[1]D. Vetrithangam

[2]R. Himabindu

[3]S. Saranya

[4]Neha

[5]Naresh Kumar Pegada

[6]Azra Fathima

[7]Ashok Bekkanti

[8]Akanksha Kulkarni

Improved ResNet Models for Chronic Kidney Disease Prediction

Abstract: - Chronic kidney disease (CKD), a consequential health issue that can deeply affect an individual's overall wellness, can be initiated by either kidney cancer or a gradual reduction in kidney function. As the chronic disease advances, it can reach a critical stage where only dialysis or surgery can save lives. Halting its progress becomes crucial. CKD patients also face a heightened risk of premature death. Early detection of associated conditions poses a challenging task for healthcare professionals aiming to prevent their onset. A unique deep learning model is presented in this work for the prediction of CKD. Many existing CKD prediction models have the drawbacks of producing less accuracy, mispredicting, utilizing more computation time, and using low-quality datasets or data with noise and missing values, leading to misprediction. So it is necessary to develop new techniques that give high predictions with less computation time. The objective of this research work is to build improved ResNet models for the prediction of chronic kidney disease and evaluate their performance in comparison to other cutting-edge machine learning and deep learning methods. This research work developed ResNet models such as improved ResNet 152v2 with inception, improved ResNet 101, and improved ResNet50 models that produced 99.90%, 96.53%, and 93.968% accuracy, respectively. The proposed ResNet models for CKD prediction will be useful to nephrologists and other medical professionals.

Keywords: Deep learning, ResNet, chronic kidney disease, prediction

[1] [1]Department of Computer Science & Engineering,Chandigarh University, Punjab , 140413,India .

ORCID ID: https://orcid.org/0000-0003-2082-9900 Email ID: vetrigold@gmail.com

[2] Department of Cyber Security, Bapatla Engineering College, Andhra Pradesh, 522102, India.

Email ID: himabindu.r@becbapatla.ac.in

[3]Department of Mathematics, Chandigarh University, Mohali, 140413, India.

Email ID: mphilsaranya@gmail.com

[4]Department of Computer Science & Engineering, Chandigarh University, Mohali, 140413

Email ID: neha.arya35@gmail.com

[5]Department of Computer Science & Engineering, KG Reddy College of Engineering and Technology,Telangana.,India.

Email ID: pnrshkumar@gmail.com

[6]Department of Computer Science & Engineering, KG Reddy College of Engineering and Technology, Telangana,India.

Email ID: azramd4@gmail.com

[7]Department of CSE,Koneru Lakshmaiah Education Foundation,Vaddeswaram,Guntur,Andhra Pradesh,India.

Email ID: ashok.bekkanti@gmail.com

[8]Faculty of Science and technology,JSPM University, Pune.

Email ID: kul.a.pws@gmail.com

1. INTRODUCTION

When affected by chronic kidney disease, or CKD, the kidneys undergo a decline in their ability to efficiently filter blood. The main role of the kidneys is to remove surplus water and toxins from the blood [1].The impacted population is 500 million, of which 78% (387.5 million) reside in LMICs (low- to middle-income nations) [2][3]. In the world, non-communicable chronic kidney disease (CKD) has considerably increased patient admission rates, mortality, and morbidity [2].CKD can be attributed to a range of illnesses, Including, yet not confined to diabetes, high blood pressure, and heart disease.

CKD development is influenced not only by major health issues but also by a person's age and gender [4]. If one or both of your kidneys are not functioning correctly, we might encounter a variety of symptoms such as back pain, abdominal discomfort, diarrhea, elevated body temperature, nosebleeds, skin rashes, and vomiting [5].Many patients with CKD do not become aware of their condition until it is too late because It often remains asymptomatic until it has advanced to a more severe stage. During its initial phases, CKD frequently remains asymptomatic, primarily because the human body can generally tolerate a substantial reduction in kidney function. is estimated that 0.4% of persons have CKD in stages4 or 5. A multifaceted pharmaceutical strategy used to treat people with established CKD and renal insufficiency with the goal of reducing all the risk variables linked to progression is likely to decrease progression in most cases and even stop it in others[7][8]. Due to the substantial and increasing incidence of CKD, it will be essential to create new models of healthcare delivery in the future that coordinate the care of nephrologists, other doctors, and other healthcare professionals. Additionally, it will be essential to assess the affordability of testing programmes for populations that are most at risk of developing CKD [9]. A kidney transplant or dialysis is likely to be required at some point. Typically, laser point energy that functions in a superposition area is created using machine vision. Convolutional networks incorporating deep learning (DL) concepts are employed in the models in this field [10]. Convolutional neural networks (CNN) and deep learning(DL) are employed to handle the low-resolution image issue in an effort to advance the quality of the image and its clarity[11]. If it were possible to identify acute renal disease within the human body using machine learning (ML) techniques, it would be advantageous to address this problem[6][12]. Some recent studies try to predict the presence of CKD utilizing sequence models like naive Bayes and artificial neural network (ANN) models like C4.5. Particle swarm optimization (PSO) is utilized to diminish the computation time [13].Renal disease is one of the life-threatening ailments that are being studied [14][15]. Ebrahime et al.[16] Suggested classification algorithms utilizing recursive feature elimination techniques for the prompt detection of CKD. This algorithm performs well on the dataset, which has only 400 samples. Polat et al.[17] suggested a method employing an SVM classifier in 2017 to make an early diagnosis of CKD . The importance of their research was highlighted by the use of feature selection methods to diminish the dataset's complexity. The wrapper and filter approaches were the two feature selection techniques used. The Best Technique for choosing Features in Search Engines using the SVM classifier and the filtered subset evaluator produced a 98.5% accuracy rate. This showed that feature selection techniques can have a big impact on how well a model performs [18][19].

In contrast to manual visual interpretation, artificial intelligence (AI), DL algorithms, and ML have demonstrated greater performance in the diagnosis of diseases just as COVID-19, Breast cancer, kidney disease, and diabetic retinopathy [2][20][21][22]. Several data augmentation techniques were used to create additional data with variations and kidney dimension observations by cutting out the kidneys' periphery from ultrasound pictures in order to retrieve the information[23]. Both genetic and non-genetic risk prediction models for CKD are created by using a logistic regression model with weighting based on the natural logarithm. [24]. With the use of opportunity-based evaluation of patients at elevated risk as a full practice, It's possible to pinpoint individuals at an elevated risk of developing CKD by means of community gatherings, workplace events, or specific locations such as pharmacies[25].

Predicting kidney disease using deep learning models present several technical hurdles that make it a challenging endeavor:

i) Limited availability of labelled data: The limited availability of high-quality labelled data for kidney disease prediction makes it difficult to train accurate models.

ii) Variability in disease progression: The progression of kidney disease can vary greatly from patient to patient, making it difficult to build models that can accurately predict disease progression and outcomes.

iii) Overfitting and generalization: Deep learning algorithms are prone to overfitting training data and underperforming on fresh, untried data.

iv) The model produces less accuracy and requires more computation.

The important objectives of this research work are: a) to develop a deep learning-based kidney disease prediction model that can directly deal with the images.

b) To do better preprocessing on the dataset to improve the quality of the images.

c) Developing models that can produce high accuracy with less computation time.

2. BACKGROUND STUDY

Kidney disease prediction involves various deep learning techniques for examining extensive sets of patient data and recognizing patterns that can be applied to evaluate the potential emergence of renal disease. Fuzhe Ma et al.[26] proposed the SVM and Multilayer Perceptron algorithms on the IOMT platform to identify, classify, and diagnose chronic renal failure beforehand. Alsuhibany et al.[27] proposed an ensemble of Deep Learning models, which were used in CKD recognition and categorization in cloud architecture powered by IoT and produced 96.9% accuracy, but the model used only 400 samples. Raj Anish et al.[28] proposed a sharpness-aware minimization technique for automatic kidney volume segmentation in patients, which uses combining cosine loss to improve segmentation accuracy. K.Shankar et al.[29] suggested a DL model for kidney disease prediction with the use of feature selection using the Ant Lion Optimization (ALO) technique and produced 90.55% testing accuracy, but this technique used only 400 patient samples. Kumar, Kailash et al.[30] proposed a special DL model that mixes a fuzzy deep neural network with an existing dataset as a csv file and obtained an accuracy of 99.23% for kidney illness diagnosis and prognosis. Ahmed J. Aljaaf et al.[31] proposed a machine learning model framework for early CKD prediction with an impressive 99.5% accuracy. However, it's worth noting that this model's limitations stem from its reliance on a relatively small training and validation dataset of just 400 patient data points. Hossam M et al.[32] proposed an Antlion optimization algorithm-based selection of features for maximizing the classification performance. Vijendra Singh et al.[33]suggested a novel deep learning model that used the most crucial CKD characteristics, such as the number of red blood cells and albumin, to detect the kidney disease. The drawback of this model is that it has been carried out on minimal data sets. As shown in table 1, most of the existing techniques have done disease prediction on the feature datasets that were already extracted. Some existing models lack good accuracy in disease detection and prediction. Some techniques used a too small dataset, and some techniques could not produce good accuracy due to a lack of pre-processing and required high computation times for training and validation. Still, there is a requirement to develop a new deep learning-based model that can deal with images directly and produce good accuracy with less computation time. So this research work provided a solution for the above-identified gaps by developing a ResNet Model to predict kidney disease.

3. MATERIALS AND METHODS

3.1 Materials

3.1.1 Dataset

The kidney image dataset consists of 12,446 images, of which 5,077 are normal images and 7,369 are images of kidney disease. The PACS (picture archiving and communication system) information was collected from a number of institutions in Dhaka, Bangladesh, where patients had already been diagnosed with renal diseases. The coronal and axial cuts were both chosen from contrast and non-contrast exams according to protocol for the total abdomen and urogram. Figure 1 shows some of the normal images and some of the images with kidney disease.

3.1.2 Data Pre-Processing

The dataset image size is 512 x 512 pixels. It was resized into a type of image size in order to equalize its size. After that, the photos were made grey.

The original image size of 512 x 512 pixels is converted to 224 x 224 pixels for further processing. For clear-cut learning of the characteristics in the training data, a sizable amount of training data is necessary at this stage of preprocessing. Image augmentation is a powerful approach for DL models to improve generalization and reduce overfitting. By applying various transformations to the original images, the model can be trained to recognize the object or environment from different viewpoints and under different conditions.

Random flipping*:* To produce a mirror image, flip the image horizontally or vertically.

Random rotation: Rotating the image by a random angle.

Random scaling: Scaling the image by a random factor.

3.2 Methods

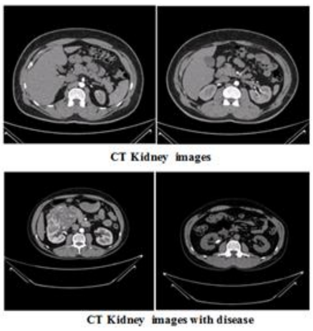


Figure 1.A Few Examples of Normal Kidney Images and CT Images with Kidney Disease

Table 2. *Literature Review on kidney disease detection and prediction*

S. No	Techniques or Methods	Performance metrics (Accuracy)	Advantages	Gap identified
1	Machine learning algorithms[34]	95.8% to 98.3%	Produced good accuracy.	A different dataset with similar features as the datasets that will be evaluated for performance is the one used to identify problems.
2	Machine learning techniques[35]	82.56% to 99.8%	Dimensionality reduction increased the speed of the model.	Machine learning techniques were experimented with on a small dataset.
3	Deep learning model[23]	85.6%	The learning rate for every parameter is adapted automatically using the Adam optimizer.	Produced less accuracy.
4	Logistic regression model [24]	89.4%	The prediction model pinpoints people most at risk of developing CKD before it progresses.	This model performs only on a small dataset.
5	Hazard regression [38]	87% to 91.1%	The model slows CKD progression.	Only 2178 patient's data were used for training and validation.
6	Gray level co-occurrence matrix[37]	98.38%	Produced good accuracy.	Only 300 samples were taken for testing.
7	Random Forest[4]	98 %	Produced good accuracy.	Only 400 samples were taken for training, and computation time is high.
8	AI deep learning algorithm[39]	89.3% to 95.7%	Produced good accuracy.	Cases of CKD may have been misclassified because the term "chronic kidney disease" depends on a single eGFR result.
9	Deep learning[23]	95%	To enhance kidney quality, the useless peripheral portion is eliminated.	High computation time.
10	Naïve Bayes[40]	97.67%	Feature selection reduces the dimensionality of the dataset.	This model cannot deal directly with images, and only 400 samples were available in the dataset.

3.2.1 Resnet152 with inception

ResNet-152v2 and Inception are two separate architectures with different design philosophies. However, it is possible to combine these architectures by using the ResNet-152v2 as a feature extractor and feeding the extracted features into the Inception modules for further processing. The ResNet-152v2 architecture is an improvement over the original ResNet architecture, with deeper layers and better performance. It uses the residual block as its basic building block, and it consists of a skip connection that bypasses one or more convolutional layers. This allows the network to gain residual mappings, which can enhance the functionality of deep neural networks. The Inception architecture, on the other hand, is based on the idea of using multiple branches of convolutional layers with distinct filter sizes and pooling operations and then combining the outputs of these branches. This allows the network to learn both local and global features at different scales. One possible way to combine these architectures is to use a ResNet-152v2 as a feature extractor and then feed the extracted features into an Inception module for further processing. This can be achieved using a sequential model in a deep learning framework such as Keras or TensorFlow.

3.2.2 ResNet-101 model

ResNet-101 consists of 101 layers, comprising shortcut connections, fully interconnected layers (FL), layers with pooling, and layers with convolution. The shortcut connections, also called skip connections, allow for the network to bypass certain layers and preserve the gradient flow during backpropagation, which assists in preventing the vanishing slope issue and allows for greater depth network training. ResNet-101 has attained cutting-edge performance on numerous computer vision exercises, including object identification, picture segmentation, and image classification. ResNet-101 was trained beforehand on the ImageNet dataset, a large-scale collection comprising millions of labelled images.

3.2.3 ResNet-50 model

In 2015, Microsoft Research unveiled ResNet-50, a well-liked deep convolutional neural network architecture. It is frequently employed for applications involving picture recognition and classification and has attained cutting-edge performance on numerous benchmark datasets. The ResNet-50 network can learn residual mappings that can enhance the performance of deep neural networks since its architecture is based on the residual block. A residual block is composed of two or three layers of convolution, and the shortcut connection omits one or more of these layers. Rather than trying to learn the full mapping from scratch, the network can instead learn the residual mapping between the block's input and output. Sixteen layers of convolution, three layers with pooling, and one fully interconnected layer are among the 50 layers that make up ResNet-50. The number of convolutional layers and filters in each of the 23 residual blocks varies.

4. PROPOSED METHODOLOGY

4.1 System model

4.1.1 ResNet-152 with Inception

ResNet-152 with Inception-v3: ResNet-152 with Inception-v3 combines the ResNet-152 and Inception-v3 architectures by replacing the regular convolutional layers in ResNet-152 with Inception modules. This helps improve the accuracy of the network while keeping the computational cost manageable. Mathematically, the output of a ResNet-152 with an Inception-v3 block can be represented in equation (1).

Y=fc1x, c3x, c5x, mpx+ x (1) where [c1(x), c3(x), c5(x), mp(x)] is the inception component's output, f (.) is the residual function, and the block's entry point is x. The function f(.) can be expressed as a series of 1x1 layers of convolution; batch normalization, and a nonlinear activation function are all performed afterward. The output of the block is the count of remaining connections and the output of the residual function.

4.1.2 ResNet -101

ResNet-101 has 101 layers in total, including fully connected layers, layers with convolution, layers with pooling, and residual connections. Here is a brief overview of the mathematical model: Residual Blocks: The core component of ResNet-101 is the residual block. Two or three convolutional layers, batch normalization, and a non-linear activation function, such as ReLU, make up a residual block. The residual block's output is described as follows: y = f(x) + x. where the block's input is x, the block's output is f(x), and y is the output of the residual connection. The function f(x) can be represented as a set of layers of convolution, batch normalization, and a nonlinear activation function, such as ReLU, which are then applied.

Convolutional Layers: ResNet-101 uses a series of convolutional layers with various kernel dimensions to obtain characteristics from the input image.

Pooling Layer: This layer combines the extracted features learned from previous layers. Both median and maximum layers with pooling are used in ResNet-101.

Fully Connected Layers: Before producing the final output, the result of the last layer of pooling is flattened and routed via a number of fully linked layers.

ResNet-101: ResNet-101 is constructed by stacking multiple residual blocks together. The residual blocks are grouped into different stages based on the spatial size of the feature maps. Each stage consists of multiple residual blocks with the same spatial size. The quantity of leftover blocks at each level depends on the depth of the network. Mathematically, the output of a ResNet-101 block can be represented in equation (2).

$$Y=f([c1(x), c2(x), c3(x)]) + x \qquad (2)$$

where [c1(x), c2(x), c3(x)] is the output of an arrangement of convolutional layers with different kernel sizes, f(.) is the residual function, and the block's input is x. The function f (.) can be expressed as a series of layers with convolutional, batch normalization, and a function of nonlinear activation coming next. The block's output is the result of adding the residual connection and residual function outputs together.

4.1.3 ResNet 50

Sixteen layers with convolutional, three layers with pooling, and a single layer that is entirely connected are among the 50 layers that make up ResNet-50. The number of layers of convolution and filters in each of the 23 residual blocks varies.

Residual Blocks: The ResNet-50's primary fundamental component is the residual block. Two or three convolutional layers, batch normalization, and an activation function that is non-linear in ReLU fashion make up a residual block. The residual block's output is specified as: $y = x + f(x)$ where in the block's input is x, f(x) is its output, and y is the block's residual connection's output. Convolutional layers, batch normalization, and a function of nonlinear excitation, such as ReLU, can all be used to represent the function f(x).

Convolutional Layers: To excerpt information from the input image, ResNet-50 employs a number of layers with various kernels. Following the convolutional layers, a non-linear activation function, such as ReLU, is used.

Pooling Layers: This layer combines the extracted features learned from previous layers. In ResNet-50, layers with maximum and average pooling are both employed.

Fully Connected Layers: Before producing the final output, the results of the last layer are flattened and routed through a number of fully linked layers.

ResNet-50 is constructed by stacking multiple residual blocks together. The residual blocks are grouped into different stages based on the spatial size of the feature maps. Each stage consists of multiple residual blocks with the same spatial size. The quantity of leftover blocks at each level depends on the depth of the network. Mathematically, the output of a ResNet-50 block can be represented in equation (3).

$Y= f([c1(x), c2(x)]) + x \qquad (3)$ where [c1(x), c2(x)] is the output of a sequence of convolutional layers with different kernel sizes, f(.) is the residual function, and X is the block's input. This function is expressed as a number of convolutional layers; batch normalization, and a non-linear activation function are then applied. The output of the block is the count of remaining connections and the output of the residual function.

4.2 Architecture and working

The input images are of size 224x224. ResNet152v2 is divided into several stages, the first of which is a convolutional layer with 64 filters and a 7x7 kernel, followed by stage 2: This stage starts with a convolutional layer that performs convolutions on the input image. The layer applies 64 filters, each using a 7x7 kernel. When a stride of 2 is utilized, the filters diminish the spatial dimensions of the feature maps by progressing 2 pixels with each step. After the convolutional layer, batch normalization is utilized. It normalizes the activations of the previous layer, which helps stabilize the training process. The batch-normalized activations are then passed through a ReLU activation function. It zeros out any negative values and maintains positive values as they are, thereby introducing non-linearity into the network. The output of the ReLU activation is fed into a max pooling layer. This layer conducts max pooling using a 3x3 window and a stride of 2. Max pooling further decreases the spatial dimensions while preserving essential information. Stages 2–5 consist of multiple residual blocks with increasing feature map sizes. Every residual block is comprised of multiple layers, usually encompassing convolutional layers, batch normalization, ReLU activation, and skip connections.

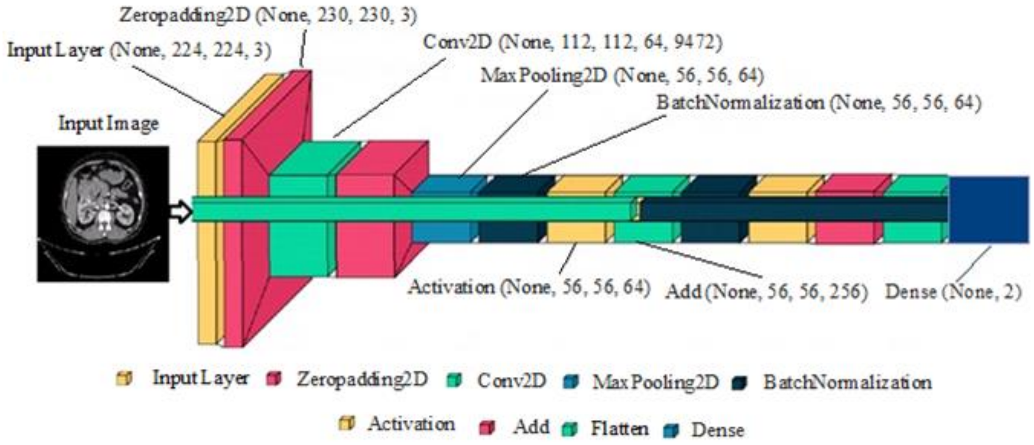

Zeropadding2D (None, 230, 230, 3)
Input Layer (None, 224, 224, 3)
Conv2D (None, 112, 112, 64, 9472)
MaxPooling2D (None, 56, 56, 64)
BatchNormalization (None, 56, 56, 64)
Input Image
Activation (None, 56, 56, 64)
Add (None, 56, 56, 256)
Dense (None, 2)
Input Layer
Zeropadding2D
Conv2D
MaxPooling2D
BatchNormalization
Activation
Add
Flatten
Dense

Figure 2. Overall architecture of the improved ResNet models

These skip connections facilitate direct gradient flow through the block, thereby mitigating the vanishing gradient issue during training. The feature map sizes typically increase with each stage, allowing the network to capture more complex and high-level features. After the previous stages, the feature maps are further processed through a global average pooling layer (GAP). The GAP calculates the mean value for each feature map, generating a constant-length vector that encapsulates the image's feature information as a whole. The GAP layer's output is then linked to a fully connected layer comprising one thousand units. Every unit within this layer is linked to every unit in the preceding layer, allowing for complex combinations of features. Figure 2 shows the overall architecture of the improved ResNet models.

ResNet101 has 101 layers, including layers for pooling and layers with convolution, fully connected layers, and residual connections. The input layer is a layer of convolution with a 7x7 kernel size and 64 filters. The stride is 2, which diminishes the spatial dimensionality of the input image. In ResNet101, Stage 1 consists of a succession of Max layers of pooling, layers of convolution, and residual blocks. The spatial size of the feature maps is reduced from 224x224 to 56x56. Stage 2 consists of a succession of Max layers of pooling, layers of convolution, and residual blocks. The spatial size of the feature maps is reduced from 56x56 to 28x28. Stage 3 consists of a succession of Max layers of pooling, layers of convolution, and residual blocks. The spatial size of the feature maps is reduced from 28x28 to 14x14. Stage 4 consists of a succession of Max layers of pooling, layers of convolution, and residual blocks. The spatial size of the feature maps is reduced from 14x14 to 7x7. Stage 5 is comprised of a fully connected layer, a GAP layer with 1000 units, and a function for activating softmax. The soft maximal function's result is, over the 1000 ImageNet categories, a probability distribution. ResNet-101 employs skip connections, also referred to as residual connections.

ResNet-50 is comprised of a total of 50 layers, which include pooling, fully connected, and residual connections. An overview of the architecture is presented here: The input layer is a convolutional layer with a 7x7 kernel size and 64 filters. The stride is 2, which diminishes the spatiotemporal dimensionality of the input image. Stage 1 consists of a succession of layers with maximum pooling, convolution, and residual blocks. The spatial size of the feature maps is reduced from 224x224 to 56x56. Stage 2 consists of a succession of layers with maximum pooling, convolution, and residual blocks. The spatial size of the feature maps is reduced from 56x56 to 28x28. Stage 3 consists of a succession of layers with maximum pooling, convolution, and residual blocks. The spatial size of the feature maps is reduced from 28x28 to 14x14. Stage 4 consists of a succession of layers with maximum pooling, convolution, and residual blocks. The spatial size of the maps of characteristics is diminished from 14x14 to 7x7. Stage 5 includes a softmax activation function, a fully connected layer with 1000 units, and a GAP layer. The output of the soft maximal operation is a range of probabilities for the 1000 ImageNet categories. ResNet-50 employs skip connections, also referred to as residual connections. These skip connections facilitate the direct flow of gradients from the block's output to its input, effectively mitigating the vanishing gradient issue and enhancing the training of deep networks.

Figure 3 shows the overall flow diagram of the proposed model. Observing the diagram, the model is divided into two main steps: data pre-processing and ResNet models for feature extraction, disease detection, and disease prediction. As a first step, the input, lambda, conv2D, MaxPooling2D, dense, and flatten Keras layers are set, and then a preprocessing library is included to prepare our input data. Data augmentation is carried out using the image data generator, with the aim of generating new images through operations like zooming in, zooming out, and horizontal and vertical shifts, as determined by the parameters specified in the image data and ResNet models for feature extraction, disease detection, and disease prediction. As a first step, the input, lambda, conv2D, MaxPooling2D, dense, and flatten Keras layers are set, and then a preprocessing library is included to prepare our input data. Data augmentation is carried out using the image data generator, with the aim of generating new images through operations like zooming in, zooming out, and horizontal and vertical shifts, as determined by the parameters specified in the image data generator.Image dimensions of 224 x 224 have been opted for in both training and testing. The top layer is set to false, which indicates whether the topmost layer should be added or removed, and in this case, it is being removed. As the weights are pre-trained and held constant, the 'trainable' parameter is set to 'false.'

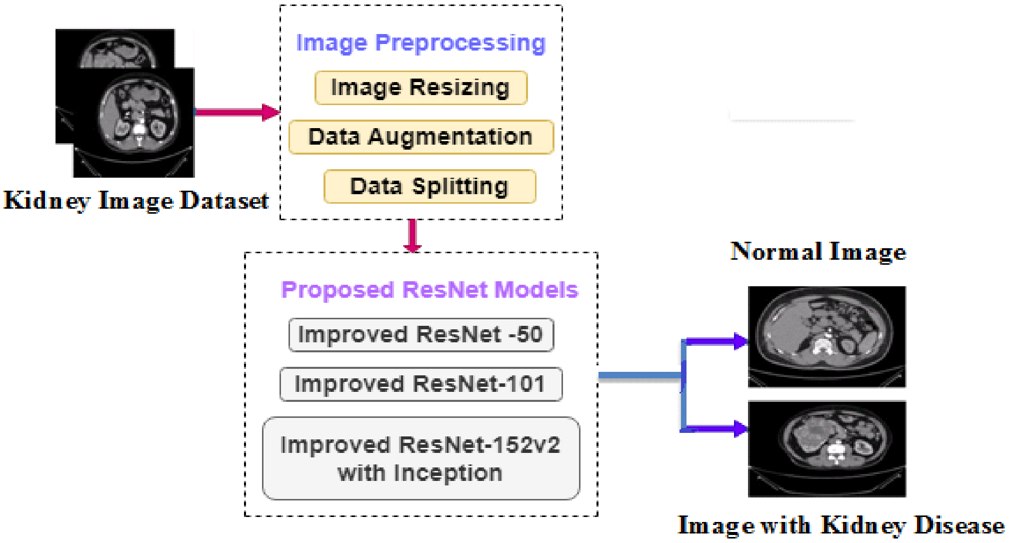


Figure 3. Overall flow diagrams of the proposed ResNet models for kidney disease detection

Without making this change, the model would need to undergo repetitive training, which could result in diminished accuracy. The quantity of output and the available output classes collectively determine the number of output classes in the dataset. ResNet152v2, ResNet101, and ResNet50 have all had their final layers taken off. ResNet152V2 (input shape=IMAGE_SIZE + [3]) is used to set the image size as 224 x 224 with a single channel for training and testing because renal images are in black and white. The flatten layer is eliminated, and weight is presented as an ImageNet, which is already included in Keras. After model evaluation, the image of kidney disease is predicted.

5. EXPERIMENTATION, RESULTS AND ANALYSIS

5.1 Experimental Setup

The total experiment is conducted using Google Colab, which provides a convenient and powerful environment for deep learning. Google Colab supports several popular deep learning frameworks, such as TensorFlow and Keras are available and come with a range of Built-in libraries and tools for analyzing and visualizing data.

Performance Metrics

The equation (4) is the formula that can be used to determine how accurately a deep learning model predicts diseases.

$$Accuracy=TP+TN \quad TP+TN+FP+FN \qquad (4)$$

Recall serves as a dedicated evaluation metric employed to gauge the classifier's effectiveness in achieving accurate classifications through the classification model, which is represented in equation (5)

$$Recall=TPTP+FN \qquad (5)$$

Precision is a performance measure utilized for assessing the accuracy of a binary classification model, with a particular emphasis on the positive class. It represents the percentage of correctly predicted positive cases among all instances that were predicted as positive. Precision tells about the understanding of the model's capacity to diminish false-positive predictions. The formula for precision is represented in equation (6).

$$Precision=TPTP+FP \qquad (6)$$

where TP (True Positive) is the count of positively predicted cases that actually occurred, TN (True Negative) is the count of negatively predicted cases actually occurring, FP (False Positive) is the tally of positively forecasted instances that indeed transpired, and FN (False Negative) is the count of negatively predicted cases actually occurring. ROC-AUC is a frequently used metric for evaluating and comparing classification models. It provides a unified value that summarizes the model's performance, regardless of the chosen threshold. Models with higher ROC-AUC values are generally considered to have better predictive capabilities. Computing the true positive rate (TPR) and false positive rate (FPR) involves multiple probability thresholds for both the total positive instances (P) and total negative instances (N).As shown in equation (7), the AUC is calculated using the following formula:

$$AUC=(sum(TPR\_i) - TPR\_0 / 2) \; P * N \qquad (7)$$

Where TPR_i is the True Positive Rate at point i on the ROC curve, and TPR_0 is the TPR at the starting point (FPR = 0) of the ROC curve. P denotes the collective count of positive instances, while N denotes the cumulative count of negative instances. The formula computes the AUC by adding together the TPR values and dividing the result by the product of the total positive instances (P) and the total negative instances (N). It subtracts TPR_0 divided by 2 to account for the trapezoidal approximation.

5.2 Results

The results of the Improved ResNet models for Chronic Kidney Disease Prediction are presented in this section. Figure 4 displays the feature maps extracted from ResNet's deep CNN, which are valuable resources for gaining deeper insights through visual analysis. This process involves a meticulous examination of the feature maps, allowing us to discern and recognize distinctive patterns, textures, and shapes within the images.

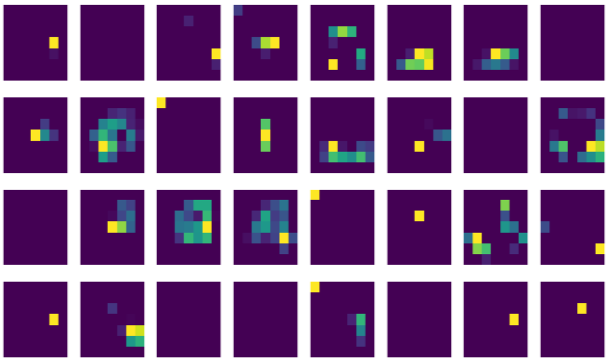


Figure 4. Feature Map Generated by ResNet Models for the Kidney Image Input

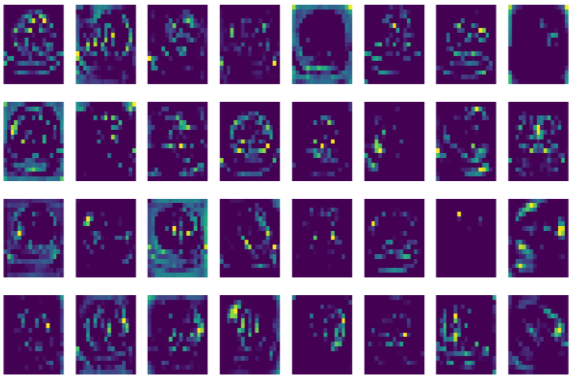


Figure 5. Feature Maps Extracted from the Inception Model for the Kidney Image

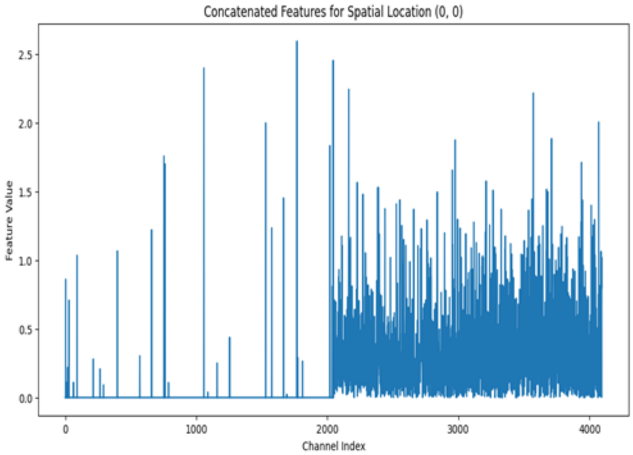


Figure 6. Combined Features Extracted from Kidney Images Using ResNet Models and Inception Model

It's essential to understand that these feature maps possess a hierarchical structure, where early layers focus on capturing fundamental elements such as edges and corners, while higher layers progressively encode more intricate and object-centric information. This hierarchical organization unveils interpretable structures, making it possible to identify features like diagonal lines or specific textures like fur scales. Figure 5 shows that the feature maps extracted from images. These maps represent different levels of abstraction, with early layers capturing basic features like edges, curves, and color variations, while deeper layers encode more intricate details like the shapes of kidney structures or fine-grained textures. Figure 6 shows the combined features extracted from kidney images using both ResNet and Inception models, providing a comprehensive representation of the spatial location of features within the images. In this image, the x-axis represents the channel index, which essentially corresponds to the different extracted feature maps or channels from the two models. Each channel captures specific patterns, edges, textures, or abstract information learned by the models during their training. On the y-axis, we have the feature values, which quantify the importance or presence of these patterns at different spatial positions. This representation allows us to visually inspect and analyze how different features from the ResNet and Inception models contribute to the overall understanding of the kidney images. By examining the values and patterns across channels, the information learned by these models can be combined to potentially make more informed decisions or further process the features for our specific task, such as kidney disease prediction. Detecting kidney cysts or abnormalities in medical images often involves using CAD systems, which can analyze image features, textures, and patterns to identify potential anomalies.

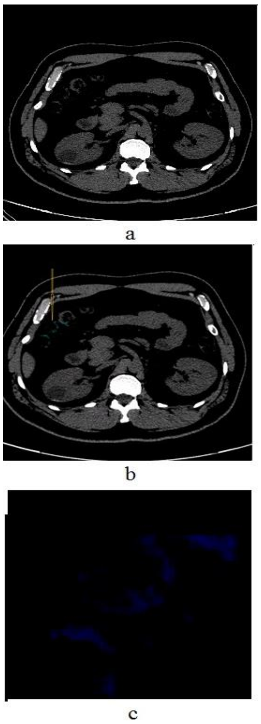


Figure 7.a)The original image b) Diagnosis indicating the possibility of cysts c) An Image with a prominent cyst area highlighted in blue.

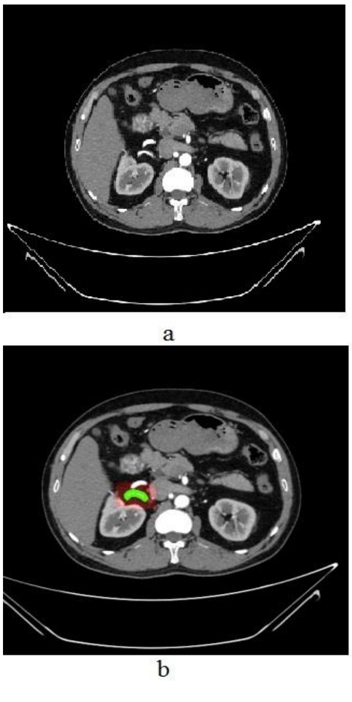


Figure 8. a) The original image b) Diagnosis indicating the possibility of tumor

Kidney cysts are fluid-filled sacs that can form in or on the kidneys, and they may appear as rounded or oval structures with varying degrees of transparency on medical images. The blue color mentioned likely represents the appearance of these cysts in the image. In Figure 7b, we observe a kidney image displaying the possibility of potential kidney cysts, represented by prominent blue-colored bubble-like shapes. Figure 7c illustrates an enlarged view of the potential cyst area, depicted in blue within the 7b image. Renal tumors, commonly referred to as kidney tumors, represent abnormal cell growth within the kidneys. Accurate prediction and timely treatment are vital in managing kidney tumors, particularly in cases of malignancy, as early-stage tumors generally offer a more favorable outlook.

In Figure 8a, the original image is presented, while Figure 8b accentuates the potential presence of a kidney tumor by highlighting distinct, dark red-shaded regions within the kidney input image. These stones can exhibit differences in size, spanning from minuscule particles to more substantial structures resembling golf balls. Kidney stones can be composed of various substances, including calcium, oxalate, uric acid, and cystine. Figure 9a presents the original image, while Figure 9b highlights the potential possibility of stones, represented in green hues. The words "train accuracy," "validation accuracy," "train loss, and "validation loss" are taken into consideration when training and evaluating a deep learning model. The training dataset's accuracy is assessed using the train accuracy metric, revealing the model's ability to classify training instances with accuracy. Conversely, the validation accuracy metric gauges the model's performance on a distinct validation dataset, reflecting its accuracy on previously unseen data. The train loss also referred to as training loss or training error, quantifies the disparity between the anticipated model output and the true target values within the training dataset. Similar to train loss, validation loss is calculated using the validation dataset. It computes the variance between the model's predicted values and the real target values for the validation dataset. As shown in Figure 10, the proposed ResNet models such as ResNet50, ResNet101, and ResNet152v2 with inception produced average training accuracies of 94.021%, 95.752%, and 99.2% and average Validation accuracies of 93.968%, 95.888%, and 99.812% for Epoch 5. The train accuracy curve starts at 90.57% and reaches the highest accuracy of 99.99%, and the validation accuracy curve starts at 89.99% and reaches the highest accuracy of 96.21%. The train loss curve starts at 0.02 and the highest loss value is 0.1143 during the training of the dataset, and the Validation loss curve starts at 0.0301 and the highest validation loss value is 0.1201. Figure 11 illustrates the performance of the proposed ResNet models on the kidney dataset for Epoch 7. The proposed ResNet models such as ResNet50, ResNet101, and ResNet152v2 with inception produced average training accuracies of 93.295%, 95.930%, and 99.08% and average Validation accuracies of 92.887%, 96.531%, and 99.901% for Epoch 7. The train accuracy curve starts at 89.63% and reaches the highest accuracy of 99.99%, and the validation accuracy curve starts at 89.678% and reaches the highest validation accuracy of 100%. The train loss curve starts at 0.0201 and the highest loss value is 0.1237 during the training of the dataset, and the Validation loss curve starts at 0 and the highest validation loss value is 0.1232. This result shows a positive sign that the proposed model did not meet overfitting or underfitting problems during its execution for epochs 5 and 7.

5.2 Analysis

Table 2 illustrates the comparative analysis of training and validation accuracy for the proposed model across various epochs. The ResNet-50 model yielded validation accuracy of 92.8871% and 94.021%, alongside training accuracy of 93.295% and 94.021% for epochs 5 and 7, respectively. Validation accuracy for the ResNet-101 model reached 96.53171%, while training accuracy reached 95.930% for epoch 7, and 95.8884% and 95.752% for epoch 5. Similarly, the ResNet152v2 with inception model excelled with validation accuracy of 99.90143% and training accuracy of 99.08% and 99.2% for epochs 7 and 5, respectively. Figure 12 presents the performance comparison and findings from multiple perspectives.

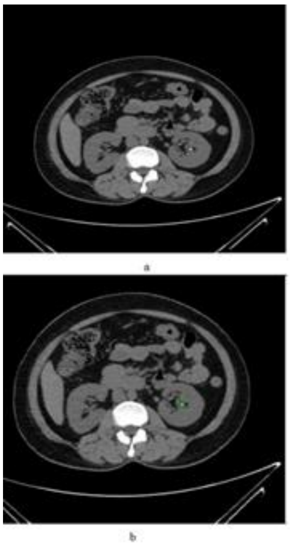


Figure 9.The original image b) Diagnosis indicating the possibility of mild stone

ResNet 152v2 with inception achieved the highest validation accuracy of 99.90143% in epoch 7. For the Resnet50 model, validation accuracy peaked at 93.968% in epoch 7. However, the same model displayed a validation accuracy of 92.887% in epoch 5, indicating a 1.081% increase in the 7th epoch. The ResNet 101 model demonstrated training accuracies of 95.752% and 95.930% for epochs 5 and 7, respectively, which were slightly lower than their respective validation accuracies. Significantly, all models demonstrated higher validation accuracy in comparison to their training accuracy. Nonetheless, the slight variance between training and validation accuracy highlights the robust performance of the models without encountering issues of overfitting or underfitting. The ResNet 152v2 with inception model yielded a sensitivity of 99.93%, a specificity of 99.84%, a precision of 99.89%, and an accuracy of 99.90%. Similarly, the ResNet101 model demonstrated a sensitivity of 97.67%, a specificity of 94.81%, a precision of 96.59%, and an accuracy of 96.53%. In contrast, the ResNet-50 model showcased a sensitivity of 95.31%, a specificity of 91.88%, a precision of 94.76%, and an accuracy of 93.96%.

As illustrated in Figure 13, when comparing sensitivity and specificity, all ResNet models exhibited better sensitivity than specificity. Moreover, when evaluating specificity and precision, all the models demonstrated higher precision values than specificity. The count of images used in this research work is 12,446 for training and 2490 for validation. The ResNet-152v2 with inception model produced 1492 correct positive predictions, along with 2 incorrect positive predictions, 1 incorrect negative prediction, and 995 correct negative predictions, as indicated in figure 14. The ResNet-101 model yielded 1460 correct positive predictions, 52 incorrect positive predictions, 35 incorrect negative predictions, and 943 correct negative predictions, as depicted in figure 15. Similarly, the ResNet-50 model generated 1439 correct positive predictions, 80 incorrect positive predictions, 71 incorrect negative predictions, and 900 correct negative predictions, as illustrated in figure 16.

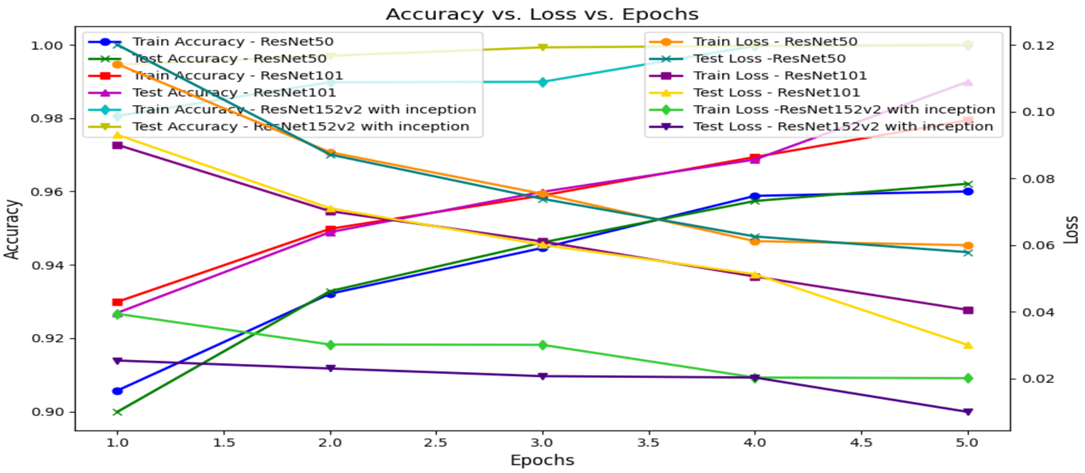


Figure 10. Performance of the proposed ResNet models based on Accuracy and Loss for Epoch 5

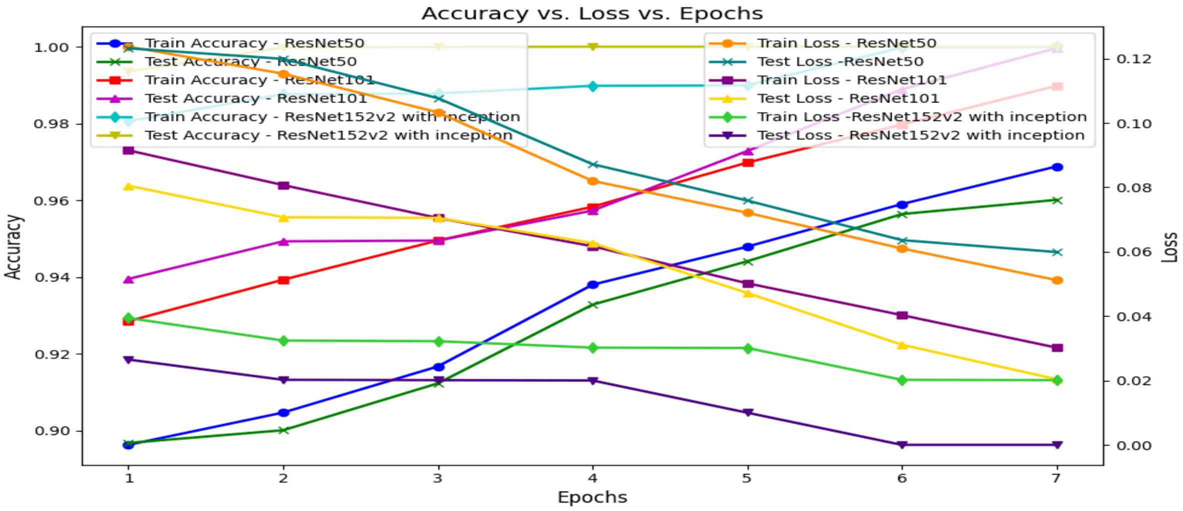


Figure 11. Performance of the proposed ResNet models based on Accuracy and Loss for Epoch 7

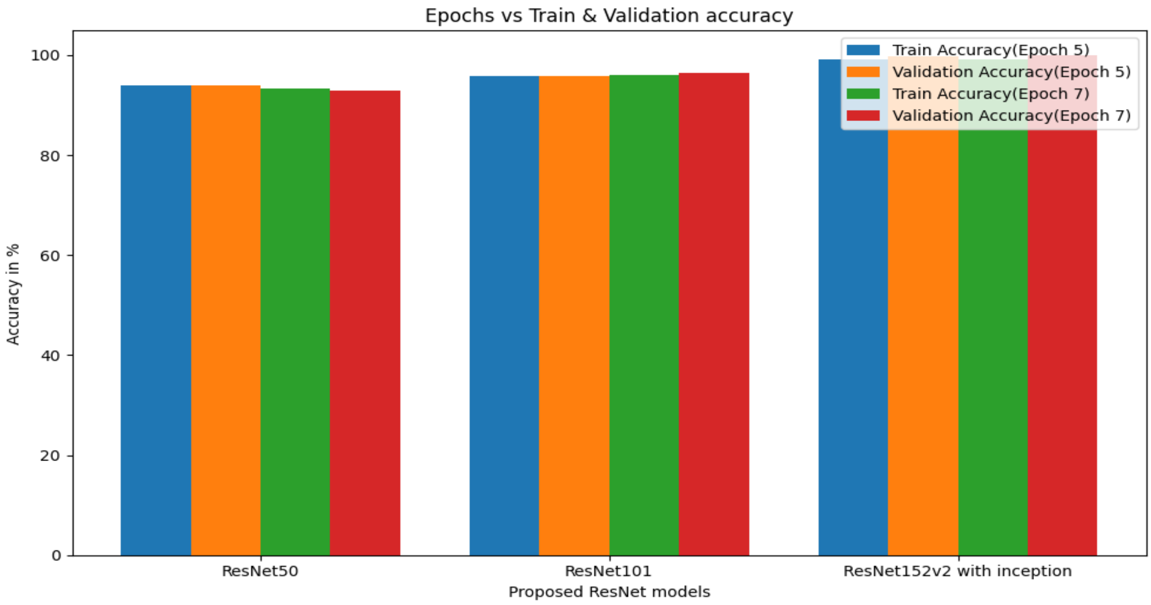


Figure 12. Performance of the proposed ResNet models based on training and Validation Accuracy for various Epochs

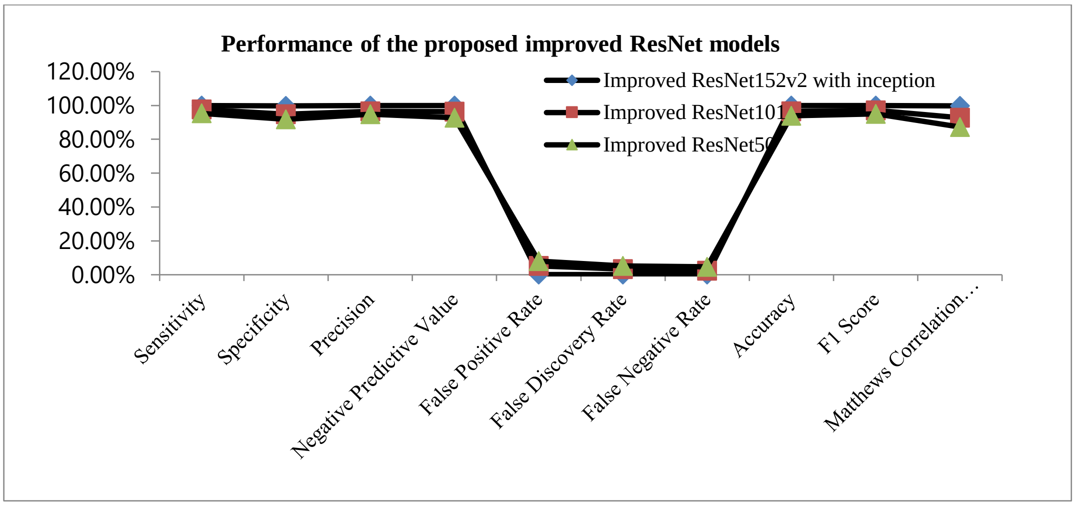


Figure 13. Performance of the proposed improved ResNet models on the kidney image dataset

The ROC-AUC (Receiver Operating Characteristic: Area under the Curve) curve serves as a visual tool for assessing and comparing model performance. It provides important insights regarding a model's capacity to distinguish between positive and negative instances at different classification thresholds.

Table 2. Train and validation accuracy of the ResNet models for various Epochs

ResNet Deep learning model	**Validation Accuracy**		**Training Accuracy**	
	Epoch 5	**Epoch 7**	**Epoch 5**	**Epoch 7**
ResNet152v2 with inception	99.812%	99.901%	99.2%	99.08%
ResNet101 model	95.888%	96.531%	95.752%	95.930%
Resnet50 model	93.968%	92.887%	94.021%	93.295%

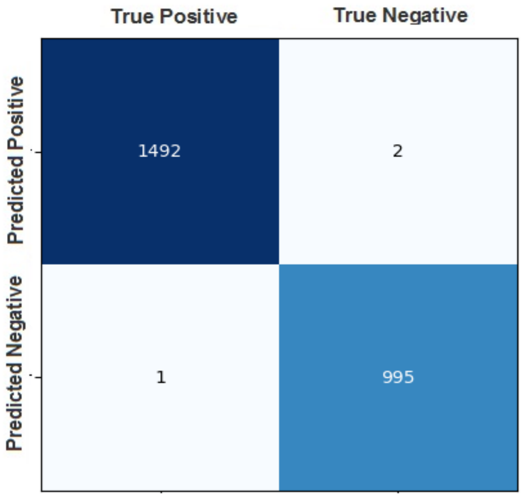


Figure 14. Confusion matrix for an Improved ResNet-152v2 with inception model

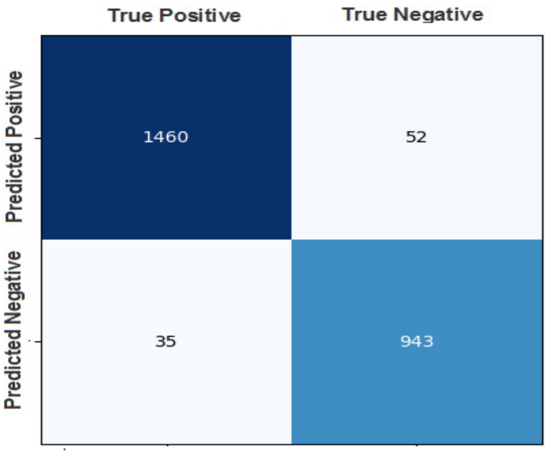
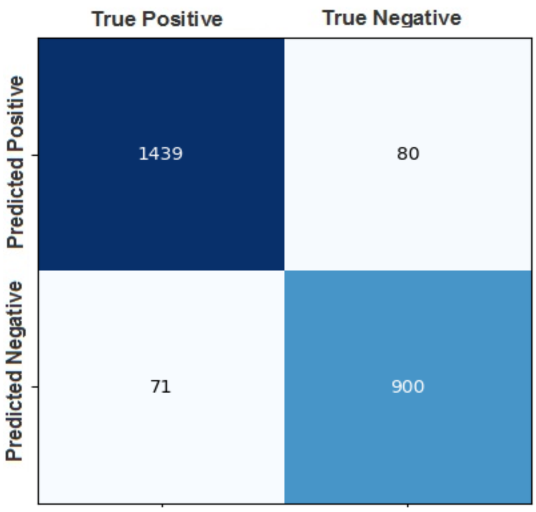


Figure 15. Confusion matrix for Improved ResNet-101 model

This curve is constructed by plotting TPR against the FPR at different classification thresholds. It illustrates how the relationship between TPR and FPR changes as the threshold is adjusted. Typically, the curve commences at the point (0, 0) and concludes at (1, 1). The AUC quantifies the classifier's overall performance and is determined by calculating the area beneath the ROC curve. Its values range from 0 to 1, with 0.5 signifying a classifier that performs at random, and 1 indicating a flawless classifier.

Table 3. Performance of the proposed improved ResNet models on the kidney Dataset

Performance metrics	***ResNet152v2 with inception***	***ResNet-101***	***ResNet -50***
Sensitivity	99.93%	97.67%	95.31%
Specificity	99.84%	94.81%	91.88%
Precision	99.89%	96.59%	94.76%
Negative Predictive Value	99.90%	96.44%	92.71%
False Positive Rate	0.16%	5.19%	8.12%
False Discovery Rate	0.11%	3.41%	5.24%
False Negative Rate	0.07%	2.33%	4.69%
Accuracy	99.90%	96.53%	93.96%
F1 Score	99.91%	97.13%	95.03%
Matthews Correlation Coefficient	99.78%	92.75%	87.33%



	True Positive	True Negative
Predicted Positive	1439	80
Predicted Negative	71	900

Figure 16. Confusion matrix for Improved ResNet-50 model

AUC simplifies the process of comparing and ranking various classifiers by providing a single numerical measure. A higher AUC corresponds to the classifier's enhanced capability in discriminating between positive and negative instances. Figure 17 displays the ROC curves and the associated AUC values for these three models: improved ResNet152v2 with the inception model, improved ResNet-101, and improved ResNet-50, which are 0.996, 0.965, and 0.935, respectively.

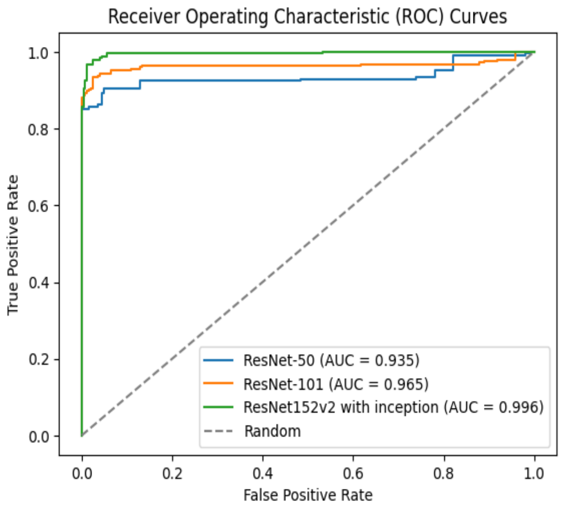


Figure 17. ROC curves of the improved ResNet models

An AUC value that is very close to 1 indicates that the models are able to achieve a high TPR while keeping the FPR low, indicating a strong ability to discriminate between the classes. Table 4 displays a performance evaluation that juxtaposes the proposed model against established techniques, focusing on accuracy. Some of the existing models used high computation times to get good accuracy and used very few samples or a very small dataset for training and validation. As shown in figure 7, our improved ResNet models used less computation time. The improved ResNet152v2 with the inception model produced better accuracy than all other existing techniques and other ResNet models in their 7 epochs. It is clear that the identified gaps in the literature survey are fulfilled in this research work.

Table 4. Comparison of the proposed models with existing models

Techniques/Methods	***Accuracy***
Deep learning model [23]	85.6%
ML Techniques[34]	98.3%
ML Algorithms[35]	99.8%
Logistic Regression model[24]	89.4%
Hazard regression [36]	91.1%
Gray level co- occurrence matrix [37]	98.38%
Random Forest[4]	98%
Artificial intelligence DL[38]	95.7%
Naïve Bayes[39]	97.67%
fuzzy deep neural network[30]	99.23%
Ensemble DL[27]	96.9%
Multilayer Perceptron algorithm with SVM[26]	98%
Ant Lion Optimization [29]	90.55%
Improved ResNet152v2 with inception(proposed)	99.90143%
Improved ResNet101(proposed)	96.53171%
Improved Resnet50 (proposed)	93.968%

6. Conclusion

Deep learning models for predicting chronic diseases are presented in this work. In this work, we have enhanced the ResNet152v2, ResNet 101, and ResNet 50 models to address the deficiencies identified in the literature review. We have employed the chronic kidney disease image dataset with the improved ResNet models. The proposed ResNet models, including the improved ResNet152v2 with the inception model, the improved ResNet 101, and the improved ResNet 50, performed well, achieving accuracy rates of 99.90%, 96.53%, and 93.968%, respectively. Among all the ResNet models, ResNet 152v2 with the inception model outperformed the others, producing a sensitivity of 99.93%, a specificity of 99.84%, and a precision of 99.89%. In our future work, we plan to develop models that integrate multiple models and incorporate clinical data, laboratory data, and genetic data for kidney disease detection and classification.

Conflicts of interest

There is no conflict of interest of this paper.

References

[1] Z. Chen, X. Zhang, and Z. Zhang, “Clinical risk assessment of patients with chronic kidney disease by using clinical data and multivariate models,” *Int. Urol. Nephrol.*, vol. 48, no. 12, pp. 2069–2075, 2016, doi: 10.1007/s11255-016-1346-4.

[2] C. George, A. Mogueo, I. Okpechi, J. B. Echouffo-Tcheugui, and A. P. Kengne, “Chronic kidney disease in low-income to middle-income countries: The case f increased screening,” *BMJ Glob. Heal.*, vol. 2, no. 2, pp. 1–10, 2017, doi: 10.1136/bmjgh-2016-000256.

[3] K. T. Mills *et al.*, “A systematic analysis of worldwide population-based data on the global burden of chronic kidney disease in 2010,” *Kidney Int.*, vol. 88, no. 5, pp. 950–957, 2015, doi: 10.1038/ki.2015.230.

[4] N. Buza and M. Dizdar, “Cmbebih 2017,” vol. 62, no. Ml, 2017, doi: 10.1007/978-981-10-4166-2.

[5] L. Zhang *et al.*, “Prevalence of chronic kidney disease in China: A cross-sectional survey,” *Lancet*, vol. 379, no. 9818, pp. 815–822, 2012, doi: 10.1016/S0140-6736(12)60033-6.

[6] J. He, J. Lin, and M. Duan, “Application of Machine Learning to Predict Acute Kidney Disease in Patients With Sepsis Associated Acute Kidney Injury,” *Front. Med.*, vol. 8, no. December, pp. 1–11, 2021, doi: 10.3389/fmed.2021.792974.

[7] A. M. El Nahas and A. K. Bello, “Chronic kidney disease: The global challenge,” *Lancet*, vol. 365, no. 9456, pp. 331–340, 2005, doi: 10.1016/S0140-6736(05)17789-7.

[8] C.-H. Lin, Y.-C. Chang, and L.-M. Chuang, “Early detection of diabetic kidney disease: Present limitations and future perspectives,” *World J. Diabetes*, vol. 7, no. 14, p. 290, 2016, doi: 10.4239/wjd.v7.i14.290.

[9] L. F., V. L.D., and P. P., “The importance of early detection of chronic kidney disease,” *Nephrol. Dial. Transplant.*, vol. 17, no. SUPPL. 11, pp. 2–7, 2002, [Online]. Available: http://ovidsp.ovid.com/ovidweb.cgi?T=JS&PAGE=reference&D=emed5&NEWS=N&AN=2002397414.

[10] W. M. Jwaid, “Image Processing Technology and Deep Learning Application: In Relation to the Context of Laser Positioning,” *J. Phys. Conf. Ser.*, vol. 1879, no. 3, 2021, doi: 10.1088/1742-6596/1879/3/032130.

[11] X. Wang and S. Li, “Image saliency prediction by learning deep probability model,” *Signal Process. Image Commun.*, vol. 78, no. July, pp. 471–476, 2019, doi: 10.1016/j.image.2019.08.002.

[12] N. T. Rao, “Prediction and Forecasting of Persistent Kidney Problems Using Machine Prediction and Forecasting of Persistent Kidney Problems Using Machine Learning Algorithms,” no. October, 2020, doi: 10.31782/IJCRR.2020.122031.

[13] F. Nie and J. Li, “Image Object Background Classification with Jensen-Shannon Divergence,” *IAENG Int. J. Appl. Math.*, vol. 52, no. 4, 2022.

[14] V. Kunwar, K. Chandel, A. S. Sabitha, and A. Bansal, “Chronic Kidney Disease analysis using data mining classification techniques,” *Proc. 2016 6th Int. Conf. - Cloud Syst. Big Data Eng. Conflu. 2016*, pp. 300–305, 2016, doi: 10.1109/CONFLUENCE.2016.7508132.

[15] S. Bala and K. Kumar, “A Literature Review on Kidney Disease Prediction using Data Mining Classification Technique,” *Int. J. Comput. Mob. Comput.*, vol. 3, no. 7, pp. 960–967, 2014, [Online]. Available: www.ijcsmc.com.

[16] E. M. Senan *et al.*, “Diagnosis of Chronic Kidney Disease Using Effective Classification Algorithms and Recursive Feature Elimination Techniques,” vol. 2021, 2021.

[17] H. Polat, H. D. Mehr, and A. Cetin, “Diagnosis of Chronic Kidney Disease Based on Support Vector Machine by Feature Selection Methods,” 2017, doi: 10.1007/s10916-017-0703-x.

[18] A. Saha, “Performance Measurements of Machine Learning Approaches for Prediction and Diagnosis of Chronic Kidney Disease (CKD),” pp. 200–204, 2019.

[19] A. Haratian, Z. Maleki, F. Shayegh, and A. Safaeian, “Detection of factors affecting kidney function using machine learning methods,” *Sci. Rep.*, no. 0123456789, pp. 1–15, 2022, doi: 10.1038/s41598-022-26160-8.

[20] A. I. Technology *et al.*, “OPTIMUM FEATURE SELECTION BASED BREAST CANCER PREDICTION USING MODIFIED LOGISTIC,” vol. 101, no. 8, pp. 2895–2908, 2023.

[21] D. Vetrithangam, V. Senthilkumar, Neha, A. R. Kumar, P. N. Kumar, and M. Sharma, “Coronary Artery Disease Prediction Based on Optimal Feature Selection Using Improved Artificial Neural Network With Meta-Heuristic Algorithm,” *J.*

Theor. Appl. Inf. Technol., vol. 100, no. 24, pp. 4771–4782, 2022.

[22] D. Vetrithangam, V. Indira, S. Umar, B. Pant, M. K. Goyal, and B. Arunadevi, “Discriminating the Pneumonia-Positive Images from COVID-19-Positive Images Using an Integrated Convolutional Neural Network,” *Math. Probl. Eng.*, vol. 2022, 2022, doi: 10.1155/2022/5643977.

[23] C. C. Kuo *et al.*, “Automation of the kidney function prediction and classification through ultrasound-based kidney imaging using deep learning,” *npj Digit. Med.*, vol. 2, no. 1, 2019, doi: 10.1038/s41746-019-0104-2.

[24] J. Zhao *et al.*, “An early prediction model for chronic kidney disease,” *Sci. Rep.*, vol. 12, no. 1, pp. 1–9, 2022, doi: 10.1038/s41598-022-06665-y.

[25] T. Mathew and O. Corso, “Review article: Early detection of chronic kidney disease in Australia: Which way to go?,” *Nephrology*, vol. 14, no. 4, pp. 367–373, 2009, doi: 10.1111/j.1440-1797.2009.01113.x.

[26] F. Ma, T. Sun, L. Liu, and H. Jing, “Detection and diagnosis of chronic kidney disease using deep learning-based heterogeneous modified artificial neural network,” *Futur. Gener. Comput. Syst.*, vol. 111, pp. 17–26, 2020, doi: 10.1016/j.future.2020.04.036.

[27] S. A. Alsuhibany *et al.*, “Ensemble of Deep Learning Based Clinical Decision Support System for Chronic Kidney Disease Diagnosis in Medical Internet of Things Environment,” *Comput. Intell. Neurosci.*, vol. 2021, no. Cc, 2021, doi: 10.1155/2021/4931450.

[28] A. Raj *et al.*, “Deep Learning-Based Total Kidney Volume Segmentation in Autosomal Dominant Polycystic Kidney Disease Using Attention, Cosine Loss, and Sharpness Aware Minimization,” *Diagnostics*, vol. 12, no. 5, 2022, doi: 10.3390/diagnostics12051159.

[29] K. Shankar, P. Manickam, G. Devika, and M. Ilayaraja, “Optimal Feature Selection for Chronic Kidney Disease Classification using Deep Learning Classifier,” *2018 IEEE Int. Conf. Comput. Intell. Comput. Res. ICCIC 2018*, pp. 1–5, 2018, doi: 10.1109/ICCIC.2018.8782340.

[30] K. Kumar, M. Pradeepa, M. Mahdal, S. Verma, M. V. L. N. RajaRao, and J. V. N. Ramesh, “A Deep Learning Approach for Kidney Disease Recognition and Prediction through Image Processing,” *Appl. Sci.*, vol. 13, no. 6, p. 3621, 2023, doi: 10.3390/app13063621.

[31] A. J. Aljaaf *et al.*, “Early Prediction of Chronic Kidney Disease Using Machine Learning Supported by Predictive Analytics,” *2018 IEEE Congr. Evol. Comput. CEC 2018 - Proc.*, pp. 1–9, 2018, doi: 10.1109/CEC.2018.8477876.

[32] H. M. Zawbaa, E. Emary, and B. Parv, “Feature selection based on antlion optimization algorithm,” *Proc. 2015 IEEE World Conf. Complex Syst. WCCS 2015*, 2016, doi: 10.1109/ICoCS.2015.7483317.

[33] V. Singh, V. K. Asari, and R. Rajasekaran, “A Deep Neural Network for Early Detection and Prediction of Chronic Kidney Disease,” *Diagnostics*, vol. 12, no. 1, pp. 1–22, 2022, doi: 10.3390/diagnostics12010116.

[34] M. A. Islam, M. Z. H. Majumder, and M. A. Hussein, “Chronic kidney disease prediction based on machine learning algorithms,” *J. Pathol. Inform.*, vol. 14, no. September 2022, p. 100189, 2023, doi: 10.1016/j.jpi.2023.100189.

[35] D. A. Debal and T. M. Sitote, “Chronic kidney disease prediction using machine learning techniques,” *J. Big Data*, vol. 9, no. 1, 2022, doi: 10.1186/s40537-022-00657-5.

[36] H. Li, Y. Pan, J. Zhao, and L. Zhang, “Skin disease diagnosis with deep learning: A review,” *Neurocomputing*, vol. 464, pp. 364–393, 2021, doi: 10.1016/j.neucom.2021.08.096.

[37] R. Ahmad and B. K. Mohanty, “Biomedical Signal Processing and Control Chronic kidney disease stage identification using texture analysis of ultrasound images,” *Biomed. Signal Process. Control*, vol. 69, no. April, p. 102695, 2021, doi: 10.1016/j.bspc.2021.102695.

[38] W. Tuntayothin, S. J. Kerr, C. Boonyakrai, S. Udomkarnjananun, S. Chukaew, and R. Sakulbumrungsil, “Development and Validation of a Chronic Kidney Disease Prediction Model for Type 2 Diabetes Mellitus in Thailand,” *Value Heal. Reg. Issues*, vol. 24, pp. 157–166, 2021, doi: 10.1016/j.vhri.2020.10.006.

[39] C. Sabanayagam *et al.*, “Articles A deep learning algorithm to detect chronic kidney disease from retinal photographs in community-based populations,” *Lancet Digit. Heal.*, vol. 2, no. 6, pp. e295–e302, doi: 10.1016/S2589-7500(20)30063-7.

[40] K. M. Almustafa, “Prediction of chronic kidney disease using different classification algorithms,” *Informatics Med. Unlocked*, vol. 24, p. 100631, 2021, doi: 10.1016/j.imu.2021.100631.