# Detecting Lung Disease Based on Chest X-ray Images Using a Hybrid CNN-KELM Approach

**Dian Candra Rini Novitasari[1], Musfiroh Musfiroh[1], and Dina Zatusiva Haq.[1]**
[1]Mathematics Department, Faculty of Science and Technology, UIN Sunan Ampel, Surabaya, Indonesia

**Corresponding author:** Musfiroh Musfiroh (musfiroh.rohmati22@gmail.com).

**ABSTRACT** Tuberculosis (TB) is a disease caused by the Mycobacterium tuberculosis (M.tb) bacterium. TB ranks among the top 10 deadliest diseases worldwide and is the second most contagious disease after COVID-19. The World Health Organization (WHO) recommends using Chest X-ray (CXR) imaging techniques, given their high sensitivity and cost-effectiveness. This study proposes a hybrid CNN-KELM (CKELM) method for the classification of four lung disease categories based on chest X-ray (CXR) images: tuberculosis, pneumonia, COVID-19, and normal, all within a short computational time. This study experimented with several types of CNN architectures implemented for feature extraction, while KELM for classification used hyperparameters that tested various kernel types and regularization coefficients. The experimental results indicate that the best performance is achieved using the DenseNet201 architecture with a polynomial kernel and a regularization coefficient of 0.1. The polynomial kernel demonstrates superior performance across all CNN architectures. Furthermore, a regularization coefficient of 0.1 exhibits the highest accuracy in the kernel and CNN architecture experiments. The DenseNet201-KELM model attains an accuracy, sensitivity, specificity, precision, and F1-score of 99.57%, 99.57%, 99.86%, 99.57%, and 99.57%, which is 7% better than without under sampling and detection using the DenseNet201-KELM method requires a computational time of 309.19 seconds. The proposed method achieved good performance in multi-class classification, especially for balanced data, with fast computational time.

**KEYWORDS** CNN, Hybrid CKELM, KELM, Lung Disease

## I. INTRODUCTION

Tuberculosis (TB) is a disease caused by the Mycobacterium tuberculosis (M.tb) bacterium. This bacterium can be transmitted through the air in various ways, such as coughing, saliva, water, and sneezing from TB patients. This disease affects people regardless of age [1]. TB doesn't only affect the lungs, known as Pulmonary Tuberculosis (PTB), but it can also affect other parts of the body, including the brain, kidneys, skin, bones, and lymph nodes, referred to as Extrapulmonary Tuberculosis (EPTB) [2]. TB presents with numerous symptoms, with the most common ones being cough, bloody sputum, weight loss, fever, night sweats, and chest pain. Additionally, other symptoms like chills, shortness of breath, fatigue, and loss of appetite may manifest. Such clinical symptoms pose a life-threatening risk if not diagnosed early. These symptoms bear similarities to pneumonia and COVID-19, which also affect the pulmonary region.

Pneumonia is an infection that primarily inflames the air sacs in the lungs responsible for oxygen exchange, while COVID-19, caused by the Severe Acute Respiratory Syndrome (SARS-CoV) virus, is also referred to as acute pneumonia. TB ranks among the top 10 deadliest diseases worldwide and is the second most contagious disease after COVID-19. Each year, around 10 million people worldwide contract TB, resulting in 1.4 million deaths [3]. In recent years, global TB incidents have declined, but they remain high in Indonesia. According to data from the Indonesian Ministry of Health, Indonesia records 969 thousand TB cases with 144 thousand deaths [4].

Accurate TB screening necessitates precise and reliable test result interpretation. WHO recommends the use of Chest X-rays (CXR) to detect lung abnormalities. CXR is a widely performed medical imaging technique known for its high sensitivity and cost-effectiveness. However, the limitation of radiologists with the required expertise to diagnose CXR poses a challenge when implemented on a large scale in countries with a high TB burden. Manual TB diagnosis by radiologists requires extensive experience, is time-consuming, and is susceptible to human errors [5].

Technological advancements in the screening process using Computer Aided Diagnosis (CAD) systems enable rapid and precise TB diagnosis [6].

CAD systems are trained with large datasets to automatically recognize patterns associated with TB in CXRs. The early detection process involves preprocessing, feature extraction, and classification steps [7]. Previous research has utilized CAD systems for TB diagnosis. Detected TB using the Histogram of Oriented Gradient (HOG) and Support Vector Machine (SVM), incorporating both linear and polynomial kernel functions. The diagnosis results has an average accuracy of 79.50% with a linear kernel and 77.50% with a polynomial kernel [8]. Research by [9] using Artificial Neural Network (ANN) method based on Hybrid Features from VGG16, incorporating LDG (LBP, DWT, and GLCM) in TB and pneumonia detection. The research achieved an accuracy of 99.60%, sensitivity of 99.17%, specificity of 99.42%, precision of 99.63%, and an AUC of 99.58%. Therefore, feature extractors play a crucial role in learning TB image features.

The CNN is one of algorithm that can effectively learn and extract image features. The application of CNN as a feature extractor has been widely explored, as seen in Novitasari et al.'s research, which applied various CNN architectures like Googlenet, Resnet18, Resnet50, and Resnet101 in the detection of COVID-19, pneumonia, and normal using SVM as the classification method and CLAHE. The Resnet50 architecture achieved an accuracy of 97.33% in separating the three classes, indicating that feature selection methods can significantly improve accuracy to around ±98% [10].

CNN training process is affected by slow convergence and the need for extensive parameter tuning [11]. This can be addressed by combining CNN for feature extraction and the Extreme Learning Machine (ELM) for image classification, known as Convolutional Extreme Learning Machine (CELM) [12]. ELM offers fast learning speed, high efficiency, and strong generalization capabilities, but it lacks in feature extraction. In contrast, CNN consumes a lot of time during the training process[13]. ELM's drawback is that the optimal number of neurons can only be obtained through trial and error, which can be overcome by adding a kernel, known as the KELM method. The KELM algorithm only requires determining the type and parameters of the kernel function. Moreover, KELM outperforms with fewer parameters and can avoid the 'dimension disaster' [14]. Pashaei et al. identified three brain tumor classes using CNN feature extraction with several classification methods. In the KELM classification method, MPL, Stacking, XGBoost, SVM, and RBF achieved accuracies of 93.68%, 88.80%, 86.91%, 87.33%, 87.51%, and 86.84%, respectively. The study's best method used CNN feature extraction with the KELM classification method, outperforming CNN, which achieved an accuracy of 81.09% [15].

The convolutional CNN features in image extraction, while KELM is known for its swift learning speed and high efficiency in the learning process. Hence, this study adopts the Hybrid CNN-KELM method, also known as the Convolutional Extreme Learning Machine with Kernel, to detect CXR images of pulmonary tuberculosis, pneumonia, COVID-19, and normal cases. The proposed method is anticipated to enhance accuracy and reduce processing time. Experimental assessments were conducted on CNN architectures, including ResNet18, ResNet50, ResNet101, GoogleNet, DenseNet201, AlexNet, MobileNet, NASNet-Mobile, and VGG19.

## II. RESEARCH METHOD

This research detects pulmonary tuberculosis, pneumonia, and COVID-19 using the CNN and KELM methods. The dataset for this study comprises 7135 CXR images obtained from Kaggle. The dataset is divided into four classes: 703 images of tuberculosis, 4273 images of pneumonia, 576 images of COVID-19, and 1583 images of normal [16]. The dataset suffers from data imbalance, necessitating random under sampling. In this study, only 2304 images were used, drawn from the four classes, each containing an equal number of 576 images. Figure 1 illustrates a set of CXR images representing these four classes.

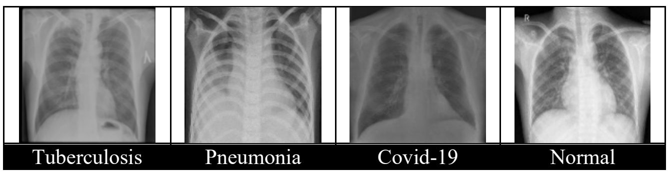



Figure 1. **Data Sample.**

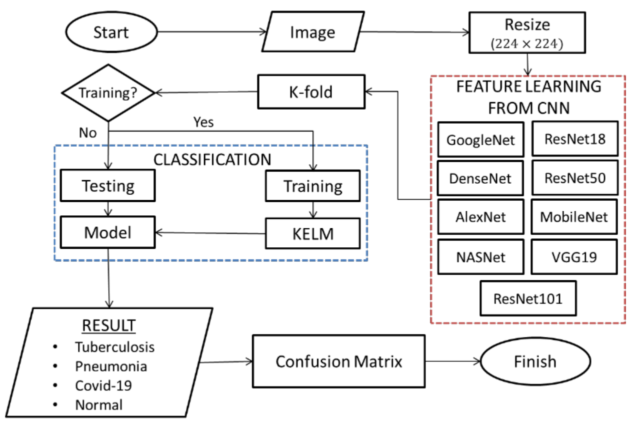



Figure 2. **Research Flowchart**

The image data preprocessing involves resizing the images to 224×224 pixels. Feature extraction is carried out using a CNN algorithm to transform the image data into numerical data. This study conducts experiments with several CNN architectures used in the feature learning process. These architectures include ResNet18, ResNet50, ResNet101, GoogleNet, DenseNet201, AlexNet, MobileNet, NASNet-Mobile, and VGG19. The extracted feature results are divided into training and testing data using K-Fold Cross Validation (K-Fold). Subsequently, the classification process is executed by applying the KELM method to the CNN feature extraction results. The training data undergoes a learning process using KELM to generate a model. This model is then utilized in the testing phase with the testing data. Model evaluation employs a confusion matrix to obtain accuracy, precision, and sensitivity values for this study. The research flow is illustrated in Figure 2.

### *A. VARIOUS ARCHITECTURE OF CONVOLUTION NEURAL NETWORK (CNN)*

The Convolutional Neural Network (CNN) is an advancement of the Multi-Layer Perceptron (MLP) method [17]. CNN comprises an input layer, convolutional layer, pooling layer, fully connected layer, and output layer. Based on the number of layers, CNN consists of 2 layers: the feature extraction layer and the Fully Connected Layer. CNN feature extraction utilizes the feature learning process of CNN located in the feature extraction layer without entering the classification process in the Fully Connected Layer [18]. The CNN architecture is depicted in Figure 3.

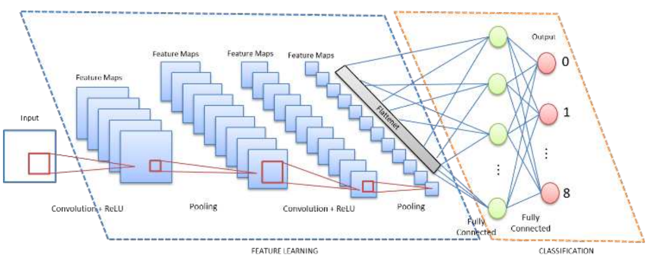



**Figure 3. Architecture of CNN**

The convolutional layer is used for feature extraction of images, and the convolution operation with filters in each local area of the image is used to learn spatial patterns. The convolutional layer contains multiple convolutional kernels. Convolutional kernels are weight matrices that can automatically learn characteristics from the input image [19]. The weights of convolutional kernels can be obtained through training. The shared parameter mechanism in each neuron shares the same kernel and weight, significantly reducing the computational complexity of training parameters, addressing overfitting issues, and enhancing the generalization ability of the CNN model to some extent. The output of the convolution process is called a feature map, which will be processed in the next layer [6]. The feature map consists of a series of neurons that are interconnected within the local receptive field of the previous feature map by the convolutional kernel [19].

GoogleNet have a module called Inception Module (IM) that can reduce data parameter. The core of the reduction process in the IM is to reduce the number of input channels, where in each inception module, a 1×1 convolution layer is added before the 3×3 and 5×5 convolution layers and after the 3×3 max-pooling layer. This can reduce the number of network parameters to support fewer computational operations [20]. The IM architecture is depicted in Figure 4, where the 1×1 filter, along with the max-pooling layer, performs the dual task of dimension reduction and content summarization from the previous layer. Figure 5 shows that the nine IMs in the GoogleNet architecture effectively extract more discriminative features from the source image [21].

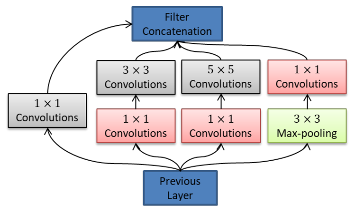



**Figure 4. Inception module with dimension reduction**

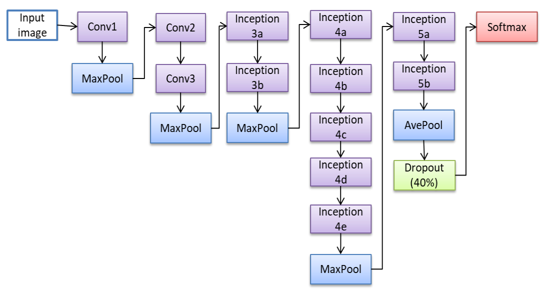



**Figure 5. GoogleNet Architecture**

The Residual Neural Network (ResNet) was proposed by Kaiming. The ResNet architecture was introduced in various variants in the ILSVRC-2015 competition, where it secured the first place with an error rate of 3.57%. The distinguishing feature of the ResNet architecture from other model architectures lies in residual learning [22]. In deep networks, the problem of degradation arises due to the increased depth of CNN, which leads to decreased accuracy. ResNet maintains its performance even as its architecture becomes deeper. Additionally, computations become more efficient, and the network's training capability improves. Figure 6 shows ResNet50, which includes a 7×7 convolution layer with 64 kernels, a 3×3 max-pooling layer with a stride of 2, 16 residual building blocks, a 7×7 average-pooling layer with a stride of 7, and a newly added fully connected layer before the softmax output layer. The variation in the ResNet layer structure lies in the residual blocks. ResNet101 has a total of 33 residual blocks [23].

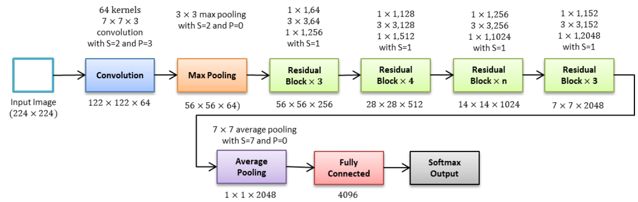



**Figure 6.** **Transfer learning architecture with modified ResNet50 and ResNet101 models (n*=6 for ResNet50 and n=23 for ResNet101; S-Stride and P-Padding).**

The Densely Connected Convolutional Network (DenseNet) was developed by Huang. In addition to addressing the vanishing-gradient problem and reducing the number of parameters, DenseNet also strengthens feature propagation and encourages feature reuse [24]. DenseNet includes dense blocks in its CNN architecture. Each dense layer has a transition layer that contains convolutional layers and pooling layers, as shown in Figure 7 [25].

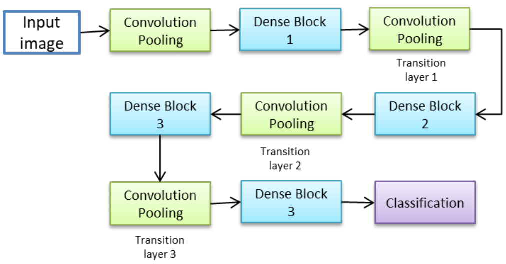



**Figure 7.** **DenseNet Architecture**

AlexNet uses input with dimensions of 227x227 and three channels. In Figure 8, AlexNet consists of 5 convolutional and 3 fully connected layers, and it employs ReLU activation functions in each convolutional layer. Max pooling is applied to several convolution layers. The first convolutional layer includes conv1 and pool1. Conv1 applies an 11x11 filter to process the input image, and pool1 uses a 3x3 filter. The third and fourth convolutional layers do not undergo pooling and normalization processes. Fc6, fc7, and fc8 are all fully connected layers, followed by softmax, and the classification layer contains 1000 neurons representing the number of classes [25].

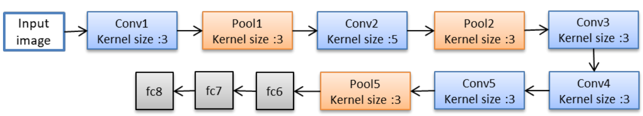



**Figure 8.** **AlexNet Architecture**

MobileNetV2 is specifically designed for computer vision applications in mobile and resource-constrained environments. MobileNetV2 outperforms MobileNetV1 with better performance and faster processing. This model introduces the inverted residual with a linear bottleneck module. Inverted residual connections refer to connections between thin blocks to facilitate gradient flow through the network. This is different from normal residual connections used to connect two expanded units. Inverted residual blocks produce far fewer parameters than standard residual blocks. There are several convolutional blocks with skip-connections [26]. The full architecture of MobileNet V2 consists of 17 consecutive bottleneck residual blocks, followed by regular 1×1 convolution, a global average pooling layer, and a classification layer, as shown in Table 1 [27].

TABLE I
ARCHITECTURE MOBILENETV2

| Input | Operator | t | c | n | s |
|---|---|---|---|---|---|
| 224×224×3 | Convolution 2D | - | 32 | 1 | 2 |
| 112×112×32 | bottleneck | 1 | 16 | 1 | 1 |
| 112×112×16 | bottleneck | 6 | 24 | 2 | 2 |
| 56×56×24 | bottleneck | 6 | 32 | 3 | 2 |
| 28×28×32 | bottleneck | 6 | 64 | 4 | 2 |
| 14×14×64 | bottleneck | 6 | 96 | 3 | 1 |
| 14×14×96 | bottleneck | 6 | 160 | 3 | 2 |
| 7×7×160 | bottleneck | 6 | 320 | 1 | 1 |
| 7×7×230 | Convolution 2D 1 × 1 | - | 1280 | 1 | 1 |
| 7×7×1280 | Average Pooling 7 × 7 | - | - | 1 | - |
| 1×1×1280 | Convolution 2D 1 × 1 | - | k | - | - |

Neural Architecture Search Network (NASNet), developed by Google Brain in 2018, achieved high performance on the CIFAR10 dataset. NASNet is designed to search for and select optimized CNN architectures using Reinforcement Learning (RL). The fundamental concept is to discover the optimal combinations of kernel size, filter size, output channels, strides, the number of layers, and other characteristics within a given search space [28]. This architecture seeks the best convolutional layers by designing controllers based on Recurrent Neural Networks (RNNs) to find the most optimal combinations. The best convolutional layers are referred to as cells, which are combinations of blocks. Convolutional cells include normal and reduction cells, where normal cells return feature maps with the same dimensions, while reduction cells return reduced feature maps [29]. The NASNet architecture can be seen in Figure 9.

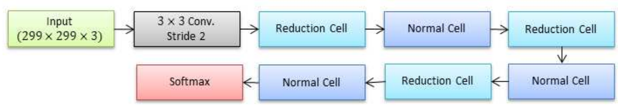



**Figure 9.** **NasNet Architecture**

VGG (Visual Geometry Group) Net was introduced by researchers at the University of Oxford. VGG is an extension of AlexNet, with an increased number of convolutional CNN layers and a change in filter size to 3x3 for all convolution layers. VGG architecture secured second position in the ILSVRC competition in 2014, with a training error rate of 7.30%, surpassing the earlier AlexNet [20]. VGGNet consists of five blocks of convolutional processing, with each block having 2-3 convolution layers with ReLU activation. Each block is connected to a 2 × 2 max-pooling layer to reduce the image size. This is followed by fully connected layers, namely, fc6, fc7, and fc8, and then the softmax layer as the output layer. The main difference between the VGG19 and VGG16 models is in blocks 3, 4, and 5, where each block has four 3 × 3 convolution layers. Figure 12 below shows the structure of VGGNet16, which comprises thirteen convolution layers, three fully connected layers, and five max-pooling layers [23]. VGG architecture shown in Figure 10.

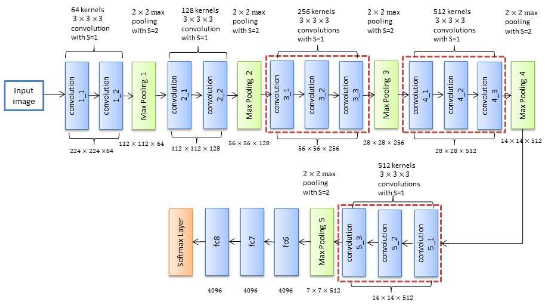



**Figure 10.** **VGGNet16 Architecture**

### *B. KERNEL EXTREME LEARNING MACHINE (KELM)*

Kernel Extreme Learning Machine (KELM) is an extension of the ELM algorithm that incorporates kernel function. The kernel is used to map data to a higher-dimensional space. The KELM algorithm only requires determining the type and parameters of the kernel function, resulting in fewer parameters for KELM [14]. Moreover, KELM not only inherits the generalization performance advantages of ELM but can also obtain solutions for least-squares optimization due to the introduction of kernel functions. As a result, KELM exhibits more stable and superior performance [30].

$$f(x) = h(x)\beta \tag{1}$$

In (1), where $x$ represents the sample, $h(x)$ is the output matrix from the hidden layer, and $\beta$ is the weight between the hidden layer and the output layer. According to the general matrix inverse theory, the least-squares solution for the output weights can be obtained as shown in (2).

$$\beta = \boldsymbol{H}^T\left(\frac{I}{C} + \boldsymbol{HH}^T\right)^{-1}\boldsymbol{T} \tag{2}$$

Where H is the hidden layer output matrix, $C$ is the regularization parameter, $I$ is the identity matrix, and $T$ represents the output vector from the training set. The ELM output function is represented in (3).

$$f(x)_{ELM} = h(x)\left(\boldsymbol{H}^T\left(\frac{I}{C} + \boldsymbol{HH}^T\right)^{-1}\boldsymbol{T}\right) \tag{3}$$

Considering that the feature mapping function $h(x)$ is unknown, a kernel function needs to be incorporated into ELM. Determine the kernel matrix $\Omega_{KELM} = \boldsymbol{HH}^T\sqrt{a^2+b^2}$ where the elements of the matrix are $\Omega_{KELM}(i,j) = h(x_i)h(x_j) = K(x_i, x_j)$. Then, the expression of the output function obtained from the weight formula and the kernel function expression is shown in (4).

$$f(x)_{KELM} = h(x)\left(\boldsymbol{H}^T\left(\frac{I}{C} + \boldsymbol{HH}^T\right)^{-1}\boldsymbol{T}\right) \tag{4}$$

$$= \begin{bmatrix} K(x, x_1) \\ \vdots \\ K(x, x_N) \end{bmatrix}\left(\frac{I}{C} + \Omega_{KELM}\right)^{-1}\boldsymbol{T}$$

Among several kernel functions, the Radial Basis Function (RBF) is one of the most commonly used due to its wide convergence domain and good generalization ability. The RBF kernel can be defined in (5).

$$K(x, x_i) = \exp\left(-\gamma\, \|x - x_i\|^2\right) \tag{5}$$

With $\gamma$ as a kernel parameter.

## III. RESULTS AND DISCUSSION

In this research, classification is conducted for four classes, namely tuberculosis, pneumonia, COVID-19, and normal. The dataset used in this study has a class imbalance issue, which causes algorithms to favor the majority class. This class imbalance issue can be addressed using resampling methods. Under sampling is one type of resampling method that works by removing some instances from the majority class to balance the sample distribution. Sample data shown in Figure 1. Under sampling was applied because the dataset has a highly imbalanced data distribution and improves the model's ability to classify the minority class while reducing bias and overfitting towards the majority class.

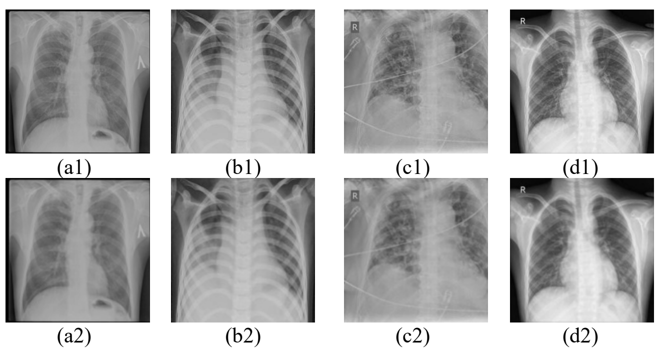



**Figure 11.** **Preprocessing CRX image**

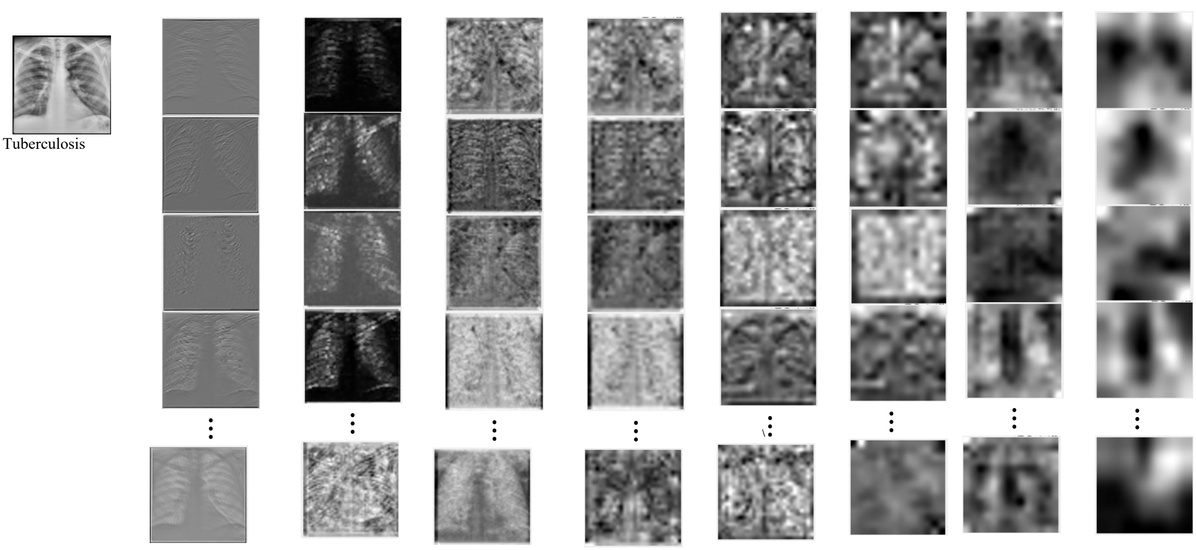
In the preprocessing stage, the image size is resized to 224×224 pixels, except for the AlexNet architecture, where the image size is resized to 227×227 pixels. The preprocessing results are shown in Figure 11, where a represents tuberculosis, b represents pneumonia, c represents COVID-19, and d represents normal. The first row displays the results of the image resizing process. Subsequently, the image's channel size, which initially has a single channel, is adjusted to three channels (Red, Green, Blue) to match the CNN input requirements. This is achieved by stacking the single-channel images with identical values, resulting in an image size of 224×224×3 as shown in the second row.

The preprocessing results were followed by feature extraction using CNN. During the feature extraction stage, a comparison of various CNN architectures was conducted. The CNN architectures used included ResNet18, ResNet50, ResNet101, GoogleNet, DenseNet201, AlexNet, MobileNet, NASNet-Mobile, and VGG19. The CNN feature extraction process involves convolution, pooling, and the application of activation functions. In Figure 7, the DenseNet architecture consists of several dense blocks and transition layers that contain convolutional and pooling layers. The feature map results for each block of the DenseNet architecture are shown in Table 3. As indicated in Table 2, the feature extraction from the same image produces different values and features for each CNN architecture.

TABLE 2
CNN ARCHITECTURE FEATURE EXTRACTION RESULTS

| No Feature | ResNet-18 | ResNet-50 | ResNet-101 | GoogleNet | DenseNet-201 | AlexNet | MobileNet-V2 | NASNet-Mobile | VGG-19 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1.3378 | 1.1685 | 0.0432 | 0.0027 | 0.0001 | 0.0000 | 0.1689 | 0.6284 | 0.0000 |
| 2 | 1.8762 | 0.6434 | 0.6293 | 0.7441 | 0.0005 | 9.2646 | 0.4194 | 0.2099 | 0.0000 |
| 3 | 0.7696 | 0.0149 | 0.0112 | 0.2408 | 0.0003 | 9.2646 | 0.1836 | 0.0919 | 0.0000 |
| 4 | 2.3618 | 0.0962 | 0.0190 | 0.0000 | 0.0002 | 5.2194 | 0.0000 | 0.4745 | 3.3858 |
| 5 | 1.4194 | 2.1443 | 0.7246 | 0.2193 | 0.0003 | 0.6681 | 0.0013 | 0.4729 | 2.9558 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| $n^{th}$ feature | 1.4564 | 0.0000 | 0.1569 | 0.2199 | 1.9817 | 0.0000 | 0.5343 | 0.0040 | 0.0000 |
| Total Features (n) | 512 | 2048 | 2048 | 1024 | 1920 | 9216 | 1280 | 1056 | 25088 |

TABLE 3
FEATURE MAP RESULTS OF EACH DENSENET-201 LAYER

| Input (224 × 224 × 3) | 7 × 7 Conv (112 × 112, 64) | Dense Blok 1 (56 × 56, 256) | Conv (56 × 56, 128) | Dense Blok 2 (28 × 28, 512) | Conv (28 × 28, 256) | Dense Blok 3 (14 × 14, 1792) | Conv (14 × 14, 896) | Dense Blok4 (7 × 7, 1920) | Global Average Pooling (1 × 1, 1920) |
|---|---|---|---|---|---|---|---|---|---|
|  Tuberculosis | | | | | | | | | -0.00006<br>0.00007<br>-0.00050<br>0.00371<br>0.00055<br>0.00034<br>-0.00024<br>0.00014<br>-0.00012<br>0.01619<br>0.00105<br>0.00087<br>0.00001<br>0.00162<br>0.00071<br>-0.00001<br>-0.00038<br>-0.00018<br>-0.00097<br>-0.00050<br>-0.00050<br>⋮<br>2.91738 |

Next, the feature extraction results were divided into training and testing data using k-fold cross-validation. The data was split using K-Fold with k=5, resulting in 1843 data points for training and 461 data points for testing. Classification was then performed using KELM. The combination of CNN feature extraction with KELM classification is known as the CKELM method. In the KELM method, a comparison of various kernel types was conducted, including the Radial Basis Function (RBF), linear, polynomial, and wavelet kernels. Additionally, a comparison of regularization coefficients for KELM, specifically 0.1, 1, 10, 100, and 1000, was performed. Model evaluation was carried out using a confusion matrix based on accuracy, sensitivity, specificity, precision, F1-score, and training duration. The purpose of this evaluation was to obtain an optimal model with a shorter training time. The results of the hybrid CNN-KELM experiments are presented in Figures 12-20.

Based on Figure 12, the results show that the highest average accuracy was achieved using the CKELM method with the ResNet18 architecture and a polynomial kernel, which reached 98.26%. On the other hand, the lowest average accuracy was observed with the wavelet kernel at 50.87%. The average accuracy with the wavelet kernel had a significant difference from the other three kernels. The experiments with regularization coefficients for each kernel showed that the RBF, polynomial, and wavelet kernels had the same accuracy values. However, the linear kernel had relatively small differences in accuracy. The highest accuracy for the linear kernel was achieved with a regularization coefficient of 1, reaching 97.17%, while the lowest accuracy was recorded with coefficients of 10, 100, and 1000, each at 96.96%.

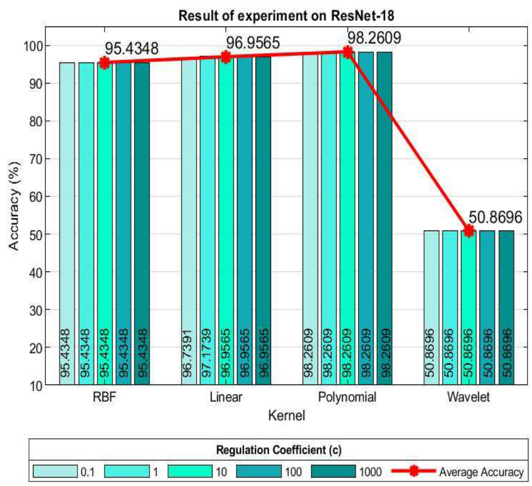



Figure 12. **ResNet18 result with the KELM classification method**

Figure 13 represents the results of CKELM with ResNet50 feature extraction, showing that the highest average accuracy was obtained with the polynomial kernel at 98.05%, while the lowest average accuracy was associated with the wavelet kernel at 47.51%. Experimenting with the regularization coefficients for the linear kernel revealed significant differences in accuracy. The linear kernel with a regularization coefficient of 0.1 achieved the highest accuracy at 97.81%, while the lowest accuracy for this kernel was observed with a coefficient of 1000, reaching 78.96%. Increasing the value of the regularization coefficient led to lower accuracy in ResNet-50 with the linear kernel.

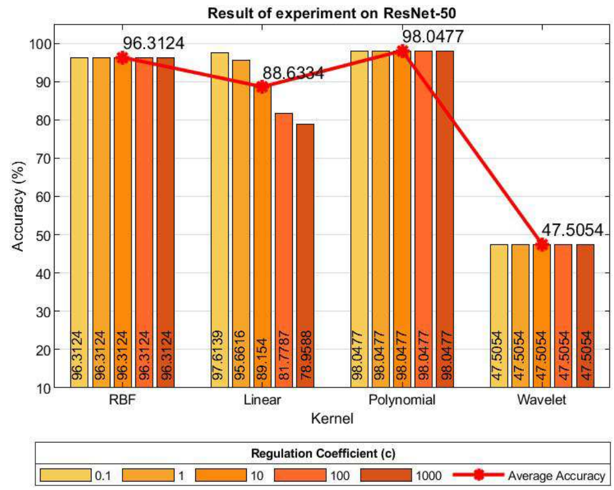



Figure 13. **ResNet50 result with the KELM classification method**

Based on Figure 14, it is evident that the accuracy of the CKELM model using ResNet101 feature extraction varied. The highest average accuracy was observed with the polynomial kernel, reaching 96.74%, while the lowest average accuracy was linked to the wavelet kernel. For the linear kernel, the experiments with different regularization coefficients had a notable impact on the accuracy achieved. The highest accuracy with this kernel was obtained with a coefficient of 0.1, while the lowest was associated with coefficients of 1000. As for the RBF, polynomial, and wavelet kernels, their accuracy remained constant across different regularization coefficient experiments.

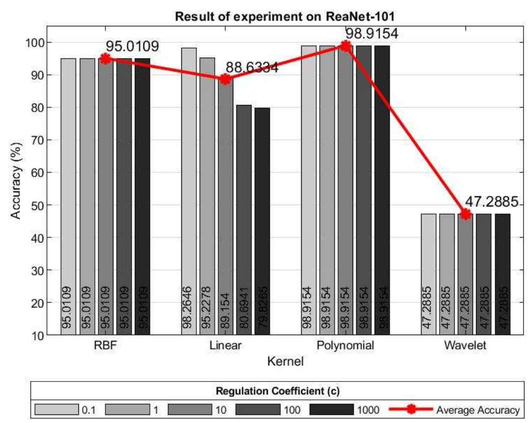



Figure 14. **ResNet101 result with the KELM classification method**

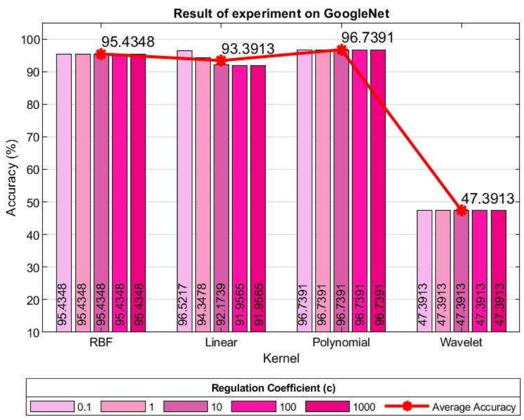



**Figure 15.** **GoogleNet result with the KELM classification method**

Figure 15 presents the results of CKELM with the GoogleNet architecture. In this figure, the highest average accuracy was achieved with the polynomial kernel at 96.74%, while the lowest average accuracy was observed with the wavelet kernel at 47.39%. The experiments involving regularization coefficients had the same accuracy values for the RBF, polynomial, and wavelet kernels. However, in the case of the linear kernel, the accuracy varied across different regularization coefficient experiments. The highest accuracy for this kernel was attained with a coefficient of 0.1, while the lowest accuracy was associated with coefficients of 100 and 1000. In both the ResNet101 and GoogleNet architectures, it can be concluded that increasing the regularization coefficient leads to a decrease in accuracy.

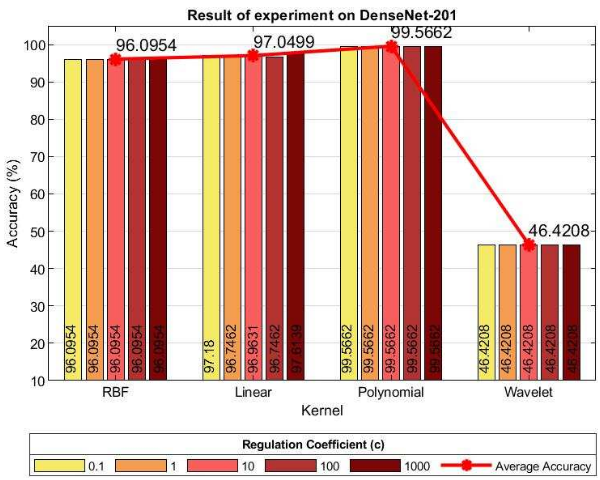



**Figure 16.** **DenseNet-201 result with the KELM classification method**

Based on Figure 16, it's evident that the highest average accuracy in the CKELM architecture with DenseNet201 was achieved with the polynomial kernel at 99.57%, while the lowest average accuracy was linked to the wavelet kernel at 46.42%. The accuracy values for the linear kernel varied with different regularization coefficients, but the accuracy of the other three kernels remained consistent across various regularization coefficient experiments


**Figure 17.** **AlexNet result with the KELM classification method**

Figure 17 presents the results of CKELM with the AlexNet architecture. In this figure, the highest average accuracy was obtained with the polynomial kernel at 98.26%, while the lowest average accuracy was associated with the wavelet kernel at 26.25%. The RBF and wavelet kernels exhibited significantly lower accuracy compared to the linear and polynomial kernels. The experiments with different regularization coefficients for each kernel yielded consistent accuracy results, indicating that there were no significant differences in accuracy across the kernels.

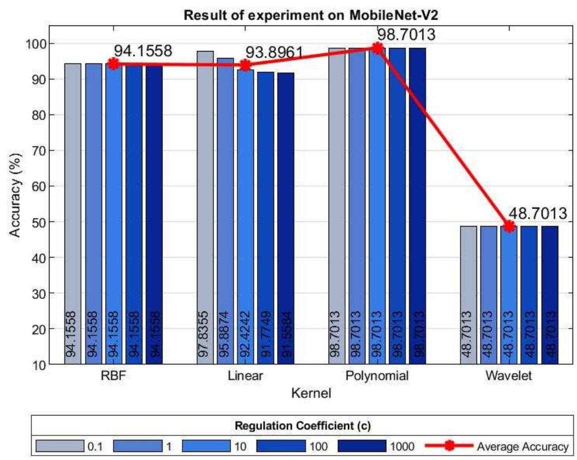



**Figure 18.** **MobileNet result with the KELM classification method**

Based on Figure 18, the highest average accuracy for the CKELM architecture with MobileNet-V2 was achieved with the polynomial kernel at 98.70%, while the lowest average accuracy was linked to the wavelet kernel at 46.97%. The linear kernel showed varying accuracy levels in each regularization coefficient experiment, with higher regularization coefficients resulting in lower accuracy. The best accuracy for the linear kernel was at a regularization coefficient of 0.1, reaching 97.84%.

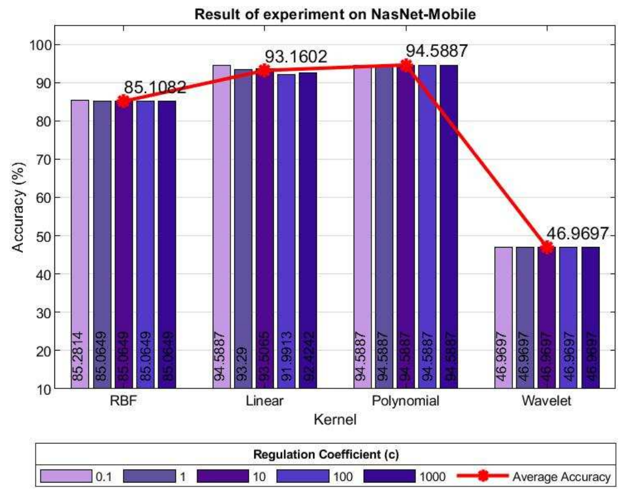



**Figure 19.** **NASNet result with the KELM classification method**

In Figure 19, the results show the highest average accuracy with the polynomial kernel at 94.59%, and the lowest average accuracy was associated with the wavelet kernel at 46.97%. The accuracy results for each regularization coefficient in the linear kernel experiments varied, while the other three kernels exhibited consistent accuracy values across different regularization coefficient tests. The best accuracy for the linear kernel was achieved with a regularization coefficient of 0.1, amounting to 94.59%.

In Figure 20, the accuracy results for CKELM with the VGG19 architecture were presented. The best experiment results were obtained with the linear kernel at 99.57%, while the lowest accuracy was associated with the RBF and wavelet kernels at 25.43%. The regularization coefficient experiments yielded the same accuracy for all kernels, indicating that the regularization coefficient did not affect the accuracy of any of the kernels.

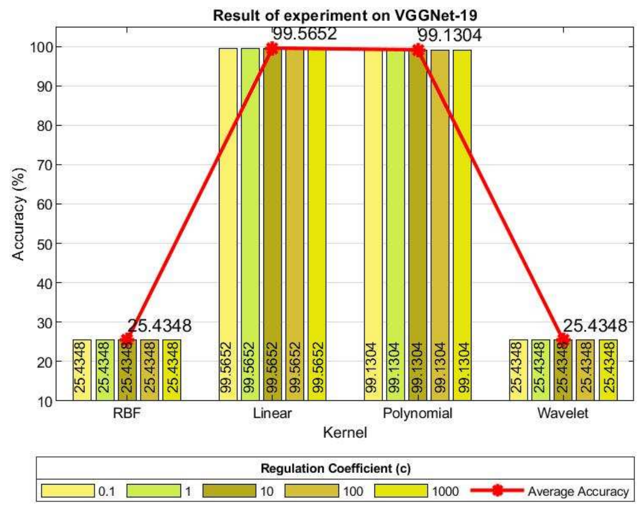



**Figure 20.** **VGG19 result with the KELM classification method**

Based on the overall experiments, it can be concluded that accuracy tends to be high with the polynomial kernel in all CNN architectures. Conversely, the wavelet kernel yielded poor accuracy, generally below 50%. A regularization coefficient of 0.1 typically resulted in higher accuracy compared to other regularization coefficients in all experiments involving CNN architectures and KELM kernels. The accuracy of the linear kernel was influenced by the regularization coefficient for each CNN architecture.

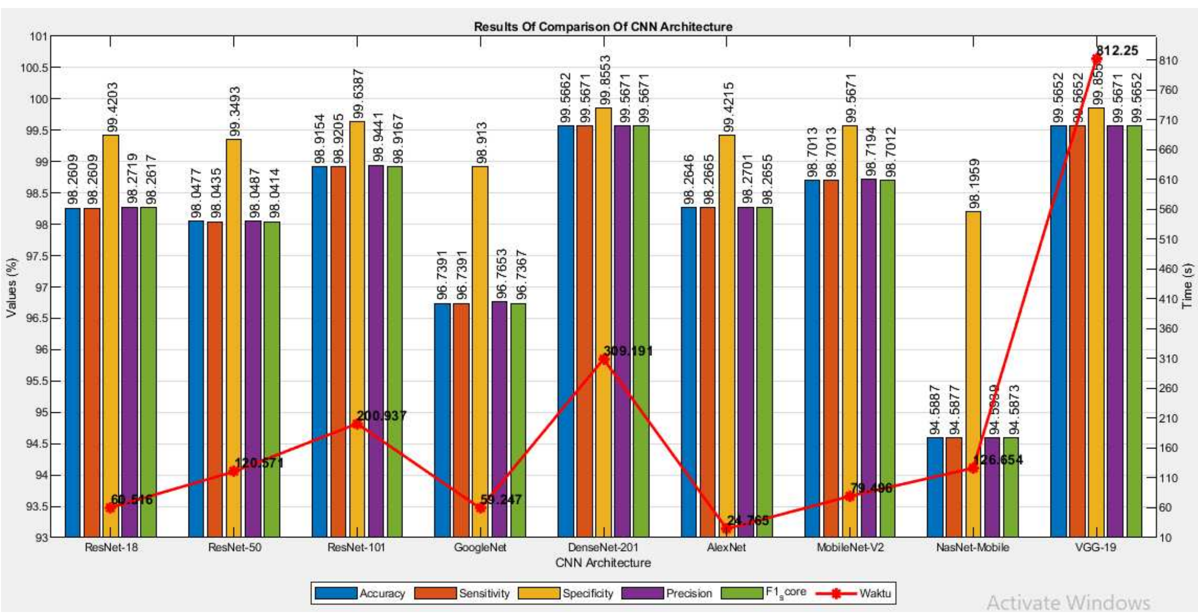



**Figure 21.** **Comparison of CNN feature extraction with KELM classification**

The confusion matrix for DenseNet201-KELM is shown in Figure 22. Labels 1, 2, 3, and 4 represent the categorical data for COVID-19, normal, pneumonia, and tuberculosis, respectively. The image indicates that the normal and pneumonia classes were correctly identified in their respective categories. This demonstrates that the DenseNet201-KELM model can classify both of these classes effectively. However, for the COVID-19 and tuberculosis classes, one image was incorrectly classified in each case. The class that should have been COVID-19 was misclassified as tuberculosis, and vice versa. The class that should have been tuberculosis was classified as COVID-19.

A comparison of previous studies conducted classification of tuberculosis, pneumonia, COVID-19, and normal is presented in Table 4. Study [32] employed AlexNet architecture and conducted tests with various resize image dimensions, learning rates, epoch values, batch sizes, and optimizers. The best results were obtained with image size 64x64 pixels, learning rate 0.0001, epoch 25, batch size 32, and the Adam optimizer, resulting in an accuracy of 95%. Study [33] proposed a model that utilized a single CNN based on XAI (Explainable AI). XAI algorithms including SHAP, LIME, and Grad were used to explore black-box model approaches for deep learning. The average accuracy achieved with 10-fold cross-validation was 94.54%. Study [34] employed a CNN to detect the four classes, achieving an accuracy of 94.53%.

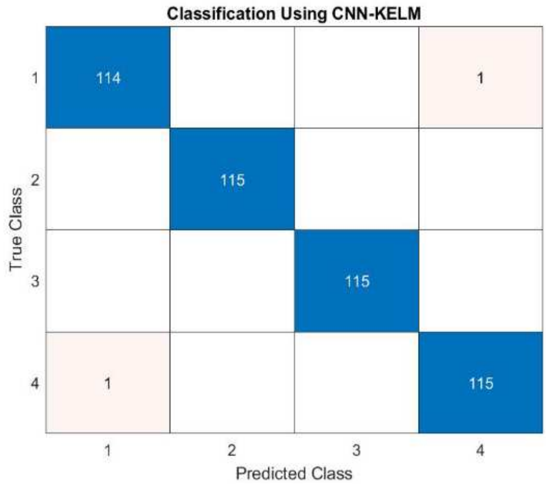



Figure 22. **Confusion matrix on DenseNet201-KELM**

TABLE 4
COMPARISON OF THE PROPOSED MODELS USING CXR IMAGE FOR CLASSIFICATION OF FOUR CLASSES

| **Method** | **Cases** | Accuracy (%) | Sensitivity/recall (%) | **Specificity (%)** | **f1-score** |
|---|---|---|---|---|---|
| CNN architecture Alexnet [31] | Normal = 576 Covid19 = 576 Pneumonia = 576 TB = 576 | 95% | 95% | 95% | 95% |
| Custom CNN and XAI [32] | Normal = 1583 Covid19 = 576 Pneumonia = 4273 TB = 700 | 94.54 % | - | - | - |
| CNN [33] | Normal = 1583 Covid19 = 576 Pneumonia = 4273 TB = 155 | 94.53% | - | - | - |
| DenResCov-19 [34] | COVID-19 = 69 pneumonia = 300 tuberculosis =310 Normal =330 | - | - | - | 83.17% |
| CNN VGG16 model with augmentation [35] | Covid-19, pneumonia, tuberculosis, and normal 1000 each | 86% | - | - | - |
| CNN model Densenet-169 with augmentation [35] | Covid-19, pneumonia, tuberculosis, and normal 1000 each | 91% | - | - | - |
| **Proposed method CKELM (DenseNet-KELM )** | Normal = 1583 Covid19 = 576 Pneumonia = 4273 TB = 700 without under sampling | 92.00% | 93.15% | 97.87% | 91.45% |
| | Normal = 576 Covid19 = 576 Pneumonia = 576 TB = 576 with under sampling | **99.57%** | **99.57%** | **99.86%** | **99.57%** |

In Study [35], a novel deep transfer learning method called DenResCov-19 was developed. This method involved modifying a CNN with additional layers that combined DenseNet121 and ResNet50 networks. The model achieved an F1-score of 83.17%. Data augmentation was performed on images of tuberculosis, pneumonia, COVID-19, and normal, resulting in 1000 images per class. Classification of these augmented images led to an accuracy of 86% using VGG16 and 91% using DenseNet169. The proposed model in this study showed superior performance compared to previous research. The method used in this study achieved an accuracy of 99.57% for under-sampled data, which is 7% better than without under-sampled sampling. Furthermore, the required computational time was relatively short, approximately 5 minutes. These results demonstrate that the CKELM method yields good classification performance and efficient training times. Future research could focus on implementing data augmentation to address class imbalance issues. Additionally, further studies may explore the use of hybrid CNN-MLELM for handling larger datasets to optimize class classification while improving time efficiency [18].

## IV. CONCLUSION

This study focused on classifying four lung diseases: tuberculosis, pneumonia, COVID-19, and normal. Preprocessing involved resizing images to match CNN input sizes, while a CKELM approach was employed for multiclass classification, using CNN for feature extraction and KELM for classification. Various CNN architectures were explored, including ResNet18, ResNet50, ResNet101, GoogleNet, DenseNet201, AlexNet, MobileNet, NASNet-mobile, and VGG19. Different KELM kernels (RBF, linear, polynomial, wavelet) and regularization coefficients (0.1, 1, 10, 100, 1000) were tested. The best results achieved 99.57% accuracy using the DenseNet-201 architecture, polynomial kernel, and a regularization coefficient of 0.1. This result is 7% better than without under sampling. KELM is highly sensitive to data distribution, and DenseNet, as a feature extractor, requires balanced representations to generate discriminative features. Polynomial kernels are able to achieve a better balance between model complexity and generalization ability than other kernels. Overall, this study found that the DenseNet-201-KELM hybrid method performed best in multiclass data separation and balanced data, despite longer computational times due to architectural depth.

## ACKNOWLEDGMENTS

We extend our gratitude to the Mathematics Study Program, UIN Sunan Ampel Surabaya, for their support.

## AUTHORS CONTRIBUTION

**Dian Candra Rini Novitasari:** Validating the experiments and Supervision;
**Musfiroh Musfiroh:** Writing, Visualization, Experimenting, Review & Editing Writing, and Testing Research;
**Dina Zatusiva Haq:** Translator, Review Writing &; Editing;



## REFERENCES

[1] A. Rachmad, N. Chamidah, and R. Rulaningtyas, "Mycobacterium tuberculosis images classification based on combining of convolutional neural network and support vector machine," *Commun. Math. Biol. Neurosci.*, vol. 2020, pp. 1–13, 2020, doi: 10.28919/cmbn/5035.

[2] F. Djannah, M. N. Massi, M. Hatta, A. Bukhari, and I. Hasanah, "Profile and histopathology features of top three cases of Extra Pulmonary Tuberculosis (EPTB) in West Nusa Tenggara: A retrospective cross-sectional study," *Ann. Med. Surg.*, vol. 75, no. 37, p. 103318, 2022, doi: 10.1016/j.amsu.2022.103318.

[3] S. Soedarsono *et al.*, "Development of population pharmacokinetics model of isoniazid in Indonesian patients with tuberculosis," *Int. J. Infect. Dis.*, vol. 117, pp. 8–14, 2022, doi: 10.1016/j.ijid.2022.01.003.

[4] K. kesehatan RI, "Tubercolusis Indonesia," 2023. https://tbindonesia.or.id/ (accessed Mar. 24, 2023).

[5] Z. Z. Qin *et al.*, "A new resource on artificial intelligence powered computer automated detection software products for tuberculosis programmes and implementers," *Tuberculosis*, vol. 127, p. 102049, 2021, doi: 10.1016/j.tube.2020.102049.

[6] D. C. R. Novitasari *et al.*, "Image Fundus Classification System for Diabetic Retinopathy Stage Detection Using Hybrid CNN-DELM," *Big Data Cogn. Comput.*, vol. 6, no. 4, 2022, doi: 10.3390/bdcc6040146.

[7] A. Z. Foeady, S. R. Riqmawatin, and D. C. R. Novitasari, "Lung cancer classification based on CT scan image by applying FCM segmentation and neural network technique," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 19, no. 4, pp. 1284–1290, 2021, doi: 10.12928/TELKOMNIKA.v19i4.18874.

[8] I. F. Sudirman, W. H. Giawa, I. P. Sarumaha, S. Ndraha, and I. Fawwaz, "Linear Kernel and Polynomial Analysis in Recognizing Tuberculosis Image Using HOG Feature Extraction," *J. Mantik*, vol. 4, no. 3, pp. 1693–1698, 2020.

[9] I. A. Ahmed, E. M. Senan, H. Salameh, A. Shatnawi, Z. M. Alkhraisha, and M. M. A. Al-azzam, "Multi-Techniques for Analyzing X-ray Images for Early Detection and Differentiation of Pneumonia and Tuberculosis Based on Hybrid Features," 2023.

[10] D. C. R. Novitasari *et al.*, "Detection of COVID-19 chest x-ray using support vector machine and convolutional neural network," *Commun. Math. Biol. Neurosci.*, vol. 2020, pp. 1–19, 2020, doi: 10.28919/cmbn/4765.

[11] Y. Park and H. S. Yang, "Convolutional neural network based on an extreme learning machine for image classification," *Neurocomputing*, vol. 339, no. 2019, pp. 66–76, 2019, doi: 10.1016/j.neucom.2018.12.080.

[12] H. Surrisyad and Wahyono, "A Fast Military Object Recognition using Extreme Learning Approach on CNN," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 12, pp. 210–220, 2020, doi: 10.14569/IJACSA.2020.0111227.

[13] S. Ding, L. Guo, and Y. Hou, "Extreme learning machine with kernel model based on deep learning," *Neural Comput. Appl.*, vol. 28, no. 8, pp. 1975–1984, 2017, doi: 10.1007/s00521-015-2170-y.

[14] C. Hou, Y. Li, X. Chen, and J. Zhang, "Automatic modulation classification using KELM with joint features of CNN and LBP," *Phys. Commun.*, vol. 45, p. 101259, 2021, doi: 10.1016/j.phycom.2020.101259.

[15] A. Pashaei, H. Sajedi, and N. Jazayeri, "Brain tumor classification via convolutional neural network and extreme learning machines," *2018 8th Int. Conf. Comput. Knowl. Eng. ICCKE 2018*, no. Iccke, pp. 314–319, 2018, doi: 10.1109/ICCKE.2018.8566571.

[16] K. Dataset, "Chest X-Ray (Pneumonia,Covid-19,Tuberculosis)." https://www.kaggle.com/datasets/jtiptj/chest-xray-pneumoniacovid19tuberculosis (accessed Apr. 06, 2023).

[17] A. M. Ayalew, A. O. Salau, B. T. Abeje, and B. Enyew, "Detection and classification of COVID-19 disease from X-ray images using convolutional neural networks and histogram of oriented gradients," *Biomed. Signal Process. Control*, vol. 74,

no. October 2021, p. 103530, 2022, doi: 10.1016/j.bspc.2022.103530.

[18] D. Candra, R. Novitasari, and R. E. Putra, "An Effective Hybrid Convolutional-Modified Extreme Learning Machine in Early Stage Diabetic Retinopathy," *Int. J. Intell. Eng. Syst.*, vol. 16, no. 2, pp. 401–413, 2023, doi: 10.22266/ijies2023.0430.32.

[19] X. Sun, L. Liu, C. Li, J. Yin, J. Zhao, and W. Si, "Classification for Remote Sensing Data with Improved CNN-SVM Method," *IEEE Access*, vol. 7, pp. 164507–164516, 2019, doi: 10.1109/ACCESS.2019.2952946.

[20] E. Suryawati, R. Sustika, R. S. Yuwana, A. Subekti, and H. F. Pardede, "Deep structured convolutional neural network for tomato diseases detection," *2018 Int. Conf. Adv. Comput. Sci. Inf. Syst. ICACSIS 2018*, pp. 385–390, 2019, doi: 10.1109/ICACSIS.2018.8618169.

[21] L. Balagourouchetty, J. K. Pragatheeswaran, B. Pottakkat, and G. Ramkumar, "GoogLeNet-Based Ensemble FCNet Classifier for Focal Liver Lesion Diagnosis," *IEEE J. Biomed. Heal. Informatics*, vol. 24, no. 6, pp. 1686–1694, 2020, doi: 10.1109/JBHI.2019.2942774.

[22] I. Ali, M. Muzammil, I. U. Haq, A. A. Khaliq, and S. Abdullah, "Deep Feature Selection and Decision Level Fusion for Lungs Nodule Classification," *IEEE Access*, vol. 9, pp. 18962–18973, 2021, doi: 10.1109/ACCESS.2021.3054735.

[23] K. Thenmozhi and U. Srinivasulu Reddy, "Crop pest classification based on deep convolutional neural network and transfer learning," *Comput. Electron. Agric.*, vol. 164, no. June, p. 104906, 2019, doi: 10.1016/j.compag.2019.104906.

[24] A. Martynenko and F. J. Verbeek, "Detection of mulberry ripeness stages using deep learning models," *IEEE Access*, vol. 9, pp. 100380–100394, 2021, doi: 10.1109/ACCESS.2021.3096550.

[25] D. U. N. Qomariah, H. Tjandrasa, and C. Fatichah, "Classification of diabetic retinopathy and normal retinal images using CNN and SVM," *Proc. 2019 Int. Conf. Inf. Commun. Technol. Syst. ICTS 2019*, pp. 152–157, 2019, doi: 10.1109/ICTS.2019.8850940.

[26] B. Singh, D. Toshniwal, and S. K. Allur, "Shunt connection: An intelligent skipping of contiguous blocks for optimizing MobileNet-V2," *Neural Networks*, vol. 118, pp. 192–203, 2019, doi: 10.1016/j.neunet.2019.06.006.

[27] A. Michele, V. Colin, and D. D. Santika, "Mobilenet convolutional neural networks and support vector machines for palmprint recognition," *Procedia Comput. Sci.*, vol. 157, pp. 110–117, 2019, doi: 10.1016/j.procs.2019.08.147.

[28] M. Mehmood *et al.*, "Improved colorization and classification of intracranial tumor expanse in MRI images via hybrid scheme of Pix2Pix-cGANs and NASNet-large," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 7, pp. 4358–4374, 2022, doi: 10.1016/j.jksuci.2022.05.015.

[29] J. Zhai, W. Shen, I. Singh, T. Wanyama, and Z. Gao, "A review of the evolution of deep learning architectures and comparison of their performances for histopathologic cancer detection," *Procedia Manuf.*, vol. 46, no. 2019, pp. 683–689, 2020, doi: 10.1016/j.promfg.2020.03.097.

[30] G. Li, C. Zheng, and H. Yang, "Carbon price combination prediction model based on improved variational mode decomposition," *Energy Reports*, vol. 8, pp. 1644–1664, 2022, doi: 10.1016/j.egyr.2021.11.270.

[31] S. Septhyan, R. Magdalena, and N. K. C. Pratiwi, "Deep Learning Untuk Deteksi Covid-19 , Pneumonia , Dan Tuberculosis Pada Citra Rontgen Dada Menggunakan CNN Dengan Arsitektur Alexnet," vol. 8, no. 6, pp. 2869–2878, 2022.

[32] M. Bhandari, T. B. Shahi, B. Siku, and A. Neupane, "Explanatory classification of CXR images into COVID-19, Pneumonia and Tuberculosis using deep learning and XAI," *Comput. Biol. Med.*, vol. 150, p. 106156, 2022, doi: https://doi.org/10.1016/j.compbiomed.2022.106156.

[33] N. N. Qaqos and O. S. Kareem, "COVID-19 Diagnosis from Chest X-ray Images Using Deep Learning Approach," *3rd Int. Conf. Adv. Sci. Eng. ICOASE 2020*, pp. 110–116, 2020, doi: 10.1109/ICOASE51841.2020.9436614.

[34] M. Mamalakis *et al.*, "DenResCov-19: A deep transfer learning network for robust automatic classification of COVID-19, pneumonia, and tuberculosis from X-rays," *Comput. Med. Imaging Graph.*, vol. 94, p. 102008, 2021, doi: https://doi.org/10.1016/j.compmedimag.2021.102008.

[35] K. Nair, A. Deshpande, R. Guntuka, and A. Patil, "Analysing X-Ray Images to Detect Lung Diseases Using DenseNet-169 technique," *SSRN Electron. J.*, 2022, doi: 10.2139/ssrn.4111864.