SAKARYA UNIVERSITY JOURNAL OF COMPUTER AND INFORMATION SCIENCES
VOL. 5, NO. 1, APRIL 2022
DOI: 10.35377/saucis.05.01.1019187

Research Article

Using a Convolutional Neural Network as Feature Extractor for Different Machine Learning Classifiers to Diagnose Pneumonia

Enes AYAN[1]
[1]Corresponding Author; Kirikkale University, Department of Computer Engineering;75450, Yahsihan/ Kirikakale, Turkey; enesayan@kku.edu.tr

Received 04 November 2021; Revised 14 February 2022; Accepted 01 March 2022; Published online 30 April 2022

Abstract

Pneumonia is a general public health problem. It is an important risk factor, especially for children under 5 years old and people aged 65 and older. Fortunately, it is a treatable disease when diagnosed in the early phase. The most common diagnostic method known for the disease is chest X-Rays. However, the disease can be confused with different disorders in the lungs or its owsn variants by experts. Therefore, computer-aided diagnostic systems are necessary to provide a second opinion to experts. Convolutional neural networks are a subfield of deep learning and they have demonstrated success in solving many medical problems. In this paper, Xception which is a convolutional neural network was trained with the transfer learning method to detect viral pneumonia, normal cases, and bacterial pneumonia in chest X-Rays. Then, five different machine learning classification algorithms were trained with the features obtained by the trained convolutional neural network. The classification performances of the algorithms were compared. According to the experimental results, Xception achieved the best classification performance with an accuracy of 89.74%. On the other hand, SVM achieved the closest classification performance to the convolutional neural network model with 89.58% accuracy.

Keywords: Artificial Intelligence, Deep Learning, Pneumonia, Convolutional Neural Networks, Machine Learning

1. Introduction

Pneumonia is an infection that appears in the lungs caused by viruses or bacteria. It presents a great threat, especially for children. Annually, 1.4 million children die because of pneumonia. 18% of these children are under 5 years old [1]. Pneumonia also threatens people aged 65 and older and those with chronic lung diseases [2]. Every year an average of 450 million people contracted pneumonia worldwide [3]. Fortunately, through early diagnosis and suitable medication pneumonia can be treated before it becomes deadly [4]. The most common method utilized for the diagnosis of pneumonia is chest X-Ray images [5]. But pneumonia could be confused with other diseases or harmless abnormalities in chest X-Rays. Because of diagnostic errors, the patient's status gets may worse, it may even result in death [6]. The diagnostic accuracy is dependent on the experience and attention of the radiologist. As a result, diagnostic decisions could be subjective and error-prone. On the other hand, training a radiologist with enough experience is required a long time and costs [7]. Also, it is very difficult to come across a radiologist in rural areas of low-income countries. Therefore, computer-aided diagnosis (CAD) systems are necessary to assist radiologists to diagnose pneumonia in chest X-Rays [8]. Artificial intelligence is a popular research area in terms of solving medical problems. Recently, Convolutional Neural Networks (CNNs) is a subfield of deep learning that has become pretty popular in computer vision tasks. CNNs are special deep artificial neural networks that have been designed inspiring by the mammalian visual cortex [9]. Some applications of CNNs in different computer vision problems are object classification, instance segmentation, and object localization [10]. CNNs also have been achieved promising results in solving medical problems using medical image data (e.g., breast cancer detection [11], brain tumor segmentation [12], skin cancer detection [13]).

The diagnosis of pneumonia in chest X-Rays using CNNs is a popular research area. Most of the studies in the literature focus on re-training of pre-trained CNN models with appropriate transfer learning and fine-tuning strategies. Recent studies have generally focused on only distinguishing between normal and pneumonia cases. Although bacterial and viral types of pneumonia are similar, their treatments are different. There are few studies that classify cases into viral pneumonia bacterial pneumonia and normal cases. The motivation of this study is utilizing a CNN model to be a feature extractor for different machine learning algorithms and comparing their pneumonia classification performance in chest X-Ray images. For this purpose, the Xception CNN model was trained using convenient transfer learning and fine-tuning strategies to classify chest X-Rays into three different classes (normal, viral pneumonia, and bacterial pneumonia). After the training process, the convolutional part of the CNN model was separated from fully connected layers and was used as a feature extractor for various machine learning classifiers. The provided features by the convolutional layers were used to train K-Nearest Neighbor (KNN), Decision Trees (DT), Support Vector Machines (SVM), Logistic Regression (LR), Naive Bayes (NB), and Fully Connected Neural Networks (FC). The trained algorithms were evaluated on test data and their classification performances were compared.

The rest of the paper is organized as follows: Section 2 presents related works. Section 3 introduces the dataset, materials and methods. Section 4 includes experimental results. Section 5 provides a brief discussion. The last section contains the conclusions of the study.

2. Related Works

Thanks to open access datasets, many studies have been carried out to diagnose pneumonia in chest X-Rays using CNNs. Some studies which are using the same dataset as in this study are following. Rajpurkar et al. [14] proposed a 121-layer CNN (CheXNet) model for pneumonia diagnosis. The proposed CNN model in the study was trained with 100000 chest X-Rays, to classify 14 different disorders in the lungs, including pneumonia. According to the test results, the proposed model achieved better classification performance than the radiologists in terms of detecting pneumonia. Kermany et al. [15] trained a pre-trained CNN model utilizing the transfer learning strategy to diagnose pneumonia in chest X-Rays. Liang et al. [16] designed a CNN model via residual connections and dilated convolutions to diagnose pneumonia. They also focused on the effect of transfer learning in pneumonia classification from chest X-Rays. Chounhan et al. [17] trained five different pre-trained CNN via suitable transfer learning strategies for diagnosing pneumonia. Then, they proposed an ensemble method, combining the estimates of the CNN models. Gu et al. [18] diagnosed bacterial and viral pneumonia using a method that comprises two stages. In the first stage, they segmented the lungs with a Fully Convolutional Network (FCN) model. In the second stage, they used a Deep Convolutional Neural Network (DCNN) to detect pneumonia from segmented lungs.

Mittal et al. [19] accomplished the diagnosis of pneumonia in chest X-Rays with the Capsule Network (CapsNet) which is a CNN model including multilayer capsules. Prayog et al. [20] utilized a siamese convolutional network (SCN) to classify chest X-Ray images into 3 classes (normal, viral pneumonia, bacterial pneumonia). Rahman et al. [21] have trained four different pre-trained CNN models (AlexNet, ResNet18, DenseNet201, and SequeezeNet) by transfer learning method to classify pneumonia in chest X-Ray images. Hasimi et al. [22] proposed a weighted CNN classifier using various models: ResNet18, Xception, Inception V3, DenseNet 121, and MobileNet V3 to diagnose pneumonia. Mahmud et al. [23] designed a CNN model named as CovXNet. They utilized depthwise convolution in varying dilation rates in the proposed architecture to differentiate Covid19 from pneumonia. Asnaoui [24] compared single and ensemble learning CNN models (InceptionResNet V2, ResNet 50, MobileNet V2) classification performances of pneumonia and Covid19. Darici et al. [25] used a CNN voting ensemble methodology to classify chest X-Rays into three classes.

Most of the studies in the literature focus on training a pre-trained CNN model for diagnosing pneumonia in chest X-rays. But determining fully connected layer size and neuron number in the classification part of CNN models is complicated and requires experience. The main advantage of our work is to show a CNN model can be used as a feature extractor to simple machine learning algorithms. Instead of using hand-crafted features, a pre-trained CNN model can be used easily as a feature extractor to machine learning algorithms and can be achieved successful results.

3. Materials and methods

In this study, our main idea is to compare the classification performance of a CNN model's fully connected layers with different machine learning classifiers. The proposed methodology is figured out in Figure 1. First of all, we trained a pre-trained Xception model using transfer learning and fine- tuning methods to classify pneumonia in chest X-Rays. Then, the convolutional part of the CNN model was separated from fully connected layers and was used as a feature extractor for various machine learning classifiers. The provided features by the convolutional layers of the model were used to train DT, SVM, LR, NB, KNN. In the final step, the trained machine learning algorithms were evaluated on test data and their individual classification performances were compared.

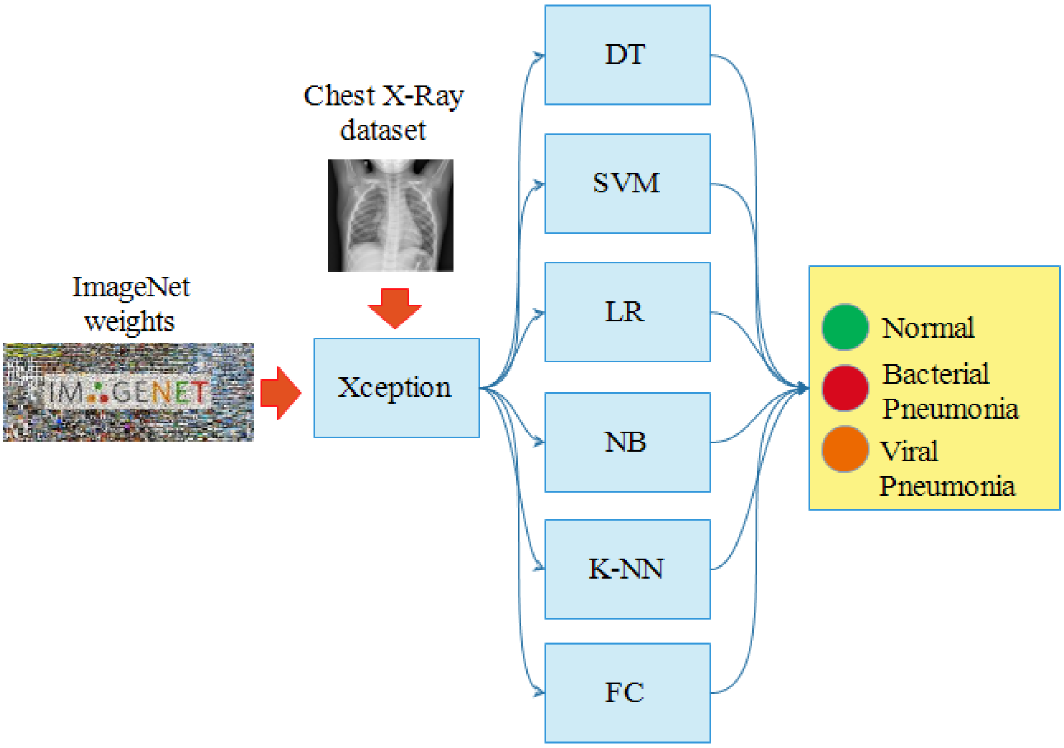

Figure 1 A semantic representation of our methodology

3.1 Dataset

The dataset used in the study was collected from Guangzhou Women and Children's Medical Center and contains chest X-Rays of children under the age of five [26]. The dataset includes a total of 5856 RGB chest X-Rays including bacterial pneumonia, normal, and viral pneumonia cases. The dataset is split into the train (5232) and test (624) sets by its creators. We used a hold-out validation strategy, our test and validation sets are the same. But, in the validation process, we applied online data augmentation to the validation set. The class distribution of the dataset is given in Table 1. In addition, some sample images in different classes from the dataset are given in Figure 2. We resized all images to (299x299x3) to match the default input size of Xception and normalized in the range of [0-1].

Table 1 Distribution of dataset in terms of classes, train validation, and test

Class	**Train**	**Validation**	**Test**
Normal	1349	234	234
Viral Pneumonia	1345	148	148
Bacterial Pneumonia	2538	242	242
Total	5232	624	624

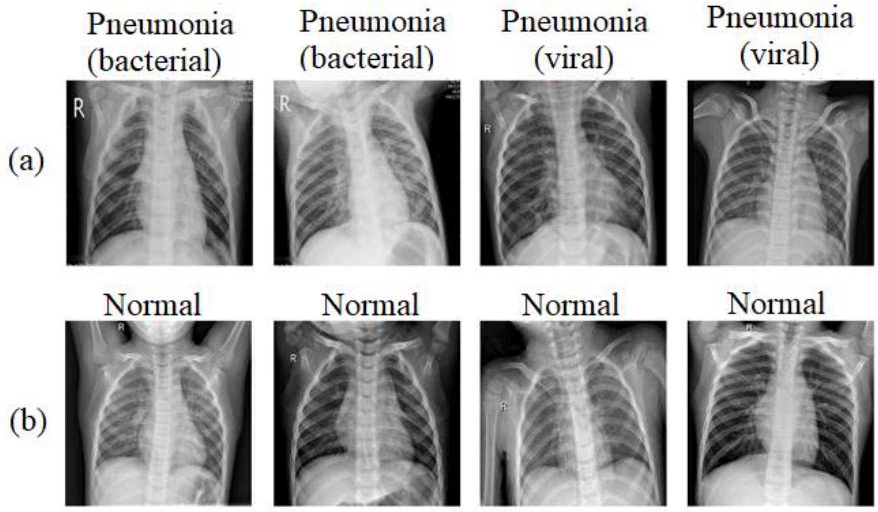

Figure 2 Case samples from the dataset

3.2 Convolutional neural networks

CNNs use a backpropagation algorithm to learn features automatically from data [27]. A CNN architecture consists of four basic components. These are convolutional layers, activation function, pooling layers, and fully connected layers. In the convolution layer, filters of a specified given number and size (3x3, 5x5, 7x7), scan the image in order to reveal meaningful features. After the convolutional layer, an activation function is applied to the obtained features. Activation functions help the model to learn more complex patterns. The Rectified Linear Unit (ReLU) is the most preferred activation function in CNN models. The pooling layers are used to reduce the size of feature maps by retaining the most important information. They also help to reduce calculation costs. The obtained feature maps after stacks of convolution and pooling layers are used in fully connected layers (FC) as the final classification output of a CNN model. In this study, we utilized Xception CNN architecture which was proposed by F. Chollet in 2017 [28]. Xception model was inspired by Inception V3 architecture. The model consists of a linear stack of reverse order depthwise separable convolutions and residual connections. Depthwise separable convolution operation consists of a depthwise convolution followed by a pointwise convolution. Recently, it is mostly preferred convolution type instead of the conventional convolution operation. Depthwise separable convolution is more efficient in terms of computations cost according to conventional convolution operation. It also has fewer parameters to adjust as compared to standard convolution, which helps to reduce overfitting. In Xception architecture depthwise separable convolution operation was modified. In the modified version, pointwise convolution was followed by depthwise convolution. This modification has a similarity with the inception module in Inception V3 architecture. The Xception architecture is composed of 14 modules and 36 convolutional layers in total. All modules have residual connections, except for the first and last modules. The last module is terminated by global average pooling instead of a flatten layer to preserve spatial features of feature maps. In this study, two FC layers with 512 neurons were added after the global average pooling layer to Xception architecture. The ReLU activation function was used in these FC layers. We were determined the FC layer number and their neuron size trying different configurations based on the performance of validation and train data. The output of the fine-tuned Xception model was arranged as 3 neurons to classify bacterial pneumonia, normal and viral pneumonia. In order to avoid overfitting and covariate shift problems, dropout, L2 regularization, and batch normalization methods were used after every FC layer (see Figure 3).

3.3 Transfer learning and data augmentation

Transfer learning is the use of knowledge and experience obtained to solve a problem in solving another similar problem [29]. Few researchers have been training a CNN model from scratch in recent years. They use weights of pre-trained models that were trained on a large dataset such as ImageNet [30]. Pre-trained CNN models can be used directly or integrated into a new model to solve new computer vision problems. In this way, the training process is reduced. There are some tricks by using pre-trained model weights. The most important of criteria is the similarity between the new dataset and the large dataset in which the model was trained previously. This similarity plays a critical role in determining the number of layers that will not participate in training. The initial convolutional layers learn data-independent general features such as edges, corners, simple textures. Through further layers, more complex features emerge such as more complex patterns and textures, objects, and parts of objects. For this reason, while training a CNN model with a new dataset, initial layers may not participate in training according to the data similarity between the new dataset and the large dataset which the model was trained previously. Consequently, a predetermined number of layer weights doesn't update. This procedure is called fine-tuning. Fine-tuning has a positive effect on reducing training time and improving classification performance [31]. In this study, we fine-tuned Xception architecture. We trained the fine-tuned model 10 times over 25 epochs. In the first training procedure, 10% percent of the model was frozen then we increased the freezing rate 10% in the next trainings. The best results were achieved at a 60% freeze rate. The first 60 layers of the model were frozen and didn't train. This number corresponds to 60% of the whole CNN model (see Figure 3). The total parameters size of fine-tuned Xception are 22,178,859 and we trained 14,021,627 of the total. The classification performance of CNNs is directly proportional to the amount of training data [32]. However, it is not possible to obtain sufficient data for every medical problem. Data augmentation is a commonly used method for increasing the number of samples in a dataset. It improves the model's generalization capability, prevents overfitting, and increases the model accuracy [33]. We used online (training time) data augmentation in this study. The train and validation datasets were augmented using various image processing methods such as zooming, rotation, shearing, width shift, horizontal flipping, height shift, and rescaling. Detailed parameters of the image augmentation methods are given in Table 2.

Table 2 Image augmentation parameters

Hyperparameter	**Value**
Zoom range	0.2
Shear range,	0.2
Rotation range	40
Rescale	1./255- [0,1]
Width shift	0.2
Height shift	0.2
Fill mode	Nearest
Horizontal flip	True

3.4 Experimental setup

The fine-tuned Xception model was retrained over 25 epochs with 32 batch sizes. We used the softmax activation function in the output layer. The categorical cross-entropy loss function was used to calculate model error and Adam optimization algorithm was preferred to minimize the loss function. Python programing language and Keras deep learning framework was used to implement the proposed comparison. All trials have been carried out on a computer with a 1080 Ti graphics card and Ubuntu operating system. The fine-tuned CNN model's accuracy and loss graphs are shown in Figure. 4. The gap between the training accuracy, validation accuracy, and training loss, validation loss graphs indicate an overfitting problem. It is clearly seen in Figure 4 the gap is an acceptable extent and there is not any overfitting sign. The fluctuation in the validation accuracy and validation loss graphs is due to the dropout layers. We determined the model hyperparameters batch size, epochs size, learning rate, regularization coefficient by trying various values based on the performance of validation and train data. The hyperparameters of the fine-tuned model are given in Table 3. The training and testing times of all algorithms are given in Table 4.

Table 3 Fine-tuned Xception model parameters

Hyperparameter	Value
Dropout	0.5
Learning rate	0.001
L2 regularization	0.001
Adam beta 1	0.9
Adam beta 2	0.999
Epochs	25

Table 4. The training and testing times of algorithms in seconds.

Algorithm	**Training Time**	**Testing Time**
Xception	1325	11
KNN	6.46	4.59
SVM	23	1.43
NB	0.22	0.022
DT	5.95	0.014
LR	1.35	0.025

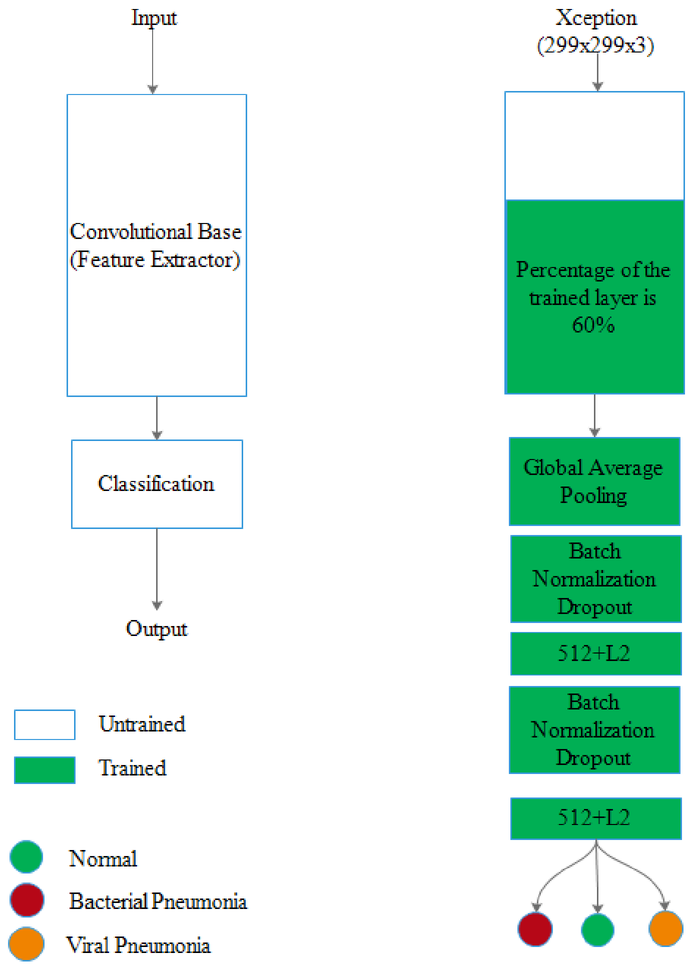

Figure 3 Fine-tuned Xception model

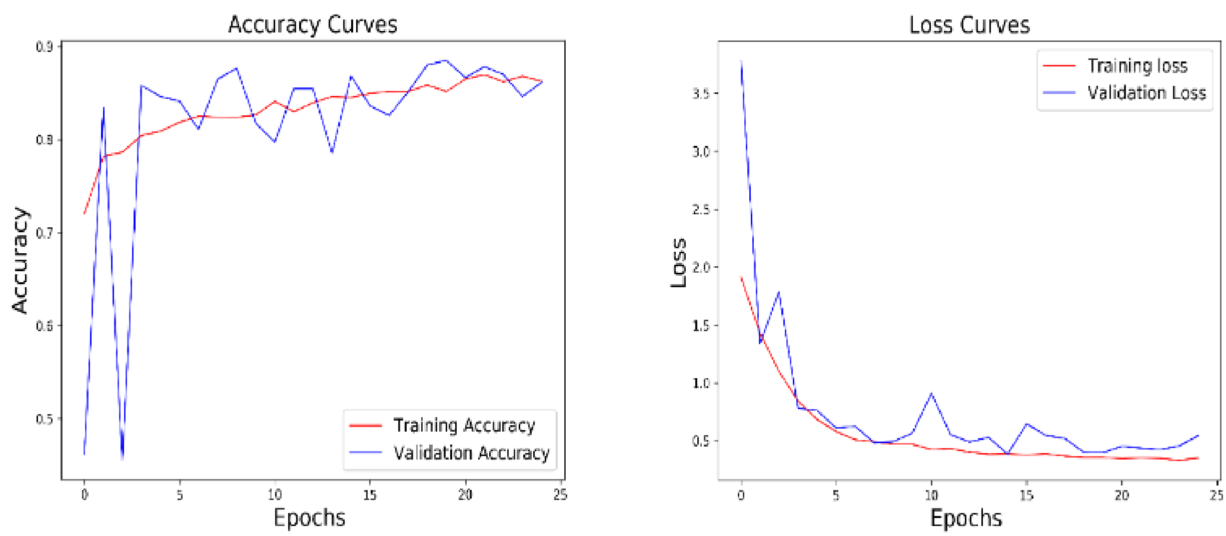

Figure 4 Accuracy and loss graphs of fine-tuned Xception model

3.5 Machine learning classifiers

Hand-crafted features have been used commonly in computer vision problems such as image classification. They are determined by human experts for the related classification problem and the determined features are extracted from the raw image data by using various image processing methods. Then the classical machine learning algorithms use hand-crafted features for solving related classification problems. However, there are some drawbacks of hand-crafted features which affect the performance of the classifier such as inexperienced experts, the unsuccess of image processing methods in feature extraction. Therefore, the generalization ability of machine learning algorithms decreases in different datasets of the same problem. Hand-crafted features are started to be less preferred after deep learning models (CNNs), which self-learn models the necessary features to solve the classification problem from raw image data. So, in this study, we used convolutional layers of the fine-tuned Xception model as a feature extractor for machine learning classifiers instead of hand-crafted features. After the global average pooling layer, the obtained 2048 features were used to train classical machine learning methods. We compared the classification performance of fine-tuned CNN model with SVM [34], KNN [34], DT [34], NB [34], LR [34] algorithms. We utilized the popular machine learning framework scikit-learn. The detailed information about used algorithms in the study is following.

***Support Vector Machines**:* The SVM is a popular supervised machine learning algorithm used in classification and regression problems. It can be used to solve linear or nonlinear problems The main idea of SVM is to find an optimal line or hyperplane which separates the data into classes. The algorithm utilizes support vectors and margins to find optimal line or hyperplane between the classes. It is not fast compared to other machine learning algorithms but it has more accurate and is less prone to overfitting [34].

We tried SVM with different kernels such as linear, radial basis, sigmoid, and polynomial functions. But best result was achieved by the sigmoid function. The C and gamma parameters were determined as 1.0 and 0.0053 respectively.

K-Nearest Neighbor: The KNN is one of the simplest machine learning algorithms which is based on the similarity between the new data points and their nearest neighbors. It is a non-parametric algorithm and can be used for classification and regression problems. The K value represents the nearest neighbor number of the new data point. It is also known as a lazy algorithm because it doesn't learn from data at the training stage, it stores the train data and uses them at the test stage [34]. In this study, we determined the K value (72) by taking the square root of the training data.

Naive Bayes: NB is a classification algorithm based on Bayes' Theorem. So, its predictions are based on the conditional probabilities of data. The algorithm assumes every feature in the dataset is independent of other features and makes predictions on this assumption. In real-world problems, the features generally depend on each other so this is one of the drawbacks of the NB algorithm [34]. There are different types of NB algorithm. We utilized the Gaussian Naïve Bayes algorithm in this study.

Decision Tree: DT is a tree-based supervised algorithm that can be used for classification or regression problems. In a decision tree, internal nodes namely decision nodes represent features, branches represent decision rules and leaf nodes represent final decisions. There are different types of DTs such as CART, ID3, C4.5 according to the used homogeneity criterion (Gini index, Entropy) [34]. We preferred CART DT algorithm in this study.

Logistic Regression: LR is a supervised classification algorithm. It is a transformed version of linear regression by using the sigmoid function and cross-entropy loss function. But it uses for classification problems and gives probabilistic outcomes between 0-1. There are different types of logistic regression such as binomial, multinomial, and ordinal [34]. In this study, we used a multinomial version of LR. We used default parameters NB, DT, and LR in scikit learn framework.

3.6 Performance Metrics

The performances of all algorithms were validated by accuracy (Acc.), recall (sensitivity), specificity (Spe.), precision (Pre.), F1 score, average (Avg.) recall(sensitivity), average specificity, average precision, and average F1 score metrics. The formulas of all metrics are detailed below. In binary classification problems, TP indicates the number of positive cases that are identified correctly by the classifier. TN indicates the number of negative cases that are identified correctly by the classifier. FP indicates the number of negative cases that are identified incorrectly positive by the classifier. FN indicates the number of positive cases that are identified incorrectly negative by the classifier. However, in multi-class problems, TP, TN, FP, and FN are calculated considering each class to other classes. Therefore, all metrics are calculated by class-based.

$$Accuracy = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}} \tag{1}$$

$$Recall - (Sensitivity) = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{2}$$

$$Specificity = \frac{\mathrm{TN}}{\mathrm{TN} + \mathrm{FP}} \tag{3}$$

$$Precision = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \tag{4}$$

$$F1 - Score = 2x \frac{(\mathrm{PrecisionxRecall})}{(\mathrm{Precision} + \mathrm{Recall})} \tag{5}$$

4. Results

We evaluated all algorithms with test data (624). Table 5 indicates the classification results of algorithms. We also calculated the confusion matrices of each algorithm (see Figure 5).

Table 5 Classification results (%) of different machine learning algorithms

Method	**Acc.**	**Avg. Pre.**	**Avg. Spe.**	**Avg. Recall**	**Avg. F1 score**
DT	71.63	74.64	86.20	70.93	69.94
SVM	89.58	**89.59**	94.69	87.71	88.35
LR	77.24	79.48	89.28	77.25	75.51
NB	81.09	82.34	91.25	81.16	79.64
KNN	82.05	82.65	91.41	81.84	80.62
Xception	**89.74**	85.95	**94.92**	**89.74**	**89.72**

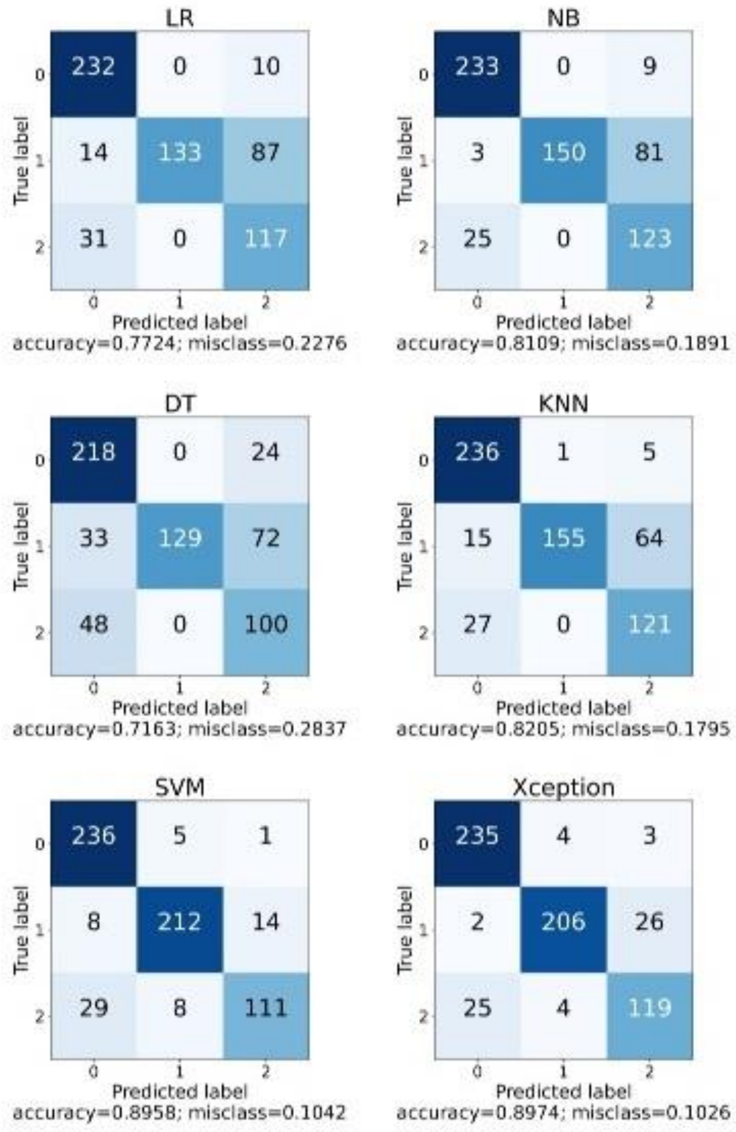

Figure 5 Confusion matrices of algorithms

5. Discussion

In this work, we compared the classification performances of classical machine learning algorithms with a fine-tuned CNN model. The five machine learning algorithms have been trained with the 2048 features extracted by the fine-tuned CNN model. The chest X-Ray images were classified as bacterial pneumonia (0), normal (1) and, viral pneumonia (2). According to Table 5, the fine-tuned CNN model is more successful than other classifiers in terms of all metrics. On the other hand, when the confusion matrices given in Figure 5 are examined, it is seen that the SVM and KNN algorithm is more successful than the others in detecting bacterial pneumonia. While the SVM was more successful in the classification of normal images, NB was outperformed other algorithms in the detection of viral pneumonia. When the assumption of independence of features, NB classifier performs better compared to other models, it needs less training data [34]. There are fewer viral pneumonia cases in the training data. Among the five machine learning algorithms, SVM is the most successful algorithm with an accuracy rate of 89.58%. SVM also achieved very close results in terms of average specificity (94.69), average recall (87.71), and average f1 score (88.35) considering the performance of fine-tuned Xception model. In addition, it outperformed fine-tuned Xception model at average precision metric (89.59) (see Table 5). According to confusion matrices in Figure 5 SVM is more successful than fine-tuned Xception model in detecting bacterial pneumonia (1 more case) and normal cases (6 more cases). In contrast, the fine-tuned Xception model was better than SVM in detecting viral pneumonia cases (8 more cases). Table 6 and Table 7 shows class-based classification results of SVM and fine-tuned Xception model respectively. Recall (Sensitivity) is regarded as the most important metric in medical studies. SVM outperformed fine-tuned Xception model in terms of sensitivity metric in bacterial pneumonia and normal cases But, fine-tuned Xception achieved better sensitivity value at detecting viral pneumonia cases. As a result, fine-tuned Xception achieved better classification performance than other machine learning algorithms. In addition, it is clearly seen from Table 6 and Table 7 that the detection of bacterial pneumonia cases is easier, while viral pneumonia cases are more difficult to detect. The main reason for this situation is that the number of images of bacterial pneumonia during the training is more than the other two classes, and the model has learned this type of pneumonia better than others.

Table 6 Class based classification results (%) of SVM algorithm

Class	Pre.	Recall	Spe.	F1-score	Sample
0 (Bacterial Pneumonia)	86.45	97.52	90.51	91.65	242
1 (Normal)	94.22	90.60	96.73	92.37	234
2 (Viral Pneumonia)	88.10	75.00	96.84	81.02	148

Table 7 Class based classification results (%) of fine-tuned Xception model

Class	Pre.	Recall	Spe.	F1-score	Sample
0(Bacterial Pneumonia)	89.69	97.11	92.93	93.25	242
1(Normal)	96.26	88.03	97.94	91.96	234
2(Viral Pneumonia)	80.41	80.41	93.90	80.41	148

After the outbreak of Covid 19, most of the studies have been focused on differentiae pneumonia from Covid 19. There are few studies for detecting types of pneumonia and normal cases in chest X-Ray images. Prayogo et al used SCN models to classify chest X-Rays into 3 three classes normal, bacterial, and viral pneumonia. They achieved 80.03 accuracy and 79.59 F1-score values; our results are better than theirs in terms of all metrics. They used the same dataset used in our study [26]. Darici et al used a CNN ensemble voting methodology to diagnose viral pneumonia, bacterial pneumonia, and normal cases. They used three CNN models, two of them were designed by them and the last one was the pre-trained Inception V3 model. The used dataset in their study is the same as in our study [26]. However, they utilized offline data augmentation to balance the dataset. The classification performance of their CNN ensemble voting method is given in Table 8. We used an imbalanced dataset in our experiments and but our results are better than theirs. Mahmud et al designed a CNN model (CovXNet) utilizing depthwise separable convolutions to classify chest X-Rays into four classes (normal cases, viral pneumonia, bacterial pneumonia, and Covid-19). The used dataset in their study is the same within used in our study [26]. They used a two-step transfer learning methodology. First, they trained the model with chest X-Ray dataset which are including normal cases, viral pneumonia, and bacterial pneumonia images. Then, they trained again pre-trained model with chest X-Ray dataset which are including normal cases and Covid-19 cases. Their proposed CNN model achieved better results in terms of all metrics. However, designing and training a CNN model from scratch needs too much experience and effort. In addition, their inception model classification performance is less successful than our method (see Table 8).

On the other hand, retraining a pre-trained CNN model for a specific problem has some challenges. There are hyperparameters in the retraining process such as the number of freezing layers, fully connected layer size, fully connected neuron number, epoch size. Suitable hyperparameters are obtained after different experiments and this may take some time. This is the limitation of our study.

Table 8 The performance comparison of the proposed method with literature

References	**Classes**	**Acc**	**Recall**	**Pre.**	**F1-score**
Prayogo et al. [20]	3	80.03	80.03	79.23	79.59
Darici et al. [25]	3	78.00	75.00	77.0	75.00
Mahmud et al. [23]	3	91.70	92.10	92.90	92.60
Mahmud et al. (Inception) [23]	3	81.10	84.90	75.40	78.90
Proposed SVM	3	89.58	87.71	89.59	88.35
Proposed Xception	3	89.74	89.74	85.95	89.72

6. Conclusion

In this study, we trained a fine-tuned Xception model to classify chest X-Rays into three classes bacterial pneumonia, normal and viral pneumonia. Then, the fine-tuned Xception model was used as a feature extractor for various machine learning algorithms. Five machine learning algorithms (SVM, KNN, LR, NB, DT) were trained using the 2048 features extracted by the fine-tuned Xception model's convolutional layers. Then, we compared the classification performance of algorithms. According to the test results, the fine-tuned Xception model achieved better classification results with 89.74% accuracy than other classifiers. Among the machine learning classifiers, SVM algorithm achieved the best score with 89.58% accuracy. Our findings show that a pre-trained CNN model can be used successfully as a feature extractor instead of handcrafted features for different machine learning algorithms.

In future work, we will use multiple CNN models as feature extractors. Then, we will concatenate extracted features and will use them to train an ensemble of ML classifiers.

Acknowledgments

Author 1: performed the design and implementation of the research, analysis of the results and writing the article. There is no need to obtain permission from the ethics committee for the article prepared. There is no conflict of interest with any person / institution in the article prepared.

References

[1] D. You, G. Jones, and T. Wardlaw, "Levels & Trends in Child Mortality: Report 2011. Estimates Developed by the UN Inter-Agency Group for Child Mortality Estimation.," New York: United Nations Children's Fund2011.

[2] WHO, "Priority diseases and reasons for inclusion," in *Chapter 6.22-Pneumonia*, 2014.

[3] O. Ruuskanen, E. Lahti, L. C. Jennings, and D. R. Murdoch, "Viral pneumonia," *The Lancet,* vol. 377, no. 9773, pp. 1264-1275, 2011.

[4] D. E. Drake, A. Cohen, and J. Cohn, "National hospital antibiotic timing measures for pneumonia and antibiotic overuse," *Quality Management in Healthcare,* vol. 16, no. 2, pp. 113-122, 2007.

[5] WHO, "Standardization of interpretation of chest radiographs for the diagnosis of pneumonia in children," Geneva: World Health Organization2001.

[6] M. I. Neuman *et al.*, "Variability in the interpretation of chest radiographs for the diagnosis of pneumonia in children," *Journal of Hospital Medicine,* vol. 7, no. 4, pp. 294-298, 2012.

[7] J. Ker, L. Wang, J. Rao, and T. Lim, "Deep learning applications in medical image analysis," *IEEE Access,* vol. 6, pp. 9375-9389, 2017.

[8] D. Shen, G. Wu, and H.-I. Suk, "Deep learning in medical image analysis," *Annual Review of Biomedical Engineering,* vol. 19, pp. 221-248, 2017.

[9] M. A. Mazurowski, M. Buda, A. Saha, and M. R. Bashir, "Deep learning in radiology: An overview of the concepts and a survey of the state of the art with focus on MRI," *Journal of Magnetic Resonance Imaging,* vol. 49, no. 4, pp. 939-954, 2019.

[10] Y. LeCun, K. Kavukcuoglu, and C. Farabet, "Convolutional networks and applications in vision," in *Proceedings of 2010 IEEE international symposium on circuits and systems*, 2010, pp. 253-256: IEEE.

[11] M. A. Al-Antari, M. A. Al-Masni, and T.-S. Kim, "Deep learning computer-aided diagnosis for breast lesion in digital mammogram," *Deep Learning in Medical Image Analysis,* pp. 59-72, 2020.

[12] H. Li, A. Li, and M. Wang, "A novel end-to-end brain tumor segmentation method using improved fully convolutional networks," *Computers in biology medicine,* vol. 108, pp. 150-160, 2019.

[13] A. Esteva *et al.*, "Dermatologist-level classification of skin cancer with deep neural networks," *Nature,* vol. 542, no. 7639, p. 115, 2017.

[14] P. Rajpurkar *et al.*, "Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning," *arXiv preprint arXiv:1711.05225,* 2017.

[15] D. S. Kermany *et al.*, "Identifying medical diagnoses and treatable diseases by image-based deep learning," *Cell,* vol. 172, no. 5, pp. 1122-1131. e9, 2018.

[16] G. Liang and L. Zheng, "A transfer learning method with deep residual network for pediatric pneumonia diagnosis," *Computer Methods and Programs in Biomedicine,* p. 104964, 2019.

[17] V. Chouhan *et al.*, "A Novel Transfer Learning Based Approach for Pneumonia Detection in Chest X-ray Images," *Applied Sciences,* vol. 10, no. 2, p. 559, 2020.

[18] X. Gu, L. Pan, H. Liang, and R. Yang, "Classification of bacterial and viral childhood pneumonia using deep learning in chest radiography," in *Proceedings of the 3rd International Conference on Multimedia and Image Processing*, 2018, pp. 88-93.

[19] A. Mittal *et al.*, "Detecting Pneumonia using Convolutions and Dynamic Capsule Routing for Chest X-ray Images," *Sensors,* vol. 20, no. 4, p. 1068, 2020.

[20] K. A. Prayogo, A. Suryadibrata, and J. C. Young, "Classification of pneumonia from X-ray images using siamese convolutional network," *Telkomnika,* vol. 18, no. 3, pp. 1302-1309, 2020.

[21] T. Rahman *et al.*, "Transfer learning with deep convolutional neural network (CNN) for pneumonia detection using chest X-ray," *Applied Sciences,* vol. 10, no. 9, p. 3233, 2020.

[22] M. F. Hashmi, S. Katiyar, A. G. Keskar, N. D. Bokde, and Z. W. Geem, "Efficient pneumonia detection in chest xray images using deep transfer learning," *Diagnostics,* vol. 10, no. 6, p. 417, 2020.

[23] T. Mahmud, M. A. Rahman, and S. A. Fattah, "CovXNet: A multi-dilation convolutional neural network for automatic COVID-19 and other pneumonia detection from chest X-ray images with transferable multi-receptive feature optimization," *Computers in biology medicine,* vol. 122, p. 103869, 2020.

[24] K. El Asnaoui, "Design ensemble deep learning model for pneumonia disease classification," *International Journal of Multimedia Information Retrieval*, vol. 10, no. 1, pp. 55-68, 2021.

[25] M. B. Darici, Z. Dokur, and T. Olmez, "Pneumonia Detection and Classification Using Deep Learning on Chest X-Ray Images," *International Journal of Intelligent Systems Applications in Engineering*, vol. 8, no. 4, pp. 177-183, 2020.

[26] D. Kermany and M. Goldbaum, "Labeled optical coherence tomography (OCT) and Chest X-Ray images for classification," *Mendeley Data,* vol. 2, 2018.

[27] J. Koushik, "Understanding convolutional neural networks," *arXiv preprint arXiv:1605.09081,* 2016.

[28] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1251-1258.

[29] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big Data,* vol. 3, no. 1, p. 9, 2016.

[30] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248-255: IEEE.

[31] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural Computation*, vol. 29, no. 9, pp. 2352-2449, 2017.

[32] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *Journal of Big Data*, vol. 6, no. 1, p. 60, 2019.

[33] E. Ayan and H. M. Ünver, "Data augmentation importance for classification of skin lesions via deep learning," in *Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT)*, 2018, pp. 1-4: IEEE.

[34] G. Bonaccorso, *Machine learning algorithms*. Packt Publishing Ltd, 2017.