Application of MobileNetV2 and SVM Combination for Enhanced Accuracy in Pneumonia Classification

Eka Putra Agus Meindiawan [1]*, Muljono [2]**
[1,2] Faculty of Computer Science, Universitas Dian Nuswantoro, Semarang, 50131, Indonesia
111202113269@mhs.dinus.ac.id [1], muljono@dsn.dinus.ac.id [2]

Article Info

Article history:

Received 2024-09-03
Revised 2024-09-14
Accepted 2024-09-18

Keyword:

Pneumonia,
CXR,
Sobel,
MobileNet,
SVM.

ABSTRACT

Pneumonia is a very common respiratory infection in low- and middle-income countries and is still a leading cause of death, especially among children under five years old. Modern technologies, such as machine learning, offer significant potential in improving the automatic detection of pneumonia through chest X-ray (CXR) image analysis. This study aims to develop a more accurate pneumonia diagnosis system by evaluating various feature extraction methods. CXR datasets of pneumonia patients were resized to 180x180 pixels and balanced using the SMOTE-Tomek technique. Three main approaches were investigated: direct classification using Support Vector Machine (SVM) on the SMOTE-Tomek balanced dataset, feature extraction using Sobel edge detection followed by SVM classification, and feature extraction using MobileNet-V2 followed by SVM classification. The results showed that Scheme 1 achieved 97% accuracy, Scheme 2 decreased to 95%, and Scheme 3 achieved the highest accuracy at 98%. The lower accuracy in Scheme 2 is due to the limitations of Sobel edge detection, which reduces the key features in the CXR image. On the other hand, the improvement in Scheme 3 is due to the effective feature extraction capability of MobileNet-V2. In conclusion, the choice of feature extraction method plays an important role in determining the accuracy of an automated diagnostic system. This study builds on existing research and is expected to make a significant contribution to the development of more accurate and efficient automated diagnostic systems, which can ultimately help reduce pneumonia-related mortality.

This is an open access article under the CC–BY-SA license.

I. INTRODUCTION

Mycoplasma pneumonia, or more commonly known as atypical pneumonia, is a respiratory infection caused by the bacterium Mycoplasma pneumoniae[1].This disease spreads through respiratory droplets containing bacteria and can lead to various respiratory issues, ranging from tracheobronchitis to upper respiratory tract disease and pneumonia. The most common symptoms of this infection include a persistent dry cough, fever, shortness of breath, and excessive fatigue[2]. Pneumonia is one of the deadliest diseases, causing high mortality among adolescents, especially in low- and middle-income countries[3]. Every year, many people die because they do not receive proper treatment at the right time. Children under five years old are particularly vulnerable to this disease, with about 95% of deaths occurring in developing countries. For instance, in 2015, approximately 17,850 children under the age of five died from pneumonia in Bangladesh, meaning about three children every hour, 67 children every day, and 24,300 children every year[4]. Pneumonia is often misdiagnosed due to a lack of adequate medical resources. Appropriate antibiotic treatment can be administered earlier if pneumonia can be diagnosed at an early stage. For diagnosing pneumonia, chest X-rays (CXR) are used to show the location of the infection and the extent to which it has spread in the lungs[5]. Radiologists analyse these CXR images to detect the presence of pneumonia or other lung diseases[6]. However, this process is time-consuming and challenging, as CXR images are sometimes unclear and blurry, which can lead to detection errors.

In this modern technological era, a reliable diagnostic system is urgently needed to reduce mortality rates[7].

Machine learning (ML) plays a major role in automatically detecting diseases, such as heart disease, breast cancer, and brain cancer. Using CXR images, ML models can be trained to develop a reliable automated pneumonia detection system and reduce the workload of radiologists.

Research in 2021 by MD. Nahiduzzaman et al. used extreme learning machine (ELM) with a hybrid convolutional neural network-principal component analysis (CNN-PCA) for CXR pneumonia images, concluding with a classification recall value of 98% and an accuracy of 98.32% for multiclass pneumonia classification. For binary classification, a recall value of 100% and an accuracy of 99.83% were achieved[4].

Research by Adnan Hussain in 2023 leveraged the capabilities of the VGG-16, DenseNet-201, and Efficient-B0 models using transfer learning techniques to extract deep features from images, showing that the proposed system outperforms contemporary techniques in terms of precision, recall, F1 score, and accuracy (acc). The proposed method achieved 97% acc, while scoring 96%, 95%, and 97% in precision, recall, and F1 respectively[8].

Further research by Elene Firmeza Ohata et al., in 2021 utilized various convolutional neural network (CNN) architectures trained on ImageNet, adapting them to act as feature extractors for X-ray images. The CNN was then combined with consolidated machine learning methods such as k-Nearest Neighbor, Bayes, Random Forest, multilayer perceptron (MLP), and support vector machine (SVM). The best-performing combination was DenseNet201 with MLP, which achieved 95.6% accuracy and F1 score[9].

Research in 2023 by Shuohan Xue and Charith Abhayaratne employed pre-processing techniques including lung segmentation to identify regions of interest, volume resampling, and a novel approach for extracting significant slices. This was followed by proposing region-aware 3D ResNet for feature learning. The backbone networks used in this research included 3D ResNet-18, 3D ResNet-50, and 3D ResNet-101, resulting in an overall accuracy of 90%[10].

From several research studies, it can be concluded that to improve ML model accuracy, proper feature extraction is crucial in identifying pneumonia or other lung diseases. Edge detection is one of the main feature extraction tools used in image analysis and pattern recognition. The Sobel operator is often used in edge detection due to its fast computation speed and good detection results. MobileNet-V2 is one of the pre-trained models that can be used for feature extraction. MobileNet-V2, in particular, is effective in object detection for various image modalities, including CXR images that may show signs of pneumonia.

With many feature extraction methods available, we propose to examine the impact of various feature extraction methods in improving the classification accuracy of X-ray images for pneumonia detection. Thus, it is hoped that this research can significantly contribute to the development of a more accurate and reliable automated diagnosis system, and help reduce the mortality rate due to pneumonia.

II. METHODOLOGY

Data scientists have utilized various machine learning (ML) architectures to predict life-threatening diseases such as COVID-19, pneumonia, and heart disease using medical images over the past few decades. This research aims to accurately identify pneumonia from X-ray (CXR) images by leveraging the benefits of ML algorithms.

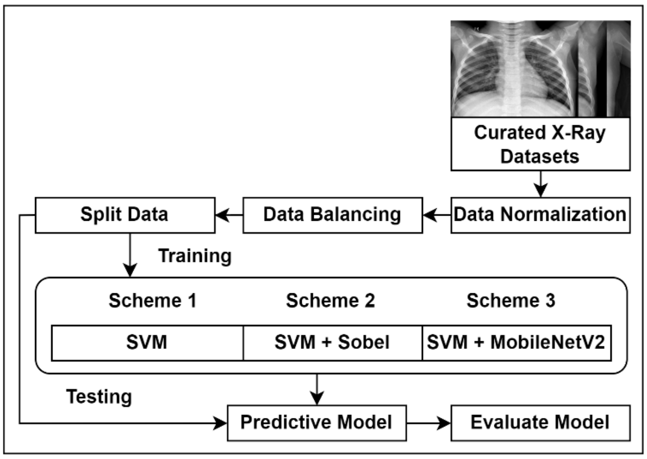

Figure 1. Research Method

As illustrated in Figure 1, the initial step in this study involves collecting CXR images from pneumonia patients and resizing them to 180x180 pixels to expedite the research process. To address class imbalance within the dataset, the SMOTE Tomek technique is employed, which combines the Synthetic Minority Over-sampling Technique (SMOTE) with Tomek links.

Once the data is balanced, the study is conducted in three schemes. The first scheme involves direct classification using Support Vector Machine (SVM) on the SMOTE Tomek dataset. The second scheme incorporates feature extraction using the Sobel edge detection method prior to classification with SVM. The third scheme uses MobileNet-V2 for feature extraction before applying SVM classification. These three schemes are analysed and compared to determine the most effective method for classifying CXR images of pneumonia. The findings of this research are anticipated to make a significant contribution to the development of more accurate and efficient automatic diagnosis systems, ultimately helping to reduce pneumonia-related mortality rates.

A. Datasets

Chest X-ray (CXR) images for pneumonia were collected from the Women and Children's Medical Center in Guangzhou and are also available on Kaggle. This dataset contains 5,938 CXR images with resolutions ranging from 400 to 2,000 pixels.

The images are categorized into three groups: normal, bacterial pneumonia, and COVID-19. The current dataset consists of 1,281 X-ray images for COVID-19, 3,270 images for normal cases, and 3,001 images for bacterial pneumonia.

TABLE 1.
UNBALANCED DATASETS

Class	Jumlah Data
Covid-19	1.281
Pneumonia Bacterial	3.270
Normal	3.001

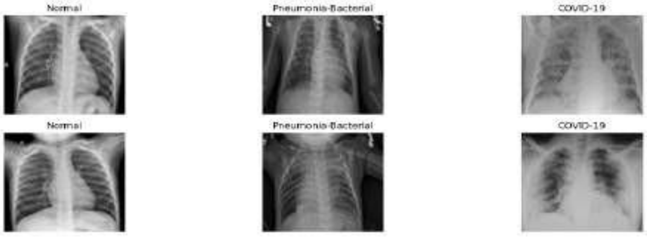

Figure 2. Sample Data From the Dataset

To address the imbalance, oversampling and undersampling techniques are applied using the SMOTE-Tomek method. This strategy balances the dataset, bringing the total number of images to 9,810, with each class containing 3,270 images. Once the dataset is balanced, it is split into training and testing sets with an 80:20 ratio. This split ensures a robust training process, providing 7,848 images for training and 1,962 images for testing, facilitating the development of a more reliable model for classifying normal, bacterial pneumonia, and COVID-19 cases. This image collection offers a broad range of resolutions and categories, providing a solid foundation for in-depth analysis and disease detection model development. It not only increases the amount of data available for medical research but also supports the advancement of better classification methods for the diagnosis and monitoring of respiratory diseases such as pneumonia and COVID-19.

B. Data Normalization

Since each image contains numerous intensity values, normalization is performed to reduce complexity without handling a very large number of pixels. This process involves scaling the range from 0-255 to 0-1 by dividing each pixel value by 128. This simplification aids in streamlining the image and accelerating the analysis process.

C. Data Balancing

SMOTE (Synthetic Minority Over-sampling Technique) and Tomek Links are employed in this study to address class imbalance[11]. SMOTE is an oversampling technique that increases the number of samples in the minority class by generating synthetic data from existing samples[12]. This technique works by randomly selecting samples from the minority class, finding their nearest neighbors, and creating synthetic data by interpolating between the sample and its neighbors with a random multiplier. In contrast, Tomek Links is an undersampling strategy that reduces the number of samples in the majority class. This method identifies pairs of samples from different classes that are nearest neighbors to each other[13]. These pairs, known as Tomek Links, are then removed from the majority class. In addition, data balancing with SMOTE-Tomek affects the performance of models trained with balanced data tend to perform better in terms of precision, recall, and f1-score, especially in detecting bacterial pneumonia and COVID-19 cases, which were previously underrepresented in the unbalanced dataset. This balancing also helps to reduce bias in the majority class, thus improving the model's ability to provide more accurate and fairer predictions for all classes.

D. Split Data

The next step involves splitting the data into training and testing subsets using a stratified label method with an 80:20 ratio after completing data preprocessing and handling class imbalance. Preparing data for machine learning models is a crucial step. The Train-Test Split method divides the processed data into two subsets: training data and testing data [14]. With an 80:20 ratio, 80% of the data is used to train the model, while the remaining 20% is used to evaluate the model's effectiveness. The stratified label method ensures that the class distribution in the original dataset is preserved in both subsets, so the test data accurately represents all classes.

E. Feature Extraction with Edge Detection Sobel

Sobel edge detection is a technique in image processing used to detect edges or changes in intensity within an image[15]. This technique employs the Sobel operator, which is a mask (or kernel) used to compute the gradient of image intensity in both horizontal and vertical directions[16]. The intensity gradients are used to highlight sudden changes in intensity, typically indicating the edges of objects within the image. The Sobel operator utilizes two 3x3 kernels applied to the original image to estimate the derivatives in the horizontal (Gx) and vertical (Gy) directions.

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

$$G = \sqrt{G_x^2 + G_y^2}$$

The two kernels (Gx and Gy) are convolved with the original image to produce two new images representing the intensity changes in the horizontal and vertical directions. The results of this convolution are used to calculate the gradient magnitude at each point (x, y) in the image using the formula(G).

F. Feature Extraction with MobileNet-V2

MobileNet-V2 is a type of Convolutional Neural Network (CNN) architecture designed for devices with low computational power, such as mobile phones and single-board computers[17]. It is an improvement over previous cellular network architecture. One of the main differences between cellular network architectures and CNNs in general is the use of convolutional layers.

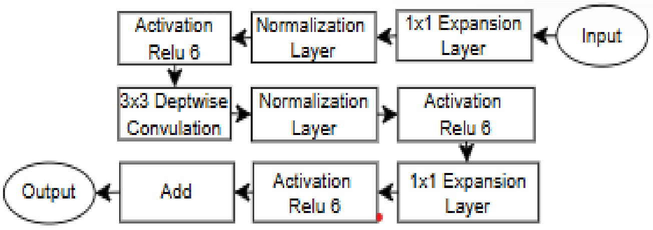

Figure 3. Architecture oh linear bottleneck

In MobileNet-V2, convolutional layers use filter thickness that matches the input image's thickness. This technique involves the use of depthwise separable convolutions, pointwise convolutions, linear bottleneck, and shortcut connections between bottlenecks. The MobileNet-V2 architecture consists of several bottleneck layers, as illustrated in Figure 3. These bottleneck layers perform three main operations: 1x1 expansion, 3x3 depthwise convolution, and 1x1 pointwise convolution. Additionally, there is a residual connection feature when the input size is the same as the output size. MobileNet-V2 features 53 convolutional layers and one average pooling layer. These changes in MobileNet-V2 architecture enhance classification accuracy and detection speed compared to both MobileNet-V2 and traditional CNN architectures.

G. *Modeling using Support Vector Machine (SVM)*

Support Vector Machines (SVM) are a classification method that works by finding a hyperplane with the maximum margin[18]. The hyperplane can be either linear or non-linear depending on the data conditions. The hyperplane is determined by the margin, which is the closest distance between data points in each class to the hyperplane. SVM operates on the principle of linear classification, meaning it can separate classes that are linearly separable[19]. However, SVM has been extended to handle non-linear problems by introducing the concept of kernels in a higher-dimensional space. In this higher-dimensional space, the goal is to find the hyperplane that maximizes the margin between data classes According to Santosa (2007), the linear SVM classification hyperplane is denoted as follows[20].

$$f(x, w, b) = sign(W^{T}Xi + b)$$

In SVM classification, the method involves finding the optimal hyperplane that acts as a separator between two data classes. To find the best hyperplane, the margin of the hyperplane is measured, and the maximum margin is sought. The margin is defined as the distance between the hyperplane and the closest patterns from each class[21]. The closest patterns to the hyperplane are referred to as support vectors.

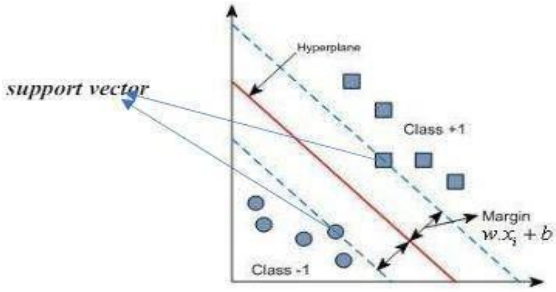

Figure 4 Structure of SVM

Based on the equation in Figure 4, Structure of SVM for Two-Class Separation (+1 and -1) Based on the Hyperplane Line, if $wx1 + b = +1$ represents the supporting hyperplane for class +1 and $wx2 + b = -1$ represents the supporting hyperplane for class -1, then the margin can be calculated by finding the distance between the two supporting hyperplanes. Thus, the margin is given by:

$$(wx1 + b = +1) - (wx2 + b = -1) => w(x1 - x2) = 2$$

$$\left(\frac{w}{||w||}(x1 - x2)\right) = \frac{2}{||w||}$$

The largest margin can be found by maximizing the distance between the hyperplane and its closest points, which isu 1/ ‖$\boldsymbol{w}$ →‖. Hal ini dapat dirumuskan sebagai Quadratic Programming (QP) Problem. his can be formulated as a Quadratic Programming (QP) Problem. For linear classification in the primal space, the SVM optimization formulation is as follows:

Xi, X2, …, Xn are independent variables or factors that influence the dependent variable. These factors could include technical indicators, fundamental data, or any other relevant metrics.

$$\frac{1}{\min\left(||w||^{2}\right)2}$$

$$Given\ the\ constraint\ Yi(W^{T}.Xi + b) \geq 1, \quad i = 1,2,3, \ldots n$$

This problem can be solved using various methods, one of which is the Lagrange Multiplier.

$$L(w, b, \alpha) = \frac{1}{2}W^{T}W - \sum_{i=1}^{n} \alpha i\{Yi[(W^{T}.Xi) + b] - 1\}\ (i = 1,2,3, \ldots., l)$$

αi are Lagrange multipliers, which are either zero or positive ($\alpha i \geq 0$). The optimal values of the above equation can be found by minimizing L with w → and b, and maximizing L with αi. To handle non-linear problems, SVM is modified by incorporating a kernel function. In non-linear SVM, first x data is first mapped by a function Φ(x) to a higher-dimensional vector space. The hyperplane that separates the two classes can then be constructed in this higher-dimensional space. As illustrated in Figure 4, the function Φ maps each data point in the input space to a new vector space with higher dimensions (dimension 3), allowing the two classes to be separated linearly by a hyperplane. Commonly used kernel functions in SVM include:

TABLE 2.
KARNEL FUNCTION

Linier	$K(Xi, Xj) = X^T Xj\ i$
Polynomial	$K(xi, xj) = (\gamma X^T Xj + r)^p, \gamma > 0\ i$
Radial Basis Function (RBF)	$2\ K(Xi, Xj) = \exp[-\gamma \lvert\lvert Xi - Xj \rvert\rvert], \gamma > 0$
Sigmoid	$K(Xi, Xj) = \tanh(\gamma X^T Xj + r)\ i$

H. Evaluating Model

A confusion matrix is a tool used in machine learning and statistics to assess the performance of classification algorithms[22]. It is a square matrix commonly used to summarize the results of a classification problem. The confusion matrix provides detailed insights into the correct and incorrect predictions made by a classification model. This tool is particularly useful for binary classification problems (two classes) but can also be applied to multi-class classification[23].

In a confusion matrix, there are four terms that describe the results of the classification process. As shown in Table II, these terms are True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). TP and TN indicate correct classification results, while FP and FN represent incorrect classification results. The formula for calculating matrix evaluated is shown in Table 2.

TABLE 3.
FORMULA OF MATRIX EVALUATION

Precision	$= \frac{TP}{TP + FP}$
Sensitifity	$= \frac{TP}{TP + FN}$
F1-Score	$= 2x \frac{Precision \times Sensitifity}{Precision + Sensitifity}$
Accuracy	$= \frac{TP + TN}{TP + TN + FP + FN}$

III. RESULT AND DISCUSSION

This research utilizes the Google Colab platform, leveraging a T4 GPU with 15 GB for computational tasks and 12 GB of RAM to facilitate data processing and training across the three research schemes. Additionally, data analysis is conducted using plot_pca_scatter to visualize the distribution of the data that will be classified.

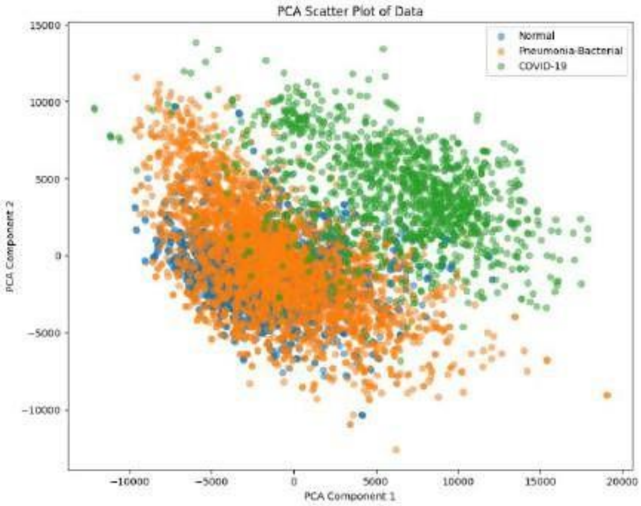

Figure 5. Plot Before Classification

As seen in Figure 5, the data distribution is still biased, necessitating preprocessing. The preprocessing process begins with applying the SMOTE-Tomek technique to address the class imbalance issue. To balance the distribution of images across each class, the SMOTE-Tomek technique is utilized. At this stage, the initial dataset of 5,938 CXR images, with the following distribution: 1,281 images for COVID-19, 3,270 images for normal cases, and 3,001 images for bacterial pneumonia, is transformed into a more balanced dataset. SMOTE-Tomek combines Synthetic Minority Over-sampling Technique (SMOTE) with Tomek Links. SMOTE generates synthetic samples for the minority class, while Tomek Links removes examples from the majority class that are close to the minority class.

The process starts by reshaping the image data from a 4D format (batch, height, width, channels) to a 2D format with data.reshape(-1, 128 * 128 * 3). With this data shape, SMOTE-Tomek can apply oversampling and cleaning to correct class imbalance. The result from smote_tomek.fit_resample(data.reshape(-1, 128 * 128 * 3), labels) includes data_resampled and labels_resampled, where data_resampled represents the resampled images, and labels_resampled are the corresponding labels after balancing. This data is then reshaped back to the 4D image format with data_resampled.reshape(-1, 128, 128, 3) for further analysis. The class distribution after balancing is plotted to visualize the changes.

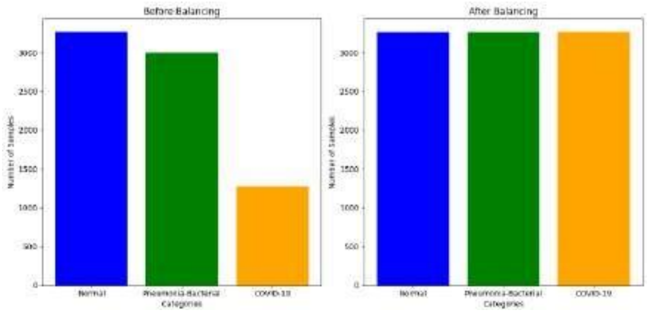

Figure 6. Before and After Balancing

Figure 6. Before and After Balancing shows the result of SMOTE-Tomek with the initial data consisting of 5,938 CXR images: 1,281 X-ray images for COVID-19, 3,270 images for normal cases, and 3,001 images for bacterial pneumonia. This data was transformed into 9,810 images, with each class having 3,270 images, grouped into the resampled data.

The first scheme of this study involves classifying the SMOTE-Tomek processed data using Support Vector Machine (SVM). SVM is an effective classification algorithm that separates data with a maximum margin. The data is split into training and testing sets using train_test_split with parameters test_size=0.2 to ensure that 20% of the data is used for testing, and random_state=42 for result consistency. The SVM model with a linear kernel is trained using the training data and tested on the testing data.

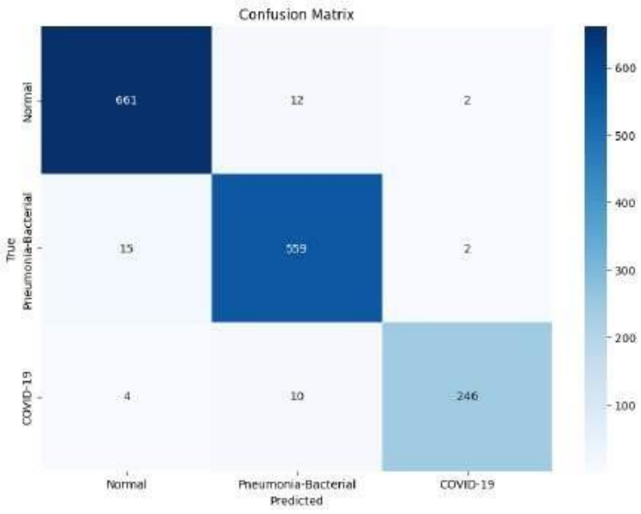

Figure 7.Confusion Matrix Scheme 1

The evaluation of this classification reveals highly satisfactory results, with an impressive accuracy rate of 97%. Additionally, the model achieved an average precision of 97%, indicating a strong ability to correctly identify positive cases out of all positive predictions. The sensitivity, also at 97%, demonstrates the model's effectiveness in correctly detecting positive cases among all actual positives. Similarly, the F1 score, which balances precision and recall, stands at 97%, reflecting the model's consistent and reliable performance across various metrics. These results underscore the robustness of the model in accurately identifying and classifying the tested cases.

For further visualization, the evaluation matrix shown below provides a clear depiction of the model's predictions compared to the actual labels. This matrix offers insights into how well the model predicts different classes within the dataset and highlights where classification errors occur.

Building on the success of Scheme 1, Scheme 2 involves resampling data followed by feature extraction using Sobel edge detection. The initial step involves converting images from color (RGB) to grayscale. Each image in the dataset is converted to grayscale using cv2.cvtColor(img, cv2.COLOR_RGB2GRAY), which simplifies the images and focuses solely on pixel intensity.

Next, Sobel edge detection is applied to extract edge features from the grayscale images. This process involves two stages: detecting horizontal and vertical edges. The horizontal gradient is computed using sobelx = cv2.Sobel(gray, cv2.CV_64F, 1, 0, ksize=5), while the vertical gradient is computed with sobely = cv2.Sobel(gray, cv2.CV_64F, 0, 1, ksize=5). The horizontal and vertical gradients are then combined to obtain the total edge magnitude using sobel = np.hypot(sobelx, sobely). Finally, the edge magnitude is normalized to ensure pixel values fall within the range of 0 to 255,using sobel = (sobel/sobel.max()* 255).astype(np.uint8).

TABLE 4.
FEATURE SOBEL EXTRACTION

feature	0	1	…	16382	16383
count	9806	9806	…	9806	9806
mean	0.0	35.41	…	21.88	0.0
std	0.0	0.000	…	37.47	0.0
min	0.0	2.000	…	0.0	0.0
50%	0.0	13.00	…	4.000	0.0
max	0.0	255.0	…	255.0	0.0

Sobel edge detection extraction results in the generation of 16,383 features in the form of array data, which are then used for classification with the SVM (Support Vector Machine) algorithm. This process is designed to predict the label of each image based on the edge features extracted by the Sobel method. The classification outcomes, summarized in Table 3, indicate a satisfactory performance with accuracy, recall, and F1 score values of 95% each. These results suggest that combining Sobel edge detection with SVM is effective for classifying the sampled images. However, it is important to consider that while Sobel edge detection is effective in highlighting edges and can improve classification performance in some cases, it may not capture finer details necessary for more nuanced classifications, especially in medical imaging contexts where subtle texture and pattern recognition are crucial.

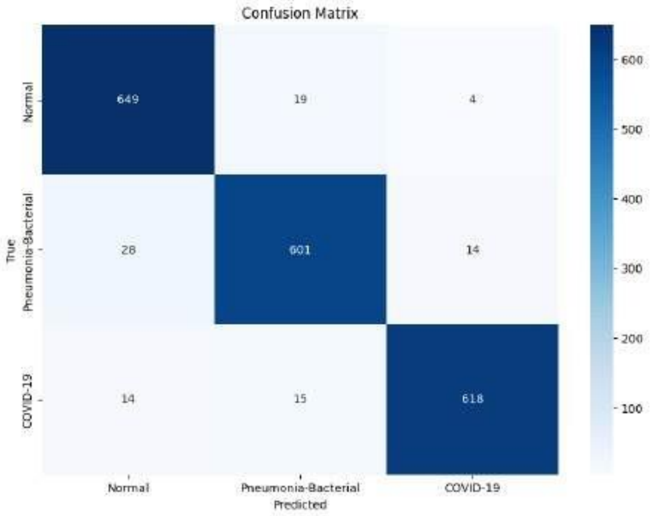

Figure 8. Confusion Matrix Scheme 2

For further visualization, the evaluation matrix image below provides a clear overview of the model's predictions compared to the actual labels. This matrix helps us understand how well the model predicts various classes within the dataset and where classification errors occur.

The third scheme involves feature extraction using the MobileNet-V2 architecture, an efficient convolutional neural network designed for resource-constrained environments.

Initially, the SMOTE-Tomek balanced dataset is preprocessed to match the input format expected by MobileNet-V2 using the preprocess_input(data_resampled) function. This step ensures the data is correctly scaled and prepared for the model. Once the data is preprocessed, features are extracted from the images using a pre-trained MobileNet-V2 model by calling mobilenet.predict(data_resampled_preprocessed), generating feature vectors that encapsulate the essential characteristics of each image. After feature extraction, the dataset is split into training and testing sets using the train_test_split(features_resampled, labels_resampled, test_size=0.2, random_state=42) method, allocating 80% of the data for training and 20% for testing. The extracted features are then fed into an SVM model, which is trained using the svm.fit(X_train, y_train) function.

To further refine the model for this specific classification task, additional custom layers are added on top of the MobileNet-V2 base model. A GlobalAveragePooling2D layer is applied to the output of the base model, reducing the spatial dimensions of the feature maps by averaging them. This layer not only flattens the feature maps into a 1D vector but also reduces the overall number of parameters, helping to prevent overfitting. Following this, a fully connected Dense layer with a softmax activation function is added, where the number of units corresponds to the number of output classes. This layer provides the final classification predictions for the task. Once the model architecture is defined, it is compiled using the Adam optimizer with a learning rate of 0.0001, and categorical cross-entropy is used as the loss function, which is suitable for multi-class classification problems. The model is then trained for five epochs using a batch size of 32, with the training process monitored on the validation set (split using X_test and y_test), ensuring that the model learns effectively from the data while generalizing well to unseen examples.

TABLE 5.
FEATURE MOBILENET-V2 EXTRACTION

feature	0	1	…	1278	1279
count	9806	9806	…	9806	9806
mean	0.007	0.510	…	0.433	0.056
std	0.049	0.660	…	0.581	0.233
min	0.0	0.0	…	0.0	0.0
50%	0.0	0.220	…	0.158	0.0
max	1.134	4.491	…	3.488	3.950

Feature extraction using MobileNet-V2 produces 1279 features in the form of array data and then classified using the SVM (Support Vector Machine) algorithm. This classification process aims to predict the label of each image based on the extracted edge features. Evaluation of this classification scheme showed that the average values for precision, sensitivity, and F1 score were very satisfactory with 98% accuracy. This high level of accuracy and consistent metrics indicate that the model effectively captures and distinguishes between different classes, making it a robust tool for image classification tasks. The impressive performance highlights the effectiveness of MobileNet-V2 in extracting meaningful features and the efficacy of SVM in classification.

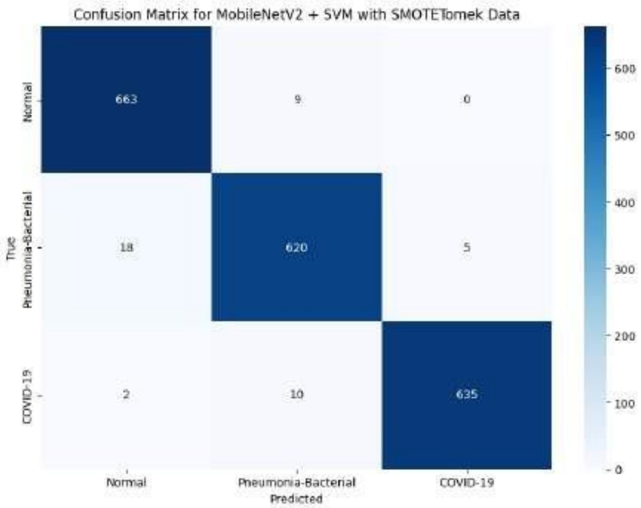

Figure 9. Confusion Matrix Schame 3

For further visualization, the evaluation matrix shown below offers a clear overview of the model's predictions in comparison with the actual labels. This matrix provides insights into the model's performance across different classes in the dataset, highlighting where classification errors occur and illustrating how effectively the model distinguishes between different categories.

TABLE 6.
EVALUATED MODELS

Scheme 1				
Class	**Precision**	**Sensitifity**	**F1-Score**	**Accuracy**
Normal	0.97	0.98	0.98	0.97
Bacterial-Pneumonia	0.96	0.97	0.97	0.96
Covid-19	0.98	0.95	0.96	0.96
Scheme 2				
Class	**Precision**	**Sensitifity**	**F1-Score**	**Accuracy**
Normal	0.94	0.97	0.95	0.95
Bacterial-Pneumonia	0.95	0.93	0.94	0.94
Covid-19	0.97	0.96	0.96	0.96
Scheme 3				
Class	**Precision**	**Sensitifity**	**F1-Score**	**Accuracy**
Normal	0.98	0.99	0.98	0.98
Bacterial-Pneumonia	0.98	0.97	0.98	0.97
Covid-19	0.99	0.99	0.99	0.99

Table 6 shows the results of scheme 1 where the average values of precision, sensitivity and f1-score were good enough to classify normal, pneumonia-bacterial, and covid-19 CXR images as 0.97, 0.96, 0.93 respectively. scheme 2 showed a decrease with the results of normal, pneumonia-bacterial, and covid-19 CXR classification as 0.95, 0.940, and 0.96 respectively. Scheme 3 showed the best results from the previous two schemes with the classification results of normal CXR, pneumonia-bacteria, and covid-19 being 0.980, 0.97, and 0.99 respectively.

V. CONCLUSION

This This research is a further development of previous studies in an effort to create a more accurate automated pneumonia diagnosis system, especially in areas with limited medical resources. Pneumonia, whether caused by Mycoplasma bacteria, viruses, or COVID-19, remains a serious threat, especially for children. One of the main challenges in identifying the disease is the quality of chest X-ray (CXR) images which are often unclear and limited medical resources. Through the application of machine learning technology, we seek to develop a model that is able to detect pneumonia more quickly and accurately, thereby easing the burden on medical personnel and potentially saving more lives.

In this study, we extend the previous approach by processing the CXR dataset of pneumonia patients, resizing the images to 180x180 pixels to speed up the analysis process. We also addressed class imbalance in the dataset using the SMOTE-Tomek technique, which combines synthetic over-sampling (SMOTE) with Tomek linkage to obtain a more balanced class distribution. After the balancing process, we performed three different approaches to evaluate how feature extraction methods can affect the classification accuracy of CXR images.

In the first approach, Support Vector Machine (SVM) was used directly on the balanced dataset. The results were satisfactory, with precision, sensitivity, and F1 scores reaching 0.976, 0.966, and 0.963 for the classification of normal, pneumonia-bacterial, and COVID-19 CXR images, respectively. Although these results demonstrate effectiveness, we believe that there are still opportunities for improvement with the use of more advanced techniques.

The second approach involves applying Sobel edge detection before classification with SVM. While this method is simple and computationally efficient, it results in a slight decrease in accuracy, with precision, sensitivity, and F1 scores of 0.953, 0.940, and 0.963, respectively. This reduction in performance is attributed to the loss of crucial information in the images. Sobel edge detection primarily focuses on highlighting the edges within an image, which can lead to the omission of important details necessary for accurate diagnosis. Specifically, Sobel edge detection may remove subtle texture patterns and fine details that are essential for distinguishing between different classes in chest X-ray (CXR) images. These fine details are critical in medical imaging for identifying nuances in lung pathology, which are often not well captured by edge detection alone. Thus, while Sobel edge detection can be useful for some applications, it may not be effective for tasks requiring detailed texture analysis and nuanced pattern recognition, as is the case in CXR image classification.

The third approach uses MobileNet-V2, an artificial neural network architecture, for feature extraction before classification with SVM. The results of this approach showed the best improvement, with precision, sensitivity, and F1 scores of 0.980, 0.966, and 0.986, confirming that MobileNet-V2 is highly effective in extracting critical features required for more precise classification.

With the results obtained, this study extends existing insights regarding the importance of feature extraction method selection in automated pneumonia diagnosis systems. The combined use of MobileNet-V2 and SVM with data that has been balanced using SMOTE-Tomek proved to be very effective, outperforming previous approaches. The contribution of this research is expected to serve as a foundation for further development in a more accurate and reliable automated diagnosis system, so that it can help reduce mortality from pneumonia, especially in areas with limited medical resources.

ACKNOWLEDGEMENTS

The author wishes to extend sincere appreciation to Universitas Dian Nuswantoro for their invaluable support for this research.

REFEERENCES

[1] A. Chharia et al., “Deep-Precognitive Diagnosis: Preventing Future Pandemics by Novel Disease Detection With Biologically-Inspired Conv-Fuzzy Network,” IEEE Access, vol. 10, pp. 23167–23185, 2022, doi: 10.1109/ACCESS.2022.3153059.

[2] A. Serener and S. Serte, “Deep learning for mycoplasma pneumonia discrimination from pneumonias like COVID-19,” in 4th International Symposium on Multidisciplinary Studies and Innovative Technologies, ISMSIT 2020 - Proceedings, Institute of Electrical and Electronics Engineers Inc., Oct. 2020. doi: 10.1109/ISMSIT50672.2020.9254561.

[3] T. S. Arulananth, S. W. Prakash, R. K. Ayyasamy, V. P. Kavitha, P. G. Kuppusamy, and P. Chinnasamy, “Classification of Paediatric Pneumonia Using Modified DenseNet-121 Deep-Learning Model,” IEEE Access, vol. 12, pp. 35716–35727, 2024, doi: 10.1109/ACCESS.2024.3371151.

[4] M. Nahiduzzaman et al., “A novel method for multivariant pneumonia classification based on hybrid CNN-PCA based feature extraction using extreme learning machine with CXR images,” IEEE Access, vol. 9, pp. 147512–147526, 2021, doi: 10.1109/ACCESS.2021.3123782.

[5] R. K. Sheu, M. S. Pardeshi, K. C. Pai, L. C. Chen, C. L. Wu, and W. C. Chen, “Interpretable Classification of Pneumonia Infection Using eXplainable AI (XAI-ICP),” IEEE Access, vol. 11, pp. 28896–28919, 2023, doi: 10.1109/ACCESS.2023.3255403.

[6] H. Malik et al., “A Novel Fusion Model of Hand-Crafted Features With Deep Convolutional Neural Networks for Classification of Several Chest Diseases Using X-Ray Images,” IEEE Access, vol. 11, pp. 39243–39268, 2023, doi: 10.1109/ACCESS.2023.3267492.

[7] C. Wijaya, H. Irsyad, and W. Widhiarso, “Klasifikasi Pneumonia Menggunakan Metode K-Nearest Neighbor Dengan Ekstraksi GLCM,” 2020.

[8] A. Hussain, S. U. Amin, H. Lee, A. Khan, N. F. Khan, and S. Seo, “An Automated Chest X-Ray Image Analysis for Covid-19 and Pneumonia Diagnosis Using Deep Ensemble Strategy,” IEEE Access, vol. 11, pp. 97207–97220, 2023, doi: 10.1109/ACCESS.2023.3312533.

[9] E. F. Ohata et al., “Automatic detection of COVID-19 infection using chest X-ray images through transfer learning,” IEEE/CAA J. Autom. Sin., vol. 8, no. 1, pp. 239–248, 2021, doi: 10.1109/JAS.2020.1003393.

[10] S. Xue and C. Abhayaratne, “Region-of-Interest Aware 3D ResNet for Classification of COVID-19 Chest Computerised Tomography Scans,” IEEE Access, vol. 11, no. January, pp. 28856–28872, 2023, doi: 10.1109/ACCESS.2023.3260632.

[11] M. M. S. Fareed et al., “ADD-Net: An Effective Deep Learning Model for Early Detection of Alzheimer Disease in MRI Scans,” IEEE Access, vol. 10, pp. 96930–96951, 2022, doi: 10.1109/ACCESS.2022.3204395.

[12] Muljono, S. A. Wulandari, H. Al Azies, M. Naufal, W. A. Prasetyanto, and F. A. Zahra, “Breaking Boundaries in Diagnosis: Non-Invasive Anemia Detection Empowered by AI,” IEEE Access, vol. 12, no. November 2023, pp. 9292–9307, 2024, doi: 10.1109/ACCESS.2024.3353788.

[13] Z. Wang, C. Wu, K. Zheng, X. Niu, and X. Wang, “SMOTETomek-Based Resampling for Personality Recognition,” IEEE Access, vol. 7, pp. 129678–129689, 2019, doi: 10.1109/ACCESS.2019.2940061.

[14] K. M. Kahloot and P. Ekler, “Algorithmic Splitting: A Method for Dataset Preparation,” IEEE Access, vol. 9, pp. 125229–125237, 2021, doi: 10.1109/ACCESS.2021.3110745.

[15] A. H. Abdel-Gawad, L. A. Said, and A. G. Radwan, “Optimized Edge Detection Technique for Brain Tumor Detection in MR Images,” IEEE Access, vol. 8, pp. 136243–136259, 2020, doi: 10.1109/ACCESS.2020.3009898.

[16] A. Ainun, D. Halim, and S. Anraeni, “Analisis Klasifikasi Dataset Citra Penyakit Pneumonia Menggunakan Metode K-Nearest Neighbor (KNN),” Indones. J. Data Sci., vol. 2, no. 1, pp. 1–12, 2021.

[17] S. Ki Hong and Y. Lee, “Optimizing Detection: Compact MobileNet Models for Precise Hall Sensor Fault Identification in BLDC Motor Drives,” IEEE Access, vol. 12, pp. 77475–77485, 2024, doi: 10.1109/ACCESS.2024.3407766.

[18] D. Hartanti and A. I. Pradana, “Komparasi Algoritma Machine Learning dalam Identifikasi Kualitas Air,” SMARTICS J., vol. 9, no. 1, pp. 1–6, 2023, [Online]. Available: https://doi.org/10.21067/smartics.v9i1.8113

[19] U. Demirel, H. Cam, and R. Unlu, “Predicting stock prices using machine learning methods and deep learning algorithms: The sample of the istanbul stock exchange,” Gazi Univ. J. Sci., vol. 34, no. 1, pp. 63–82, 2021, doi: 10.35378/gujs.679103.

[20] H. Hananti and K. Sari, “Perbandingan Metode Support Vector Machine (SVM) dan Artificial Neural Network (ANN) pada Klasifikasi Gizi Balita,” Semin. Nas. Off. Stat., vol. 2021, no. 1, pp. 1036–1043, 2021, doi: 10.34123/semnasoffstat.v2021i1.1014.

[21] D. Diana Dewi, N. Qisthi, S. S. S. Lestari, and Z. H. S. Putri, “Perbandingan Metode Neural Network Dan Support Vector Machine Dalam Klasifikasi Diagnosa Penyakit Diabetes,” Cerdika J. Ilm. Indones., vol. 3, no. 09, pp. 828–839, 2023, doi: 10.59141/cerdika.v3i09.662.

[22] R. Zhang, Q. Xiao, Y. Du, and X. Zuo, “DSPI Filtering Evaluation Method Based on Sobel Operator and Image Entropy,” IEEE Photonics J., vol. 13, no. 6, Dec. 2021, doi: 10.1109/JPHOT.2021.3118924.

[23] A. Nur, A. Thohari, A. Karima, K. Santoso, and R. Rahmawati, “Crack Detection in Building Through Deep Learning Feature Extraction and Machine Learning Approach,” vol. 8, no. 1, pp. 1–6, 2024.99