	55.704701	21.282052	0.307978
InceptionV3(With DA)	92.695552110672	79.166669	76.315790	96.666664	0.852941
DenseNet121	97.23926186561584	81.891024	77.867204	99.230766	0.872604
DenseNet121(With DA)	84.87346768379211	88.461536	97.321427	83.846152	0.900826

6.3 Batch size 16

While executing the code for batch size 16, the *ResourceExhaustedError* is in the middle of training of model process. As the GPU utilized in the research experiment is 4GB, the training model process ran out of space. To tackle this issue a threshold was also applied of 50 % utilization for GPU space, but it did not work as in the next model training again faced with *ResourceExhaustedError*.

7 Discussion

Both segments of the research experiment unfolded new insight to analyze the impact of deep learning and data augmentation techniques on medical image classification.

In the initial stage of the research experiment, multiple hyperparameters were considered for execution. The batch sizes 4, 8, 16 and 32, due to the limited size of the medical image dataset and constrained computational resources, for batch 16 and 32 result in failure with an error of *ResourceExhaustedError*. For batch sizes 4 and 8, the model performed well providing satisfactory results. In phase one of the experiment while execution, Where steps are decided as total sample/batch size performed better in comparison to where steps are decided as 100. But if we take computational time into consideration, the time difference in the execution of Python code is quite large. Usually, a file with higher number of steps takes almost 1 to 2.5 hours extra in comparison to a file with a

100-step file, The training model runs through a complete dataset in one epoch which increases the chances of redundancy while training the deep learning model.

Data Augmentation techniques have a large impact on the performance of models, as cross-entropy loss was calculated while the training model severely decreased, overall performance accuracy of model also increased this enhanced the dice score evaluation metric of model. In some cases, the performance of the model decreased as data augmentation affected the precision and recall of the model.

In the comparative analysis of model performance for batch size 4, the VGG16 model outperformed other models in medical image classification with high sensitivity or recall value. The dice score of VGG16 Network with and without augmented data was better with consideration of other evaluation metrics. VGG16 provides a simple feed-forward network and uses data provided to train the model efficiently. For the batch size of 8, provides diverse results as all models performed quite well.

8 Conclusion and Future Work

This research was carried out to investigate the impact of data augmentation and deep learning techniques on deep learning models for the detection of pneumonia patients using frontal chest X-rays of patients. The comparative analysis of the performance of the model on same training data provides a large amount of insight. The medical datasets are highly biased as the process of collecting medical data about a particular disease is focused on investigating the features of a particular disease. In this research experiment dataset weight was distributed in the form of 74% of pneumonia cases and only 26% of normal chest X-rays. The collection of medical data has a high cost and complexity of image procedures. There are multiple efforts made to enhance diversity within the dataset. In 2020, research of Fujiwara et al. (2020), explains the steps to be taken in order to enhance diversity in the data. In conclusion, the authors explain that there are multiple limitations of collected data.

Deep learning techniques such as batch size, steps per epoch, learning rate and other hyperparameters plays a vital role in the performance of deep learning models. Data augmentation techniques are also helpful to an extent as medical image data a highly biased. With fine-tuning of input data performance of models can be enhanced.

In conclusion, this research contributes in the understanding of the impact of data augmentation and deep learning techniques on different deep learning architectures. Data augmentation can enhance the performance of the model to an extent but with more advanced augmentation techniques and hyperparameter tuning more robust and reliable model can be developed for medical image classification. The research might act as a foundation for understanding the working impact of data augmentation and deep learning techniques on deep learning models for medical image classification.

9 Acknowledgement

I would like to express my sincere gratitude to my supervisor Dr. Abid Yakoob for his valuable guidance and support in navigating me through the research project. Continuous support by him enriches the quality of the research experiment.

References

Adetiba, E. and Olugbara, O. O. (2015). Lung cancer prediction using neural network ensemble with histogram of oriented gradient genomic features, *The Scientific World Journal* **2015**: 786013.
URL: *https://www.hindawi.com/journals/tswj/2015/786013/*

Anwar, T. and Zakir, S. (2021). Effect of image augmentation on ecg image classification using deep learning, *2021 International Conference on Artificial Intelligence (ICAI)*, pp. 182–186.

Bisogni, C., Castiglione, A., Hossain, S., Narducci, F. and Umer, S. (2022). Impact of deep learning approaches on facial expression recognition in healthcare industries, *IEEE Transactions on Industrial Informatics* **18**(8): 5619–5627.

Cai, L., Gao, J. and Zhao, D. (2020). A review of the application of deep learning in medical image classification and segmentation, *Annals of translational medicine* **8**(11).

Chauhan, T., Palivela, H. and Tiwari, S. (2021). Optimization and fine-tuning of densenet model for classification of covid-19 cases in medical imaging, *International Journal of Information Management Data Insights* **1**(2): 100020.
URL: *https://www.sciencedirect.com/science/article/pii/S2667096821000136*

Fayyad, U. M., Haussler, D. and Stolorz, P. E. (1996). Kdd for science data analysis: Issues and examples., *KDD*, pp. 50–56.

Fujiwara, K., Huang, Y., Hori, K., Nishioji, K., Kobayashi, M., Kamaguchi, M. and Kano, M. (2020). Over- and under-sampling approach for extremely imbalanced and small minority data problem in health record analysis, *Frontiers in public health* .
URL: *https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7248318/*

He, K., Zhang, X., Ren, S. and Sun, J. (2016). Deep residual learning for image recognition, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K. Q. (2017). Densely connected convolutional networks, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708.

Kermany, D., Zhang, K. and Goldbaum, M. (2018). Labeled optical coherence tomography (oct) and chest x-ray images for classification, Mendeley Data.

Kumar, A., Ashok, A. and Ansari, M. A. (2018). Brain tumor classification using hybrid model of pso and svm classifier, *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pp. 1022–1026.

Marques, G., Agarwal, D. and de la Torre Díez, I. (2020). Automated medical diagnosis of covid-19 through efficientnet convolutional neural network, *Applied Soft Computing* **96**: 106691.
URL: *https://www.sciencedirect.com/science/article/pii/S1568494620306293*

Mohanapriya, N., Kalaavathi, B. and Kuamr, T. s. (2019). Lung tumor classification and detection from ct scan images using deep convolutional neural networks (dcnn), *2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, pp. 800–805.

Monshi, M. M. A., Poon, J., Chung, V. and Monshi, F. M. (2021). Covidxraynet: Optimizing data augmentation and cnn hyperparameters for improved covid-19 detection from cxr, *Computers in Biology and Medicine* **133**: 104375.
URL: *https://www.sciencedirect.com/science/article/pii/S0010482521001694*

Sharma, S. and Guleria, K. (2023). A deep learning based model for the detection of pneumonia from chest x-ray images using vgg-16 and neural networks, *Procedia Computer Science* **218**: 357–366.

Shorten, C. and Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning, *Journal of big data* **6**(1): 1–48.

Showkat, S. and Qureshi, S. (2022). Efficacy of transfer learning-based resnet models in chest x-ray image classification for detecting covid-19 pneumonia, *Chemometrics and Intelligent Laboratory Systems* **224**: 104534.
URL: *https://www.sciencedirect.com/science/article/pii/S0169743922000454*

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition, *arXiv preprint arXiv:1409.1556* .

Stephen, O., Sain, M., Maduh, U. J., Jeong, D.-U. et al. (2019). An efficient deep learning approach to pneumonia classification in healthcare, *Journal of healthcare engineering* **2019**.

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

Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks, *International conference on machine learning*, PMLR, pp. 6105–6114.

Varshni, D., Thakral, K., Agarwal, L., Nijhawan, R. and Mittal, A. (2019). Pneumonia detection using cnn based feature extraction, *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pp. 1–7.

World Health Organization (2023). Pneumonia in children.
URL: *https://www.who.int/news-room/fact-sheets/detail/pneumonia*

World Health Organization and others (2001). Standardization of interpretation of chest radiographs for the diagnosis of pneumonia in children, *Technical report*, World Health Organization.
URL: *https://iris.who.int/bitstream/handle/10665/66956/WHO$_{V}$and_{B0}1.35.pdf*

Yadav, S. S. and Jadhav, S. M. (2019). Deep convolutional neural network based medical image classification for disease diagnosis, *Journal of Big Data* **6**(1): 113.
URL: *https://doi.org/10.1186/s40537-019-0276-2*