North East University Bangladesh

Department of Computer Science and Engineering

Using Machine Learning Models to Classify Pneumonia from X-ray Images

By

Md. Abdul Mutalib	Shahriar Hussain	Kopil Das
Reg. No: 190303020001	Reg. No: 190303020008	Reg. No: 190303020009
BSc(Engg) in CSE	BSc(Engg) in CSE	BSc(Engg) in CSE
4^{th} year 2^{nd} semester	4^{th} year 2^{nd} semester	4^{th} year 2^{nd} semester

Supervised By

Tasnim Zahan

Assistant Professor

Department of Computer Science and Engineering

North East University Bangladesh

16^{th} July, 2023

Using Machine Learning Models to Classify Pneumonia from X-ray Images

A Thesis submitted to the Department of Computer Science and Engineering, North East University Bangladesh, in partial fulfillment of the requirements for the degree of Bachelor of Science in Computer Science and Engineering

By

Md. Abdul Mutalib
Reg. No: 190303020001
BSc(Engg) in CSE
4th year 2nd semester

Shahriar Hussain
Reg. No: 190303020008
BSc(Engg) in CSE
4th year 2nd semester

Kopil Das
Reg. No: 190303020009
BSc(Engg) in CSE
4th year 2nd semester

Supervised By

Tasnim Zahan
Assistant Professor
Department of Computer Science and Engineering
North East University Bangladesh

16th July, 2023

Recommendation Letter from Thesis Supervisor

These Students, *Md. Abdul Mutalib, Shahriar Hussain, Kopil Das,* whose thesis/project entitled *"Using Machine Learning Models to Classify Pneumonia from X-ray Images",* is under my supervision and agrees to submit for examination.

Signature of the Supervisor :

Tasnim Zahan
Assistant Professor
Department of Computer Science and Engineering
North East University Bangladesh

Qualification Form of BSc(Engg) Degree

Student Name : Md. Abdul Mutalib, Shahriar Hussain, Kopil Das
Thesis Title : Using Machine Learning Models to Classify Pneumonia from X-ray Images

This is to certify that the thesis is submitted by the student named above in July, 20023. It is qualified and approved by the following persons and committee.

Head of the Dept.	**Supervisor**
Tasnim Zahan	Tasnim Zahan
Assistant Professor	Assistant Professor
Department of CSE	Department of CSE
North East University Bangladesh	North East University Bangladesh

Abstract

Pneumonia is a significant cause of child mortality in Bangladesh, and the shortage of radiologists exacerbates the problem. To address this, machine learning (ML) techniques can be utilized to assist in pneumonia diagnosis. In this project, we explore the application of different ML implementations for classifying pneumonia from chest X-ray images. We consider a Naive Bayes classifier, a support vector machine (SVM) classifier, and a transfer learning-based classifier using DenseNet161 as the base model. We preprocess the images and extract features using techniques such as Discrete Cosine Transform (DCT), Histogram of Oriented Gradients (HOG), and Local Binary Patterns (LBP). Additionally, we apply data augmentation techniques to enhance the diversity and quantity of the training data. Our results demonstrate the potential of these ML models in accurately classifying pneumonia from chest X-ray images, providing valuable support for radiologists in Bangladesh and other regions facing similar challenges.

Keywords: Pneumonia, Machine Learning, Chest X-ray images, Naive Bayes classifier, Support Vector Machine, Transfer Learning, DenseNet161, Feature Extraction, Data Augmentation

Table of Contents

List of Figures

Chapter 1

INTRODUCTION

Introduction: Pneumonia is a leading cause of death in Bangladesh, accounting for 13% of deaths in children under the age of five. In 2021, an estimated 12,000 children in Bangladesh died from pneumonia. This is a major public health problem, and it is exacerbated by the fact that Bangladesh has a shortage of radiologists which is not enough to meet the needs of the population.

Machine learning (ML) can be used to help diagnose pneumonia. ML algorithms can be trained on a dataset of chest X-ray images to learn to identify the telltale signs of pneumonia. This can help to improve the accuracy of diagnosis, and it can also help to reduce the workload on radiologists.

In this project, we explore different ML implementations that can be used to classify pneumonia from chest X-ray images. We will be looking at a Naive Bayes classifier, a support vector machine (SVM) classifier, and a transfer learning-based classifier. Our input is in the form of chest X-ray images, and we output a classification of the presence of pneumonia.

Chapter 2

RELATED WORKS

2.1 Image Feature Extraction

Image feature extraction is a crucial step in computer vision and image processing tasks. It involves transforming raw images into a meaningful and compact representation of their visual characteristics. These extracted features capture essential information from the images and can be used for various purposes, such as image classification, object detection, image retrieval, and more.

There are several popular techniques for image feature extraction, and some commonly used methods include:

2.1.1 Discrete Cosine Transform (DCT)

Sub-subsections are numbered as given above and included as a third level of text. Sub-

2.1.2 Histogram Oriented Gradient (HOG)

The Histogram of Oriented Gradients (HOG) feature extraction descriptor has been widely studied and applied in image classification tasks. HOG, introduced by Dalal in 2005, is renowned for its ability to capture shape and edge information in images, making it a popular choice for image classification.

HOG operates by dividing the image into small cells and computing histograms of gradient orientations within each cell. These histograms effectively capture the local patterns and textures in the image, forming the foundation of the HOG descriptor. The selection of HOG parameters plays a crucial role in its performance.

Researchers have extensively explored various aspects of HOG to enhance its effectiveness in image classification. The choice of cell size determines the level of detail captured by the descriptor, with smaller cells capturing finer details at the expense of increased computational complexity. The number of orientation bins determines the level of granularity in the gradient computations, allowing the descriptor to capture different edge orientations accurately.

Furthermore, researchers have investigated gradient computation methods and normalization techniques to improve the robustness and discriminative power of HOG features. These advancements have contributed to the success of HOG in accurately classifying images into different categories.

In the realm of image classification, numerous researchers have contributed to the understanding and advancements of HOG. Notably, the work by Bouchra NASSIH*, Aouatif AMINE, Mohammed NGADI, Nabil HMINA [Add complete reference here] has provided valuable insights into the variations and applications of HOG in various image classification scenarios.

2.1.3 Data Augmentation

We were very concerned about our model's performance, that's why we want to improve generalization and reduce overfitting issue. For this reason, we explore various techniques to enhance the diversity and quantity of image data for training our models. By augmenting the training data, we can expose the model to a wider variety of variations and increase its ability to generalize well to unseen data.

2.1.4 Data Set

Using X-ray scan datasets from Kaggle, we split data into a training, validation and testing set . We had 5856 images, resulting in a respective 5216, 16 and 624 split.

Data augmentation is a method of applying distortion to the original dataset. It is used to increase the size and diversity of a training dataset. It applies various transformations to the existing data samples. The goal of data augmentation is to enhance the model's ability to generalize and improve its performance. In our data augmentation section, we used different transformations like Random flip, random rotation, random translation, gray scaling and random crop etc. Here we explained our transformations briefly.

- Gray scale: Grayscale conversion is one such transformation that can be applied to images. It is a representation of an image where each pixel's intensity is represented by a single value, ranging from 0 (black) to 255 (white). Gray scale conversion takes a color image, which typically consists of three-color channels (red, green, blue), and reduces it to a single channel representation.
- Random flip: Random horizontal flip is one such transformation that can be applied to images. Where an image is flipped horizontally with a certain probability. This

transformation involves reversing the image from left to right, simulating a mirror reflection.

- Random rotation: This technique involves randomly rotating images by a certain angle within a specified range. By randomly rotating images, the model learns to recognize objects and patterns from different angles, making it more versatile in handling images with various orientations.
- Random translation: This transformation involves randomly shifting or translating images in both the horizontal and vertical directions by a certain number of pixels. By randomly shifting images, the model learns to recognize objects and patterns regardless of their specific position.
- Random crop: This transformation involves randomly selecting and cropping a portion of an image. The size of the crop is typically specified as a range or maximum dimensions in pixels.

Here are the examples of our augmented images.

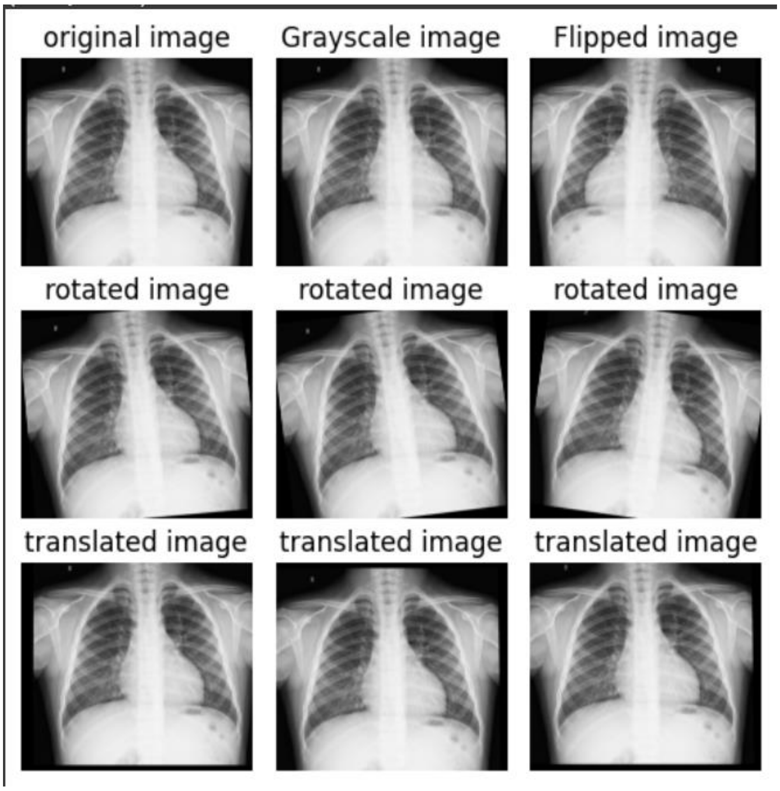

Figure 1: Augmented data for Transfer Learning

Chapter 3

PROPOSED METHOD

3.1 Support Vector Machine (SVM)

The Model: Support Vector Machines (SVM) are a powerful and widely used machine learning algorithm for classification tasks. They belong to the family of supervised learning models and have gained popularity due to their effectiveness and versatility in handling various types of data. SVM can be used for both binary and multi-class classification problems.

One of the key characteristics of SVM is the ability to find an optimal hyperplane that separates different classes in the feature space. This hyperplane is chosen to maximize the margin, which is the distance between the hyperplane and the nearest data points of each class. By maximizing the margin, SVM aims to achieve better generalization and improve the model's ability to classify new, unseen data accurately.

SVM can handle linearly separable data, where a straight line or hyperplane can separate the classes. However, they can also handle non-linearly separable data by using kernel functions. Kernel functions transform the original feature space into a higher-dimensional space, where the data becomes linearly separable. This allows SVM to capture complex relationships and patterns in the data.

How SVM works: Support Vector Machines (SVM) is a classification algorithm that aims to find an optimal line or hyperplane that can separate data points belonging to different classes. The key idea behind SVM is to maximize the margin, which is the distance between the separating line/hyperplane and the nearest data points of each class. This margin provides a safety buffer, reducing the chances of misclassifying new, unseen data.

SVM can handle cases where the data points are perfectly separable by a straight line or hyperplane. However, it is also designed to handle cases where the data is not linearly separable. In such cases, SVM uses a technique called the "kernel trick." The kernel trick transforms the original feature space into a higher-dimensional space, where the data becomes separable by a linear boundary. This allows SVM to capture complex relationships and patterns in the data.

By finding the optimal separating hyperplane, SVM learns the best decision boundary between classes. The points closest to the hyperplane, called support vectors, play a crucial role in defining the boundary. SVM focuses on these support vectors and ignores other data points, making it efficient in high-dimensional spaces.

The choice of the kernel function in SVM is critical. Different kernel functions define different relationships between features, allowing SVM to handle various types of data. Common kernel functions include the linear kernel for linear separability, polynomial kernel for non-linear decision boundaries with higher degrees, and radial basis function (RBF) kernel for non-linear and complex patterns.

In our SVM model, we employed the HOG and LBP feature extraction techniques to feed our data to the SVM model. HOG captures shape and edge information by analyzing local gradient orientations, while LBP characterizes texture by examining pixel intensity comparisons. These features provide meaningful representations of the images, enabling the SVM model to learn discriminative patterns and make accurate class predictions.

Method for SVM: In our SVM model we utilized two feature extraction techniques, namely Histogram of Oriented Gradients (HOG) and Local Binary Patterns (LBP), in combination with the Support Vector Machines (SVM) algorithm.

LBP is a texture analysis technique that focuses on the pixel intensity comparisons within a local neighborhood. It operates by comparing the intensity value of a central pixel with its surrounding neighbors and encoding the results as binary patterns. These binary patterns capture the local texture variations in the image. LBP features are computed by constructing histograms of the different binary patterns occurring within the image. These histograms serve as descriptors that represent the texture information in the image. LBP features are known for their simplicity and effectiveness in texture classification tasks.

Local Binary Patterns (LBP) is a texture analysis technique widely used in computer vision for its simplicity and effectiveness in capturing local texture variations in an image. LBP operates by comparing the intensity value of a central pixel with its surrounding neighbors. This comparison is done by thresholding the intensity values, resulting in binary values of 0 and 1.

To compute LBP features, a local neighborhood is defined around each pixel in the image. The neighborhood can be circular or rectangular and can vary in size. For each pixel, the binary pattern is created by comparing the intensity value of the central pixel with its neighboring pixels. If a neighbor's intensity value is greater or equal to the central pixel's value, a binary value of 1 is assigned; otherwise, a binary value of 0 is assigned.

3.2 Naive Bayes Classifier

The model: The Naive Bayes model was chosen as the initial algorithm for our pneumonia detection model due to its simplicity and ability to provide a baseline accuracy. This allowed us to gain insights into the dataset and establish a starting point for further improvements with more complex models.

How it works: The Naive Bayes model works by making the Naive Bayes assumption, which assumes that the probability of each feature occurring is conditionally independent of all other features given the class. While this assumption is not entirely accurate, it offers a practical simplification that often performs well in practice. The simplification allows a computation for the probability of some input X belong to come class C to be,

$$P(class = features = X) = rj_i\, P(class = features_i = X_i)$$

This expression can be computed much more easily by counting the number of times feature i occurs in class C vs the total number of times feature i occurs at all.

Image Pre-processing: To ensure standardized input for our Naive Bayes classification model, we performed pre-processing steps on our dataset. we resized all images to a fixed size of 100 x 100 pixels, maintaining consistent dimensions across the dataset. By doing so, we ensure that the images are in a uniform format for further processing and analysis. While augmentation data , we converted all the images to grayscale.

Feature Extraction: As the Naive Bayes model is not capable of directly processing raw pixel data like more complex deep learning models, we incorporated effective feature extraction techniques to derive informative features from the image data. We experimented with only Discrete Cosine Transform (DCT).

DCT is commonly employed in image compression, where images are represented as a weighted sum of cosine waves. By calculating the DCT on a sliding square of pixels, typically 8 x 8, local DCT values are obtained. In our case, the 100 x 100-pixel images were divided into a grid, and DCT was computed for each square. This resulted in a grid of DCT values, capturing frequency components of the image. We performed DCT calculations using different window sizes, including 4 x 4, 8 x 8, to experiment and identify the optimal size for our model.

3.3 Transfer Learning

Transfer learning is a technique where we can take a pre-trained model. Typically trained on a large-scale dataset, and adapting it to a new, smaller dataset or a different task (target task). In our project our target task is identifying whether a chest x-ray image shows signs of pneumonia or normal. When we used transfer learning to solve our target task, we selected a pre-trained model as our base model. There are may two possible approaches to using knowledge from the pre-trained model. The first way is to freeze few layers of the pre-trained model and train other layers on our new dataset for the new task. The second way is to make a new model, but also take out some features from the layers in the pre-trained model and use them in a newly created model. For our target task we have selected the first approach. We will discuss about our approach in our classification layer section.

3.3.1 Base model

We have selected DenseNet161 as our pretrained model. DenseNet161 is a deep learning neural network model used for image classification tasks. It is based on a DenseNet architecture, which means that it has densely connected layers where each layer receives input from all preceding layers. DenseNet architecture have different model such as DenseNet121, DenseNet161, DenseNet169 and so on. DenseNet161 has 161 layers. It was introduced in a research paper in 2017 and has achieved state-of-the-art results on several image classification tasks, including the ImageNet dataset. Its architecture includes convolutional layers, pooling layers, and fully connected layers. It also uses techniques such as batch normalization and dropout to improve its performance. Now we are going to discuss briefly different types of layers in DenseNet architecture.

- Convolutional layers: Convolutional layers are the fundamental building blocks of DenseNet-161. They perform convolution operations on the input data to extract features and capture spatial patterns. In DenseNet-161, multiple convolutional layers are stacked together to form dense blocks.
- Bottleneck layers: Within each dense block, bottleneck layers are employed. A bottleneck layer consists of a batch normalization layer, a 1x1 convolutional layer, and a rectified linear unit (ReLU) activation function. The 1x1 convolutional layer reduces the number of input channels, often referred to as the bottleneck layer, significantly reducing computational complexity.

- Pooling layers: After the convolutional layers, pooling layers are inserted to down-sample the input image and reduce the size and computational complexity. Two common types of pooling layers are:
  - Max Pooling: it selects the highest pixel value from the window of the image currently covered by the feature detector which helps to identify the salient features of the image.
  - Average Pooling: it calculates the average value from the window of the image currently covered by the feature detector which identifies the full extent of the image.

There are few other terms which are the important concept in DenseNet.

- Growth rate: growth rate determines the number of feature maps are generated into individual layers inside dense blocks.
- Dense connectivity: In DenseNet-161, layers within a dense block are densely connected to each other in a feed-forward manner. Each layer takes as input the feature maps from all preceding layers within the same dense block.
- Composite functions: Composite functions refer to mathematical functions that are formed by combining two or more individual functions. In a composite function, the output of one function becomes the input for another function, creating a chain of transformations. So we can define as a composite function of three consecutive operations they are batch normalization, ReLU and a 3x3 convolution (Conv).

3.3.2 Classification layers

The classification layer in DenseNet architecture is the final layer of the neural network that is responsible for performing the task of image classification. It takes the features by the preceding layers and maps them to the specific output classes. As our dataset is not very large so we have decided to freeze the whole pre-trained model layers and simply add a linear classifier at the end that take the output of DenseNet161 as its input and produces outputs two classes (Pneumonia or Normal). Our DenseNet161 architecture is shown below.

$$\text{DenseNet161} \rightarrow \text{Affine} \rightarrow \text{SoftMax}$$

Chapter 4

RESULTS AND DISCUSSION

4.1 Results

4.1.1 Support Vector Machine

The results obtained from the HOG method show that the overall accuracy achieved is 76.44%. When considering different SVM kernels, it is observed that the poly kernel performs the best with a test accuracy of 78.53%, followed by the linear kernel with a test accuracy of 77.24%. The rbf and sigmoid kernels achieve test accuracies of 76.44% and 74.36% respectively. These results suggest that the choice of SVM kernel has an impact on the classification performance of the HOG method, with the poly and linear kernels yielding the highest accuracies.

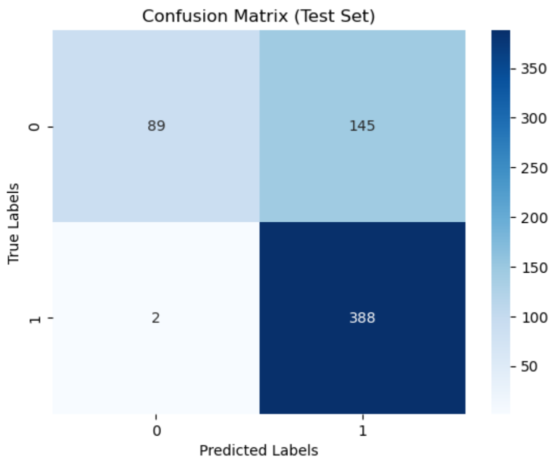

Figure 2 Confusion matrix for HOG method

On the other hand, the results obtained from the LBP method show an overall accuracy of 72.12%. Among the different SVM kernels, the poly kernel achieves the highest test accuracy of 73.24%, followed by the RBF kernel with a test accuracy of 72.12%. The linear and sigmoid kernels yield lower test accuracies of 62.50% and 61.70% respectively. These results indicate that the choice of SVM kernel also affects the classification performance of the LBP method, with the poly and RBF kernels performing better compared to the linear and sigmoid kernels.

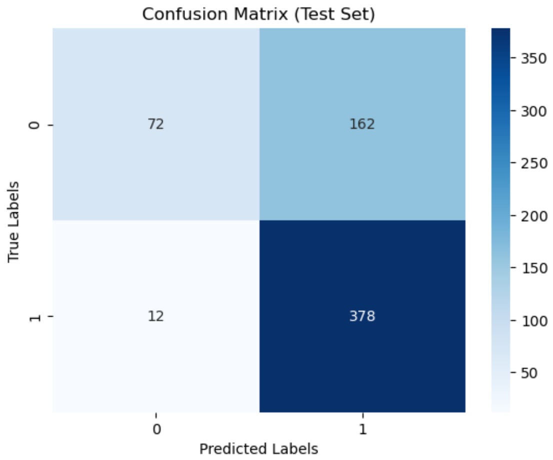

Figure 3 Confusion Matrix for LBP method

In conclusion, the SVM classifier achieved an accuracy of 76.44% with the HOG feature extraction method and an accuracy of 72.12% with the LBP feature extraction method. The poly kernel demonstrated the highest accuracy among the tested SVM kernels, with a test accuracy of 78.53% for HOG and 73.24% for LBP. The linear and RBF kernels also yielded reasonably good accuracies, while the sigmoid kernel showed lower performance. These results suggest that the combination of the SVM classifier and the HOG feature extraction method is more effective in accurately classifying the given dataset compared to the LBP method.

4.1.2 Transfer Learning

As our pretrained mode is very large, when we were training it on our dataset it takes around four hours. Whatever our model consistently provided better performance, with an overall accuracy of 88.94% across our test data set. It also well performed to classifying normal and pneumonia. In this case, it was showing less overfitting signs, so we decided to use 20% dropout regularization to reduce overfitting issue. Then we got overall accuracy 87.50% across our test data set. Since we have used dropout regularization our model performed well generalized. The confusion matrix for both is shown below.

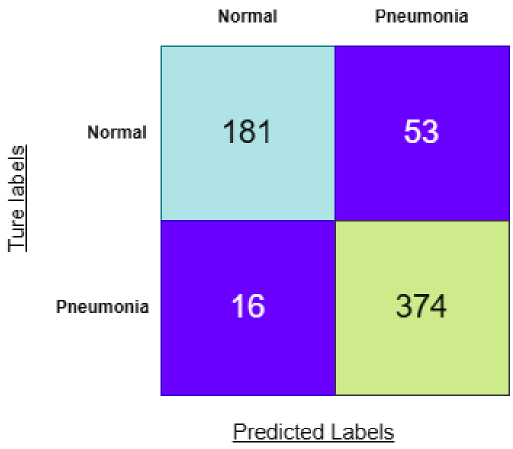

Figure 4 Confusion matrix of Transfer Learning without droput regularization

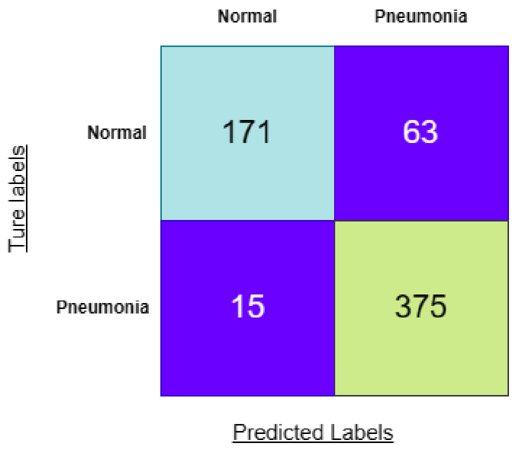

Figure 5 Confusion matrix of Transfer Learning with droput regularization

4.2 Table Format

Chapter 5

CONCLUSION

Place your conclusions here.

References

[1] Caselles V., Lisani L., Morel M., and Sapiro G., “Shape Preserving Local Contrast Enhancement,” *in Proceedings of International Conference Image Processing*, pp. 314 317, 1997.

[2] Chen, S.D., Ramli, A.R., “Contrast enhancement using recursive mean-separate histogram equalization for scalable brightness preservation”, *IEEE Trans. on Consumer Electronics*, vol. 49, no. 4, pp. 1301–1309, 2003.

[3] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Prentice Hall, Upper Saddle River, New Jersey, EUA, 3nd edition, January 2008.