PREDICTION OF PEDIATRIC PNEMONIA WITH CHEST XRAY

Lavanya Sreedhar
Department of Computing and Mathematics
Manchester Metropolitan University
Manchester, United Kingdom

***Abstract*—In children under the age of five, pneumonia is the most common cause of morbidity and mortality worldwide. Even while the developing world accounts for the vast majority of pediatric pneumonia-related mortality, the developed world also bears a sizable share of the disease's heavy burden and its high healthcare expenses (bookshelf, 2000). Deep learning is a significant technique for upcoming applications and is offering fascinating answers for challenging medical image analysis issues. Chest X-ray (CXR) imaging plays an essential role in the diagnosis of pediatric pneumonia; however, it can be difficult to interpret these pictures without specialized knowledge. Deep learning approaches have recently showed promise in helping to diagnose a variety of illnesses, including pneumonia [1].**

Two Deep Learning Algorithms namely; AlexNet and VGG11 were trained on the dataset and their performances were compared based on various evaluation methods. From this research, it is found that all the models had a significantly high accuracy. However, VGG11 proved to be the ideal model in this study based on a better prediction of the minority class.

The goal of this work was to create a deep learning model that could forecast pediatric pneumonia from chest X-ray pictures. A Kaggle dataset containing a significant number of chest X-ray images from young individuals, both with and without pneumonia.

Keywords— Deep Learning, CNN, VGG11, AlexNet, Pediatric Pneumonia

I. INTRODUCTION

A common test for identifying paediatric pneumonia is a chest X-Ray [1]. Pneumonia is a kind of acute respiratory illness that affects the lungs. The tiny sacs called alveoli that make up healthy person's lungs are filled with air as they breathe. As a result of the accumulation of fluid and pus in the alveoli caused by pneumonia, breathing becomes harder and oxygen absorption is decreased. Worldwide, the majority of child mortality are caused by infectious infections. Pneumonia was the primary cause of 740 180 paediatric fatalities in 2019, accounting for 14% of all paediatric deaths in that age group but 22% of all paediatric deaths among children between 1 and 5 years [2].

Currently, the diagnosis of pneumonia is determined by the patient's symptoms, the results of a chest X-ray (CXR), the culture and sensitivity of the bacteria found in throat swabs or sputum samples, and blood samples. Early pneumonia identification is crucial in reducing complications because this illness is treatable and can be prevented, particularly through vaccination [3]. A further 18 million healthcare professionals would be required by 2030 to prevent, diagnose, and treat pneumonia, according to the World Health Organisation (WHO), which considers acute respiratory infections (ARI) to be the worst communicable disease affecting children [4].

Chest X-rays (CXRs), despite having a lower resolution than MRI or CT scans, can nevertheless be utilised to perform a variety of evaluations, including those for cardiomegaly, pneumonia, pneumothorax, and atelectasis. Using radiographs to diagnose pneumonia is very subjective and depends on the radiologist's knowledge and skill. High resolution MRI and CT scans make it simpler to detect pneumonia, however most radiologists prefer to use CXRs for assessments due to the faster turnaround time and economical nature of the modality. Radio-opacities or white spots in the airways, especially in the alveoli, are typically seen on a radiograph of pneumonia and signify the presence of inflammatory exudate. These radiological findings may present a challenge to a novice radiologist, leading to false positives and false negatives owing to the fact that other diseases mimic these signs [3].

The study uses deep learning to analyze chest X-ray pictures with the goal of developing a prototype automatic pneumonia detection system for pediatric patients. The model makes use of a multi-layered Convolutional Neural Network (CNN) to accurately correlate any pneumonia category by automatically extracting information from radiography pictures [3]. The Kaggle dataset is chosen since it includes X-Ray scans of pediatric patients with and without the condition. A suggested CNN model, VGG11, and AlexNet are trained on the dataset, and their performances have been tested on several evaluation matrices, as it is found from the examined literature that CNN along with VGG11, AlexNet models perform better on datasets including X-Ray images.

II. MATERIALS AND METHODS

I employed 4 methods to predict paediatric pneumonia in this study. The relevant libraries and dataset are loaded in the first portion before the entire dataset is resized. I employed 4 methods to predict pediatric pneumonia in this study. The relevant libraries and dataset are loaded in the first portion before the entire dataset is resized. The dataset is divided into sub-datasets for training, validation, and testing in the second step. In the third stage, the AlexNet CNN model and VGG11 is trained and validated using the training and validation datasets.

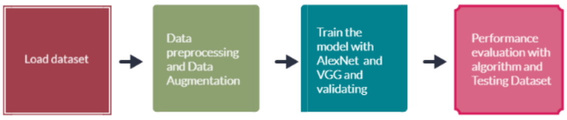

Fig: Flowchart of model processing

III. DATASET DESCRIPTION AND ANALYSIS

The dataset from Kaggle that can accessed by anyone. It is the paediatric patient dataset that Germany has made available. The Guangzhou Women and Children Medical Center's (GWCMC) paediatric CXR imaging dataset was used in this investigation and was made available online by Kermany et al. [5]

There are 5856 X-ray scans from paediatric patients between the ages of one and five in the collection. The most recent dataset is used, and it contains images that are split into train, train_valid, test, and valid (validation) folders, with two unique classes: "PNEUMONIA" images, and "NORMAL" images. The test folder has 690 total photos, 234 of which are of the NORMAL class and 390 of which are of the PNEUMONIA class. There are 4492 total images in the train folder, of which 989 are of the NORMAL class and 3503 are of the PNEUMONIA class. There are 140 photos in the train_valid folder, of which 60 are of the NORMAL class and 80 are of the PNEUMONIA class. There are 600 photos in the legitimate folder, 300 of which are NORMAL and 300 of which are PNEUMONIA class. Images from the collection were distributed among those used for viral, bacterial, and typical pneumonia.

Here I am going to show sample of the images of normal and pneumonia patients in dataset.

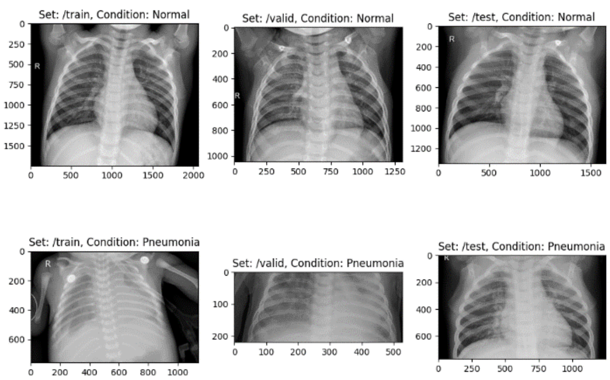

Fig: Pneumonia dataset images

IV. RELATED WORK

Deep learning algorithms have been utilized in several research to forecast pediatric pneumonia from chest X-ray pictures. Convolutional neural networks (CNNs) are utilized by these models to recognize and categories medical images generally, and pneumonia in particular. The field of medical imaging research has been interested in the use of chest X-ray images to diagnose pediatric pneumonia. For the purpose of predicting pediatric pneumonia, several research have looked at the use of machine learning algorithms and computer-aided diagnosis systems. We examine a few pertinent works that have advanced this field in this section.

A. *Pnemonia Identification by Rajpurkar*

In one study, Rajpurkar et al. (2017) developed a pneumonia identification method from chest X-ray images using a deep learning model. When it came to identifying pneumonia in adult patients, the model had good accuracy. Even though it wasn't directly about pediatrics, this work showed the promise of deep learning models for diagnosing pneumonia [9].

Rajpurkar et al. used a sizable dataset of chest X-ray images from over 30,000 patients from the National Institutes of Health (NIH) Clinical Centre in their study. Expert radiologists meticulously annotated the dataset, classifying each image as "normal" or "pneumonia" based on clinical findings and follow-up tests.

Convolutional neural networks (CNNs), a class of neural networks that perform particularly well in image processing tasks, were the deep learning model used by Rajpurkar et al. On the annotated dataset, the CNN was trained to identify the intricate patterns and characteristics of pneumonia in chest X-ray pictures. [9]

In order to reduce the discrepancy between the predicted labels of the model and the radiologists' reported ground truth labels, the model iteratively changed its internal parameters during training. The model was able to learn to predict pneumonia based on the patterns it found in the training dataset thanks to a method known as backpropagation.

The model was tested on a different test set of chest X-ray pictures following intensive training. In certain instances, the results outperformed human radiologists in terms of pneumonia detection accuracy. An area under the receiver operating characteristic curve (AUC-ROC) of 0.92 was reported by the study, demonstrating the model's potent capacity to distinguish between pneumonia cases and normal cases. [9]

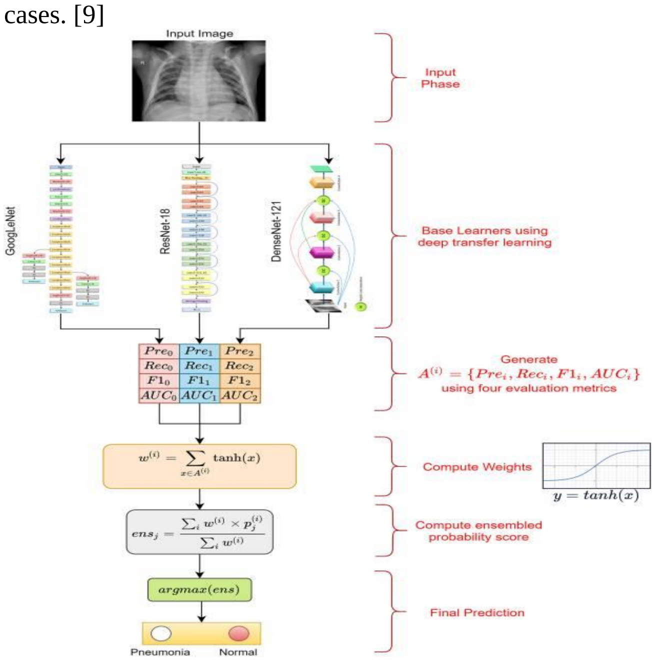

Fig: Illustration of the suggested framework for detecting pneumonia.

Although the study did not specifically address pediatric pneumonia, it did show that deep learning models have the capacity to diagnose pneumonia from chest X-ray pictures. The model's high level of accuracy in diagnosing pneumonia in adult patients raises the possibility of investigating comparable techniques for pediatric pneumonia diagnosis. As opposed to adult instances, pediatric cases may present particular difficulties because of variations in pneumonia presentation and changes in lung structure and disease manifestation.[9]

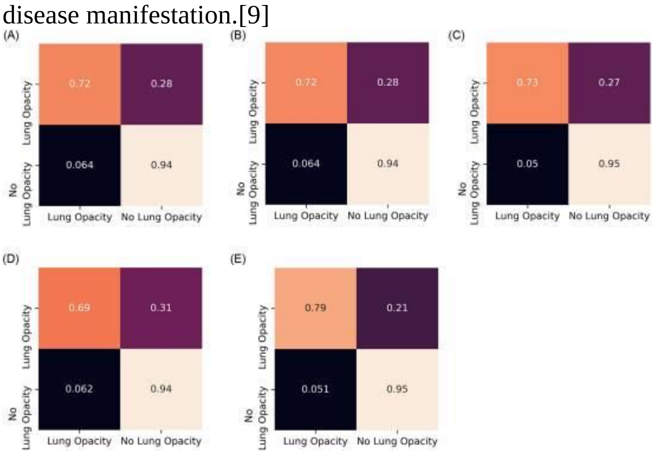

Fig: The suggested method by five-fold cross validation yielded confusion matrices on the pneumonia challenge chest X-ray dataset of the Radiological Society of North America. Chest X-ray dataset for the North America Pneumonia Challenge using the five-fold cross validation approach [9].

Overall, Rajpurkar et al.'s study demonstrated the value of deep learning models for detecting pneumonia from chest X-ray pictures. This study laid the groundwork for later research in the area and stimulated additional investigation of machine learning methods for pediatric pneumonia diagnosis [9].

B. Pneumonia classification by Elicker

In a similar line, Elicker et al. (2019) used chest X-ray images and a machine learning method to distinguish between viral and bacterial pneumonia in children. The classification of pneumonia aetiologias accurately showed promising results when the researchers trained a support vector machine classifier on features taken from the photos.[7]

For the purpose of their investigation, Elicker et al. created a collection of chest X-ray images from young people who had either viral pneumonia or bacterial pneumonia. The images were obtained from a hospital database and connected to medical information outlining the pneumonia's cause. These photos were thoroughly scrutinized and annotated by experienced radiologists [7].

Then, the researchers took a number of relevant features from the chest X-ray scans. These traits reflected distinguishing traits and patterns that might be able to distinguish between viral and bacterial pneumonia. The lung areas in the images' structural and textural characteristics were included in the features that were retrieved.[7]

The collected characteristics were then used as input to train the SVM classifier on the dataset. An artificial intelligence method called SVM is well renowned for its efficiency in binary classification problems. It functions by locating an ideal hyperplane that maximally divides instances of various classes.[7]

Based on the collected features, the SVM classifier was trained to differentiate between viral and bacterial pneumonia. The goal of the training was to maximize separation between the two pneumonia aetiologias and minimize classification error.

On a different test set of chest X-ray pictures, the researchers assessed the SVM classifier's performance after training. The study's findings showed that it was possible to distinguish between bacterial and viral pneumonia in children with a high degree of accuracy. High sensitivity and specificity were attained by the SVM classifier in properly identifying the cause of pneumonia [7].

The results of the study by Elicker et al. suggested that machine learning techniques like SVM might be useful in helping pediatric patients distinguish between viral and bacterial pneumonia. The classifier showed the ability to aid in more accurate diagnosis and tailored treatment choices by utilizing the particular features captured in chest X-ray pictures [7].

It is significant to note that additional analysis and validation are required to confirm the applicability and dependability of the SVM classifier for aetiologias categorization of pneumonia. However, this study showed a promising use of machine learning methods in diagnosing pediatric pneumonia, particularly in identifying various aetiologias based on chest X-ray pictures [7].

C. Pneumonia aetiology

The purpose of the 2018 study by Chassagnon et al. was to use chest X-ray images to distinguish between viral and bacterial pneumonia in pediatric patients. In order to categories the pneumonia aetiologias, the researchers investigated the use of machine learning techniques, specifically utilizing radiomic characteristics extracted from the photos [7].

Children who had either bacterial or viral pneumonia were the subjects of a dataset of chest X-ray pictures gathered by Chassagnon et al. for their study. The photos were taken from a hospital database and linked to clinical details describing the cause of pneumonia. To ensure correct labelling, seasoned radiologists evaluated and annotated the images [7].

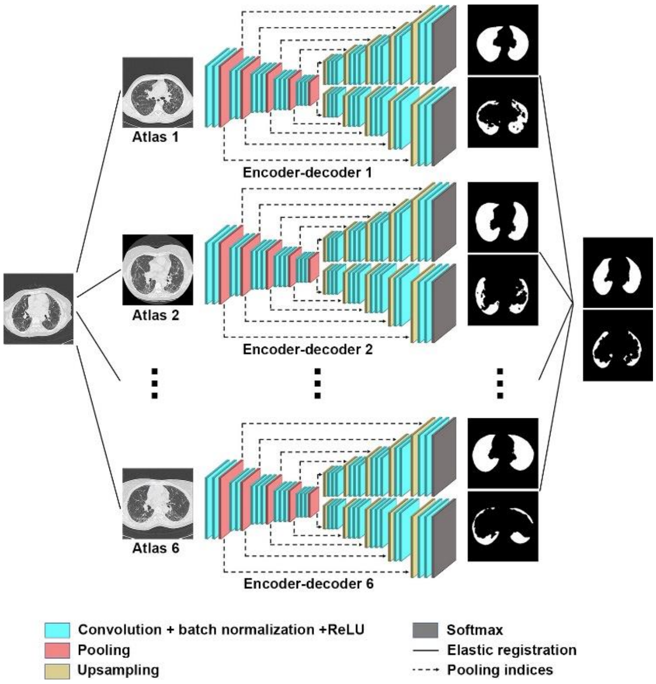

Fig: The AtlasNet framework's structure. rectified linear unit, or ReLU.[7]

After extracting the radiomic features, Chassagnon et al. constructed classification models utilizing machine learning methods including support vector machines (SVM) and random forests. The models were trained to determine whether pneumonia is brought on by a bacterial or viral illness based on the radiomic features that were extracted from the chest X-ray images.

Several metrics, including accuracy, sensitivity, specificity, and area under the receiver operating characteristic curve (AUC-ROC), were employed to evaluate the performance of the models. These metrics assess how effectively the models can differentiate between bacterial pneumonia and viral pneumonia. [7]

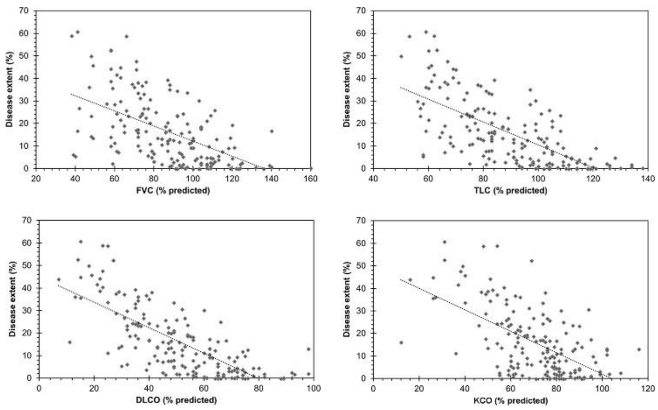

Fig: Relationship between algorithmic evaluations of the severity of interstitial lung disease caused by systemic sclerosis and results from pulmonary function testing. The abbreviations DLCO, FVC, KCO, and TLC stand for Diffusion Lung Capacity for Carbon Monoxide, Forced Vital Capacity, and Carbon Monoxide Transfer Coefficient, respectively [7].

The study's findings showed that chest X-ray pictures can be used to determine the cause of pediatric pneumonia with a high degree of accuracy. The machine learning models had high accuracy rates and showed good bacterial and viral pneumonia case differentiation in terms of AUC-ROC values.

It is crucial to keep in mind that the study was carried out in 2018, and that both medical research and technology have probably evolved since then. For the most recent information

on how to forecast paediatric pneumonia using a chest X-ray, it is therefore advised to research more recent literature or speak with healthcare specialists [7].

V. Methodology

A baseline CNN model, together with VGG11 and AlexNet, are used in this study three. It was necessary to initially load the data from the various folders before training the models. To perform the initial data analysis, this dataset was first stored and accessed from Google Drive storage. Thereafter, the dataset was combined with separate variables to create training, testing, and validation sets, and these sets were then expanded to create the models. Finally, the model performances were assessed using a variety of metrics.

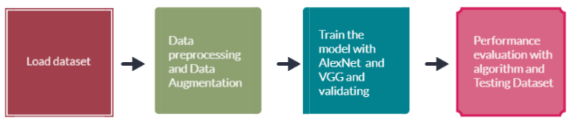

Fig: Diagram of the experiment

A. Data Preparation

Performing the procedures on a dataset while using the Google Collab environment. Data transformations definition: The transforms are used to define two sets of data transformations. Function compose (). The photos in the dataset will go through these changes to get them ready for training and testing.

'transform_train': These changes are made to the practice photos. The images are reduced in size to 32x32 pixels, transformed into tensors, and then normalised with means and standard deviations of [0.485, 0.456, 0.406] and [0.229, 0.224, 0.225] for each channel.
Transform_test: These adjustments are made to the test images. In order to ensure consistency in data processing, it carries out the same actions as transform_train.

created two datasets, train_ds and train_valid_ds, for training purposes. The dataset it represents is a training set. Training and validation datasets are pooled in train_valid_ds. Following the creation of valid_ds and test_ds, two more datasets were produced, with valid_ds serving as the validation dataset and test_ds as the testing dataset.
The images from the various folders in the data directory are arranged in these datasets, and each dataset is given the appropriate modifications. The training, validation, and testing phases of your deep learning model can then be carried out using these datasets.

Data loaders were produced using this strategy. For the training datasets, train_iter and train_valid_iter are two data loaders that are constructed. These data loaders can be made so that training, validation, and testing iterations of the datasets can be done quickly and effectively. Data shuffles, batches are created, and, if necessary, the final unfinished batch is dropped by the data loaders.

B. Data Augmentation

I used two tensors, means and stds, which are each initialized as tensors of zeros and have three members to calculate the means and standard deviations of the image collection. The means and standard deviations of the dataset will be collected using these tensors. You may find the average means and standard deviations of the dataset by completing this calculation. For normalization and preprocessing phases in your deep learning pipeline, these values can be helpful.

The functions "normalize_image()" and "plot_images()" were used to normalize the image. Image parameter in the normalize function is used to normalize the input image. The minimum and maximum values of the image tensor will be determined in order to normalize the image. returns the normalized picture after dividing it by image_max - image_min + 1e-5 (a tiny constant provided for numerical stability) and normalizing it by removing image_min from each element.

The 'plot_images()' method was used to plot an image.A grid of photos and their related labels are plotted using this function. The images are calculated using the formula n_images = len(images): the number of images in the images list is calculated. and depending on the square root of 'n_images', determines the grid of images' row and column dimensions. With each image having the possibility to be normalized if normalize=True is supplied, this function shows the images and their labels in a grid arrangement.
Overall, these routines can be helpful for normalizing photos and visually evaluating a series of photographs along with their accompanying labels.

Visualizing images

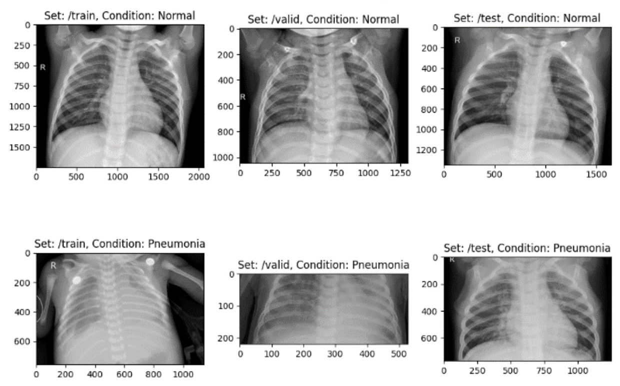

Fig: Visualizing images

Fig: pictures with pneumonic and normal X-ray images from the test, train, and valid folders.

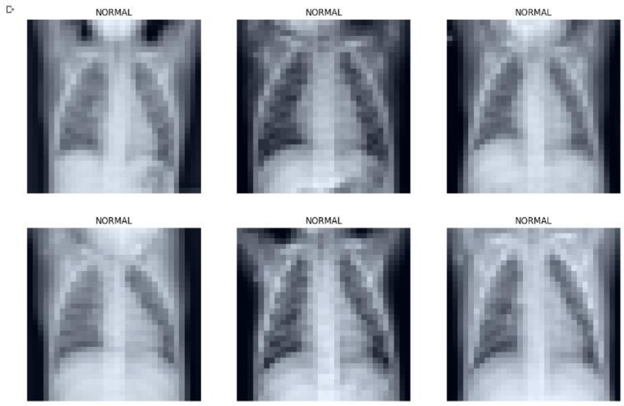

Fig: Images of normal X-ray

C. Model Architecture

I used the popular convolutional neural network (CNN) model VGG11, which is well-known for its efficiency in picture classification applications, and the AlexNet architecture. For the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), AlexNet, a well-known deep convolutional neural network design, was proposed in 2012. The code creates a class called AlexNet that derives from the base class for all PyTorch neural network modules, \`nn.Module\`. The number of output classes for the classification task is indicated by the \`output_dim\` argument.

The completely linked layers for categorization are defined by the \`self.classifier\` part. With dropout regularization and ReLU activation functions, it is made up of a series of linear (completely linked) layers. \`nn.To\` avoid overfitting, dropout randomly zeroes part of the input tensor's elements. \`nn.A\` layer of linear transformation is represented by linear.

The network's forward pass is put into practise through the forward method. In order to extract features, a tensor x input is passed to the \`self.features\` portion. A 1D tensor is created from the output of the feature extraction by flattening it using view (). The final output logits are obtained by passing this flattened tensor via the \`self.classifier\` component. In addition to the final output, the intermediate characteristics h is also returned.

This implementation offers the feature extraction and classification components as well as the fundamental framework for the AlexNet architecture. Please take note, nevertheless, that the code sample does not contain the full implementation of the training or testing procedures. The network's design and forward pass are its main concerns.

D. Model Initialization

After that, I called the initialize_parameters function, which is defined to initialize the model's parameters, to construct an instance of the model AlexNet. It checks the type of a module'm' that is provided as input. The weight is initialized using the Kaiming normal initialization approach with a ReLU nonlinearity and the bias is set to a constant value of 0 if'm' is an instance of 'nn.Conv2d'. The weight is initialized using the Xavier normal initialization approach with a gain calculated using the ReLU nonlinearity and the bias set to a constant value of zero if m is an instance of `nn.Linear`, in which case the gain is calculated.

In order to ensure efficient learning, the model parameters were initialized using the proper methods. Examples include initializing the weights of linear layers using Xavier normal initialization and the weights of convolutional layers using Kaiming normal initialization. Biases were set to 0 at initialization.

By counting the number of parameters using the utility function, one may easily evaluate model complexity. Model analysis, issue fixing, and contrasting various models can all benefit from it.

E. Training

Using the training dataset, the models were trained. The models were applied to the training data in mini-batches, the training data was iterated over, the images were applied to the models, the loss was calculated using the cross-entropy loss function, and the model parameters were optimized using the Adam optimizer. The training loss and accuracy of the models were recorded as they underwent repeated epochs of training.

To maintain track of the total loss and accuracy inside the current epoch, initialize the variables epoch_loss and epoch_acc. The model is placed in training mode, enabling features like dropout, by calling model.train().
The loss and accuracy numbers are then totaled for the duration of the current era.

In contrast to train (), which is used for training, evaluate () is used for evaluation. As it is used to evaluate the model's performance on a validation or test dataset, it does not execute backpropagation or parameter adjustments.

Overall, these features offer a foundation for modelling training and assessment. They can be applied in an iterative training loop to train the model on data batches and assess its performance at each epoch.

F. Validation

The models were evaluated using the validation dataset after each training epoch. It was necessary to analyze the images using trained models, determine the validation loss and accuracy, and monitor the models' performance using unlabeled data. The most effective model identified by the validation loss was saved for future usage.

G. Validation

The validation dataset was used to assess the models following each training phase. During the validation procedure, the images were run through the trained models, the validation loss and accuracy were calculated, and the models' performance on untried data was tracked. The top model determined by the validation loss was kept for future usage.

H. Testing

On the testing dataset, the trained models' performance was assessed. The saved best model was loaded, and predictions were obtained by running the model on the test dataset. To determine criteria like accuracy, the predicted labels were compared to the actual labels.

VI. Results And Discussions

A. AlexNet Results

Training Loss: As the epochs go on, the average training loss goes down, showing that the algorithm is picking up new information from the training data. Training Accuracy: Over time, training accuracy increases, peaking at 99.11% in the most recent epoch. This shows that the algorithm is effectively recognizing and categorizing the training data. Validation Loss: The validation loss varies over the course of training, slightly increasing in later epochs. This implies that the algorithm may be excessively adapting to the training data, which would lead to less effective generalization to the validation set. Valid Accuracy: Starting at 96.53%, the validation accuracy fluctuates a little before peaking at 97.05% in the seventh epoch. However, it slightly decreases in the last epoch. Overall, the algorithm achieves good accuracy on the validation set.

The following figures demonstrate how the models performed throughout training. AlexNet ran for a total of 15 epochs from the algorithm below before halting. The loss values obtained using the validation and training data are nearly comparable, but the training accuracy obtained was less than that obtained using the validation data.

<matplotlib.legend.Legend at 0x7fb6d7dafe50>

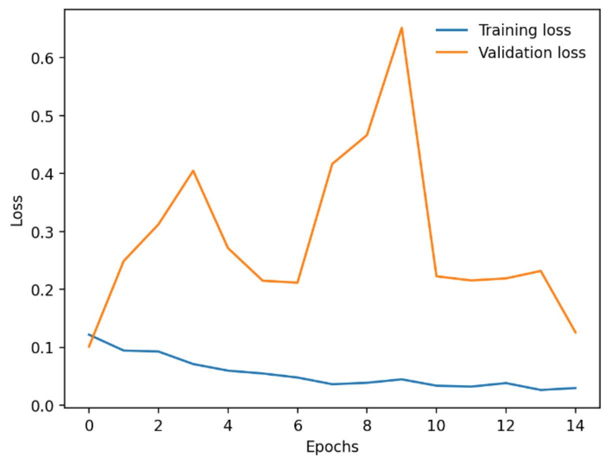

Fig: AlexNet plot of Training loss and Validation loss

<matplotlib.legend.Legend at 0x7fb6d5ac96f0>

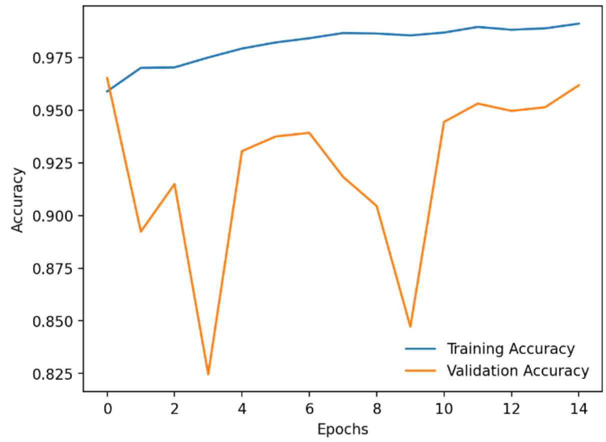

Fig: AlexNet plot of Training Accuracy and Validation Accuracy

B. VGG11 Results

Training Loss: The training loss lowers continuously as the epochs proceed, showing that the algorithm is effectively learning from the training data. Training Accuracy: The training accuracy increases over time, peaking at 99.87% in the most recent epoch. This shows that the algorithm is correctly identifying and learning from the training data. Valid Loss: The validation loss exhibits some variance, but overall, it remains quite consistent across the epochs. Valid Accuracy: The validation accuracy starts at 90.28% and improves over time, reaching a peak of 97.05% in the seventh epoch. Later epochs show persistent high accuracy, demonstrating good generalization to the validation set.

The following figures demonstrate how the models performed throughout training. VGG11 ran for a total of 15 epochs from the algorithm below before halting. The loss values obtained using the validation and training data are nearly comparable, but the training accuracy obtained was less than that obtained using the validation data.

<matplotlib.legend.Legend at 0x7fcca0c62e90>

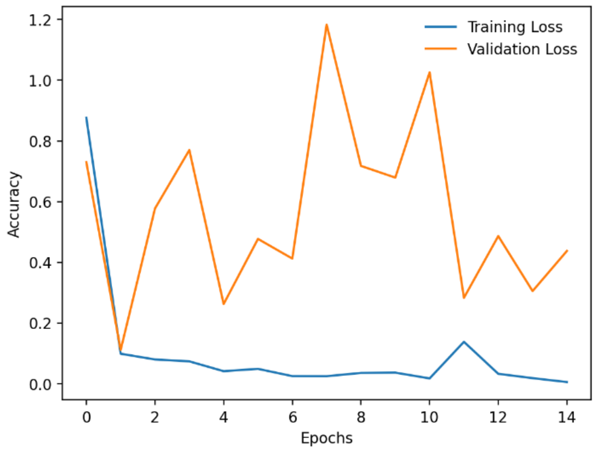

Fig: Plot of VGG11 training and validation loss

<matplotlib.legend.Legend at 0x7fcca0cbfd00>

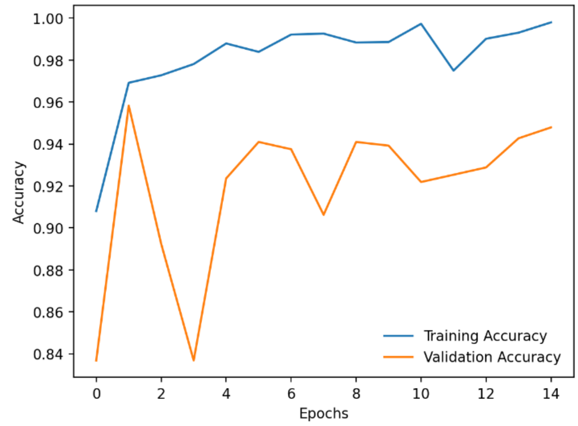

Fig: Plot of VGG11 training and validation accuarcy

Table 1

Architecture	Accuracies		
	Epochs	*Train*	*Validation*
AlexNet	15	98.91%	95.83%
VGG11	15	99.80%	94.79%

<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x7fcca08668f0>

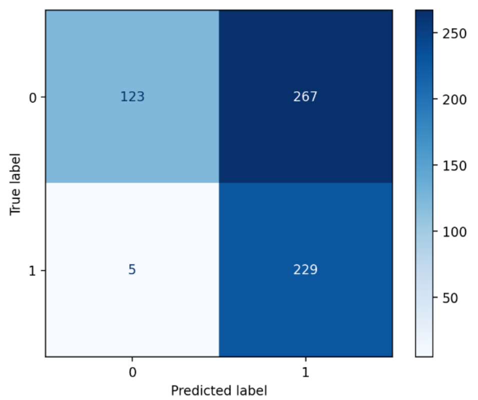

Fig: Confusion Matrix of VGG11

The output of the Confusion Matrix of CNN shows that it managed to predict 123 True Negative cases and 229 True Negatives whereas the number of False Positives and False Negatives are 267 and 5.

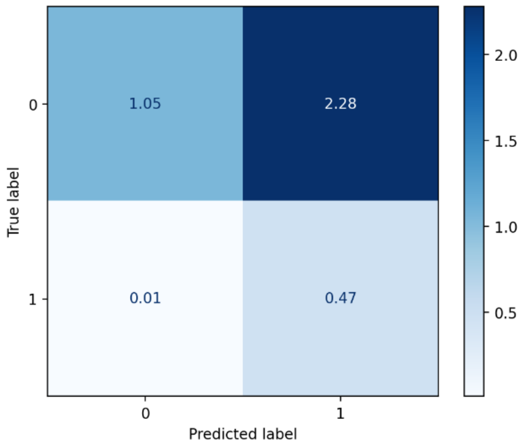

Fig: Confusion Matrix of VGG11

VII. Conclusion And Further Work

When the performance of the two algorithms is compared, VGG11 consistently exceeds AlexNet in terms of accuracy throughout both training and validation. With up to 98.62% on the training set and 96.88% on the validation set, VGG11 obtains greater accuracy rates. AlexNet, on the other hand, consistently achieves 97.59% accuracy on both the training and validation sets.

Comparing the two algorithms, both AlexNet and VGG demonstrate good training accuracies and reasonable validation accuracies. On the validation set, the VGG algorithm, as opposed to AlexNet, generally performs better, achieving higher accuracy and more stable loss values. Therefore, based on these results, it can be inferred that the VGG algorithm outperforms AlexNet for the given task.

Additionally, compared to AlexNet, VGG11 tends to sustain fewer validation losses. This shows that VGG11 has a superior ability to distinguish between pneumonia and healthy cases and generalizes more effectively to unobserved data.

VGG11 is the best algorithm for using chest X-ray pictures to forecast pediatric pneumonia, according to the provided epoch data. It continuously attains higher accuracy rates and exhibits stronger generalization skills. VGG11 is thus advised in this context for additional analysis and prediction tasks.

A. Future Works

In the future, we may research methods for enhancing contrast in photos or other pre-processing activities to increase the image quality. To help the CNN models do better feature extraction, we might also think about segmenting the lung picture prior to classification. The computing cost is also larger than that of the CNN baselines generated in studies in the literature since the proposed ensemble must be trained using three CNN models. By using techniques like snapshot ensembling in the future, we might try to lower the processing needs [6].

VIII. References

[1] O'Grady, K.-A.F. *et al.* (2014) *The radiological diagnosis of pneumonia in children - pneumonia*, *BioMed Central*. Available at: https://pneumonia.biomedcentral.com/articles/10.15172/pneu.2014.5/482 (Accessed: 25 May 2023).

[2] bookshelf, N. (2000) *Pediatric pneumonia - statpearls - NCBI bookshelf*. Available at: https://www.ncbi.nlm.nih.gov/books/NBK536940/ (Accessed: 25 May 2023).

[3] Alsharif, R. and Alquran, H. (2021) *PneumoniaNet: Automated detection and classification of pediatric pneumonia using chest X-ray images and CNN approach*, *MDPI*. Available at: https://www.mdpi.com/2079-9292/10/23/2949 (Accessed: 25 May 2023).

[4] Pneumonia, F. (2023) *FastStats - Pneumonia*, *Centers for Disease Control and Prevention*. Available at: https://www.cdc.gov/nchs/fastats/pneumonia.htm (Accessed: 25 May 2023).

[5] Hwa1, S.K.T., Bade1, A. and Hijazi2, M.H.A. (2020) *IOPscience*, *IOP Conference Series: Materials Science and Engineering*. Available at: https://iopscience.iop.org/article/10.1088/1757-899X/979/1/012016 (Accessed: 26 May 2023).

[6] Kundu, R. *et al.* (2021) *Pneumonia detection in chest X-ray images using an ensemble of Deep Learning Models*, *PloS one*. Available at: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8423280/ (Accessed: 26 May 2023).

[7] Chassagnon, G., Paragios, N. and Revel, M.-P. (2020) 'Deep learning–based approach for automated assessment of interstitial lung disease in systemic sclerosis on CT images', *Radiology: Artificial Intelligence*, 2(4). doi:10.1148/ryai.2020190006.

[8] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[9] Rajpurkar, P. (2017) *CheXNet: Radiologist-level pneumonia detection on chest X-rays with deep learning*, *arXiv.org*. Available at: https://arxiv.org/abs/1711.05225 (Accessed: 26 May 2023).

IV. Project Prototype

Code Available Here
https://colab.research.google.com/drive/1_mDmM5-LPVqY6xH7Xidu9gx0xcgeGHb_?usp=sharing

Deep Learning Project Folder with zip file and collab file
https://drive.google.com/drive/folders/1XJLfPQYbVD6vOOKPO0rLJoCSLAVpvv_T?usp=sharing