

Bio-Inspired Feature Selection for Improving AI-based Kidney Disease Prediction

Mrunali Sonwalkar¹, Dr. Sharvari C. Tamane²

¹Assistant Professor, Computer Science Engineering, College of Engineering Ambajogai, Beed, Maharashtra, India

Email ID: mrunali.sonwalkar@gmail.com

²S Head of a Department, Information Technology, JNEC and University Department of Information and Communication Technology (UDICT), MGM University, Chatrapati Sambhaji Nagar, India

Email ID: hodudict@mgmu.ac.in

Cite this paper as: Mrunali Sonwalkar, Dr. Sharvari C. Tamane, (2025) Bio-Inspired Feature Selection for Improving AI-based Kidney Disease Prediction. *Journal of Neonatal Surgery*, 14 (10s), 954-970.

ABSTRACT

Kidney disease occurs when the kidneys become weakened and lose their ability to cleanse the blood. Most individuals show no symptoms in the early stages of kidney disease. As the condition progresses, toxins can accumulate in the bloodstream, causing complications such as anemia, hypertension, diabetes, osteopenia, and nerve damage. While these issues often develop gradually and without noticeable symptoms, they can eventually lead to sudden renal failure. Early identification of kidney disease allows for the most effective treatment. Predicting kidney function and disease using kidney ultrasound imaging is widely considered in clinical practice due to its safety, simplicity, and affordability. Several works on kidney disease prediction have already been done, but accuracy improvement is still needed. To solve this issue, the research proposes optimized Feature Selection (FS) and an Artificial Intelligence (AI) model for effective kidney disease prediction from ultrasound images. The own dataset with normal and diseased kidney images is created and processed. The processed image features are extracted using the Gray-Level Co-Occurrence Matrix (GLCM) technique. The most significant features are retrieved using Particle Swarm Optimization (PSO), a bio-inspired algorithm. The features of GLCM and PSO are given to the Support Vector Machine (SVM) and Convolutional Neural Network (CNN) models for the classification of diseased and normal images. The SVM model with GLCM and PSO features and CNN with GLCM and PSO features are evaluated using accuracy, precision, recall, and F1 score. Both models show better accuracy improvement by using PSO features. The experimental findings show that the PSO-CNN model gives the maximum accuracy of 98.57% when compared with other models.

Keywords: Kidney Disease, Particle Swarm Optimization, Feature Selection, Ultrasound Images, Artificial Intelligence, Accuracy.

1. INTRODUCTION

The kidneys are very important to the body because they get rid of waste, maintain an appropriate balance of fluids and electrolytes, and generate red blood cells and blood pressure hormones [1]. The kidneys produce urine as a consequence of filtering waste items from the blood. The kidneys govern the body's water, salt, and mineral levels, as well as its blood pressure [2]. Their main function is to remove impurities from the blood and transform the toxic substances into urine. Even though most people have two kidneys, the normal functioning of one permits them to live a normal life. When the kidneys don't perform properly, waste accumulates in the body, causing illness. Extreme cases of kidney failure may be fatal. Nevertheless, many people may manage kidney failure with appropriate treatment. Diabetes and high blood pressure are the most prevalent causes of kidney failure [3]. Untreated kidney disease can lead to renal failure [4]. Secondary health problems caused by kidney failure include weak bones, neurological impairment, and malnutrition. Other potential causes of kidney disease include polycystic kidney disease, glomerular diseases, and autoimmune diseases, which can affect multiple bodily systems.

Medical imaging techniques used in the diagnosis and treatment of kidney disease include computed tomography (CT), ultrasound (US), and magnetic resonance imaging (MRI). The wealth of information about renal architecture and function supplied by these imaging modalities can aid in properly diagnosing and treating kidney disorders, including information regarding renal blood flow and tissue properties. Traditional medical image interpretation can be time-consuming and error-prone, particularly in complicated scenarios like kidney cancer or chronic kidney disease (CKD). Recently, there has been significant discussion on using Deep Learning (DL) algorithms to improve the efficiency and accuracy of medical image analysis in kidney disease detection.

The research focuses on fully automated kidney disease detection from US images. The images are collected, processed, and features are extracted using the GLCM model. The most important features are selected by PSO. SVM and CNN AI models are employed to classify the images as normal or diseased. The AI models are evaluated using GLCM and PSO features, which helps to identify the importance of features in the AI model.

The research article is structured as follows: Section I discusses kidney function and possible diseases, and available medical images to identify these diseases. Section II surveys recent articles on kidney disease using different data types. Sections III-VI discuss the methodology of the proposed framework, including data collection and processing, feature extraction and selection methods, and classifiers. Section VII discusses the outcomes of the SVM model using GLCM and PSO features, and the CNN model using GLCM and PSO features. Section VIII concludes the research and provides future directions.

2. LITERATURE SURVEY

Many researchers are working on kidney disease prediction using numerical and various medical image data. This literature survey section details some of the recent work and analyzes the studies to identify their drawbacks. These issues will be addressed by the proposed research. The article [6] discusses the identification of kidneys for stones based on US images. The CNN and VGG16 models are employed to retrieve the features from US images. Random forests and extreme gradient boosting classifiers (XGBoost) are used for classification. We tested XGBoost and random forest using CNN and VGG16 features. Normal and renal stone images were categorized. The US images for this investigation were provided by Iraq's Al-Diwaniyah General Teaching Hospital. The most accurate model is CNN-XGBoost, which is extremely accurate. The paper [7] presents a strategy for automatically detecting cysts and kidney stones in images. Preprocessing methods are used to enhance the image quality. The next phase involves segmentation based on the image's entropy. The image's brightness has been increased using the gamma correction method. To classify the kidney images, this study used ResNet-50 for feature extraction and SVM for classification. Several more classification methods are utilized to investigate the CNN model. According to the findings, ResNet-50 with SVM is an ideal solution for identifying kidney disease. In the study [8], the author provides a CNN model for accurately classifying normal, cyst, tumor, and stone CT kidney images. In every category, the proposed CNN model outscored the competitors. The model accurately detected all test images, demonstrating its ability to detect kidney abnormalities in CT scans. The study's findings indicate that CNNs can accurately identify kidney CT images, potentially improving patients' diagnostic and treatment outcomes.

Paper [9] introduces a multi-scale CNN-based kidney segmentation model. The model consists of three components: a pyramid pooling, an encoder, and a decoder. To accommodate features of varied sizes, we build a multi-scale input pyramid in the encoder with parallel branches. The decoder creates an output supervisor module that has many outputs. With the multi-output supervision module, the network can train to predict increasingly accurate segmentation outcomes as it scales up. Using the same kidney US dataset, we compare the proposed methodology to other cutting-edge methodologies using six quantitative criteria, highlighting its effectiveness. The paper [10] describes an inductive transfer-based ensemble Deep Neural Network (DNN) for automatically detecting CT kidney stone images. Three datasets were created to retrieve features by a pre-trained DNN framework. Following the assembly of several pre-trained DNNs, the ensemble feature vector is constructed. To detect kidney stones, the K Nearest Neighbor model is updated with a Bayesian optimizer and a 10-fold cross-validation method. The ensemble deep feature vectors selected using the Iterative Relief feature selection approach are then fed into this classifier. By using both high-quality and noisy image datasets, the proposed method surpasses earlier DNN-based and more conventional picture detection algorithms in terms of accuracy.

The paper [11] utilizes a hybrid technique by combining ML and DL approaches. A prognostic technique is developed based on the kidney dataset from the TCIA public data collection. The suggested system's purpose is to find the Explorer data analysis strategy for the CKD dataset by combining random forest regression and multinomial regression techniques. The secondary method involves analyzing a dataset of CT images. This dataset was processed using a CNN architecture that included several layers of deep convolution filters. This approach aims to accurately identify and categorize kidney issues. To determine whether or not there is renal disease, primary and secondary outcomes are combined. The author compares the proposed Hybrid Deep K-Net (H-DKN) approach to the existing studies. The research [12] aims to develop a framework for predicting the start of CKD using DL and ensemble learning approaches. Using DL and ensemble architecture, this study looks into the understudied subject of early CKD prediction. To address the gap in present detection techniques and preventive interventions, we built an ensemble approach by integrating the most effective individual models (LSTM-Adam, CNN-Adamax, and LSTM-BLSTM-Adamax), and also addressed data imbalance, FS, and optimization for 6-12 month CKD prediction. The proposed ensemble system outperforms previous research in terms of accuracy.

Paper [13] employs a successful new hybrid FS strategy to determine the most relevant aspects for CKD disease classification. Next, the data from Kaggle and the UCI repository are fed into a prediction model that uses two proposed classification algorithms, Enhanced Multi-Layer Perceptron (MLP) and Optimized Multi-Layer Perceptron (OMLP), to identify complex patterns and optimize the learning algorithm for early CKD prediction. The analysis of performance metrics enables the measuring of sickness categories. Compared to the suggested MLP and other standard approaches, the experimental outcome shows that the OMLP strategy works better. The study [14] examined eight ensemble learning

techniques for detecting CKD using UCI datasets. The classifier performance improved when the datasets were cleaned by filling the missing data by the imputation methodology and dealing with imbalance characteristics by the SMOTE technique. To increase classifier performance and find optimal solutions, hyperparameter tuning, recursive feature removal, and the Boruta approach to identify the most essential features and minimize the execution time. In terms of accuracy and compilation time, LightGBM outperformed cutting-edge and other ensemble techniques.

By analyzing recent work, we identified that AI models can detect kidney disease, but there is room for improvement in accuracy. Additionally, the computation time and complexity of models to analyze image data are very high. The proposed model aims to overcome both of these issues. One of the reasons for complexity and time consumption is the high number of features. In this work, we employ PSO for effective FS. CNN models have shown promising results in recent days for medical image analysis. The accuracy is enhanced by integrating the PSO outcomes with the CNN model.

3. METHODOLOGY

For classifying kidney health status from US images, we proposed a novel strategy using advanced technologies. The proposed strategy starts with collecting real-time kidney images from healthcare and processing images to enhance their quality. After pre-processing, the features are extracted using the GLCM method. The most important features are selected from the extracted features using the PSO method. The GLCM and PSO features are given to AI models like SVM and CNN. The performance of SVM and CNN models using GLCM and PSO features is evaluated using performance metrics. From the results, the best method for kidney disease prediction is identified. The methodology workflow is given in Figure 1.

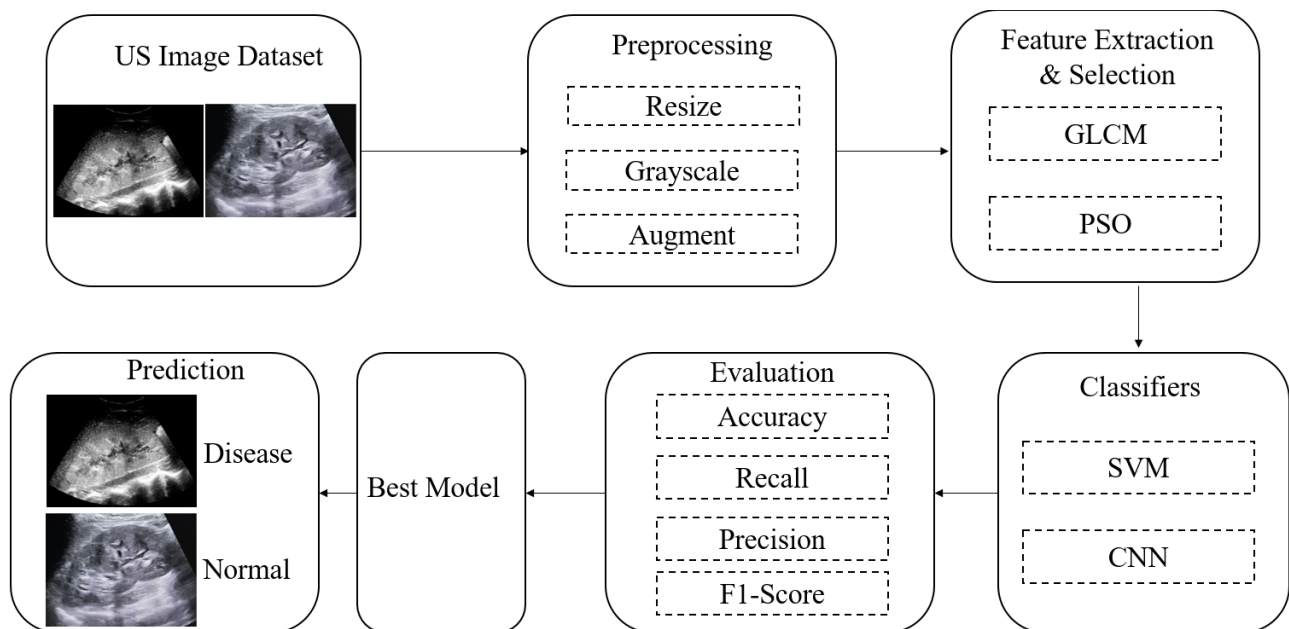


Fig. 1. Workflow of the proposed methodology for kidney disease prediction

4. DATA ACQUISITION AND PROCESSING

The kidney images were collected from Aadhar Diagnostic Centre in Maharashtra and annotated by Dr. Nitin Rajaram Potdar, MD Radiology, DMRD, Consultant Radiologist, who has 18 years of ultrasound experience and 5 years of experience with the Army Medical Corps. We collected a total of 15 images from the diseased and 15 images from the normal category. Samples of diseased and normal images are shown in Figure 2. The real images are not perfect for direct input to an AI model. Some pre-processing is required for the raw images. The pre-processing done on the US kidney images is detailed in this section.

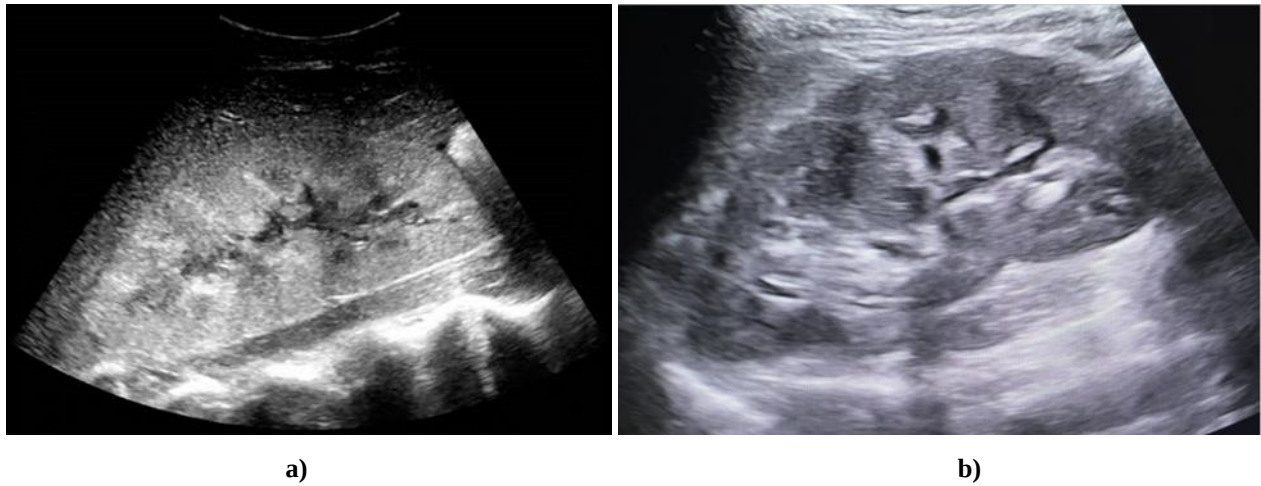


Fig. 2. Sample ultrasonic kidney images

The first pre-processing step is resizing [15]. The collected images are of various dimensions, so all the images are resized to 244×244 pixels. Next, the resized images are converted to grayscale [16]. This conversion helps to reduce the dimensions and facilitate feature extraction. The outcomes of the pre-processing stage for healthy and diseased images are shown in Figures 3 and 4.

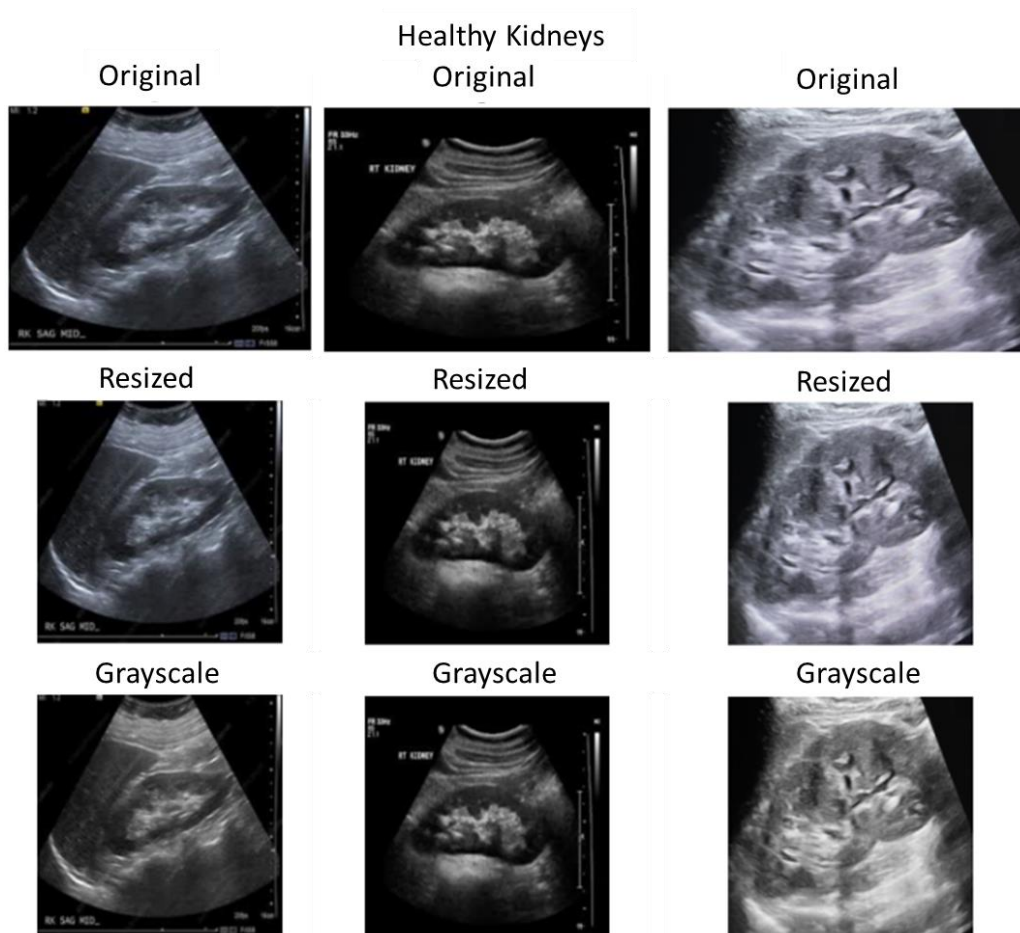


Fig. 3. Outcome of pre-processing healthy images

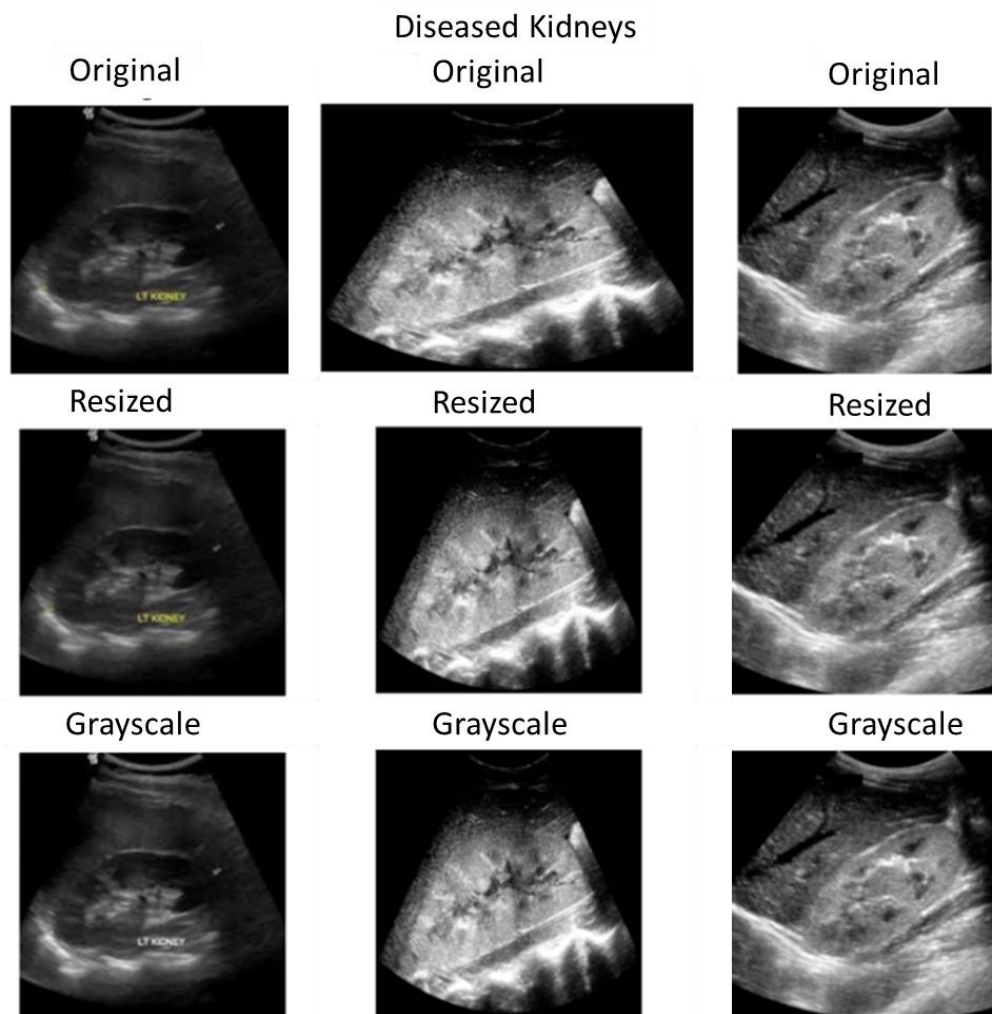


Fig. 4. Outcome of pre-processing diseased images

Since the collected samples are limited, some augmentation [17] is done to increase the sample quantity. The augmentations used include rotation up to 40 degrees, shear and zoom range of 0.2, horizontal flip, width shift of 0.2, and vertical shift of 0.1. The outcomes of augmented healthy and diseased kidney US images are shown in Figures 5 and 6. The final dataset consists of 350 images for the healthy and 350 for the normal category. Of these 700 images, 80% are used for training and 20% are used to test the AI model.

Augmented Normal (Healthy) Kidneys

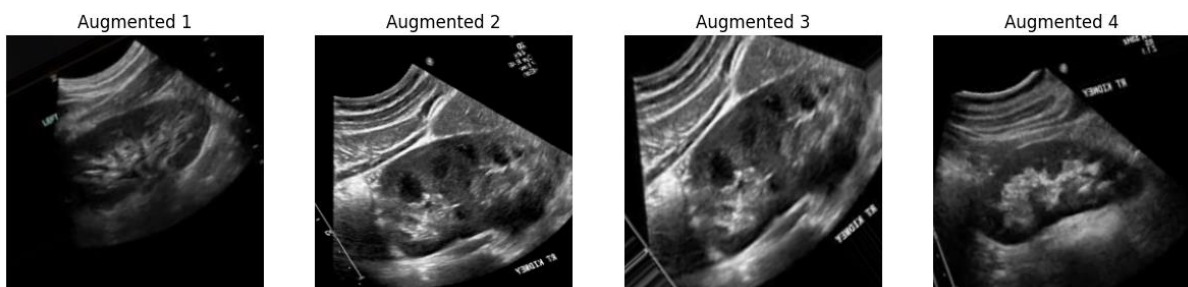


Fig. 5. Augmented healthy images

Augmented Abnormal (Diseased) Kidneys

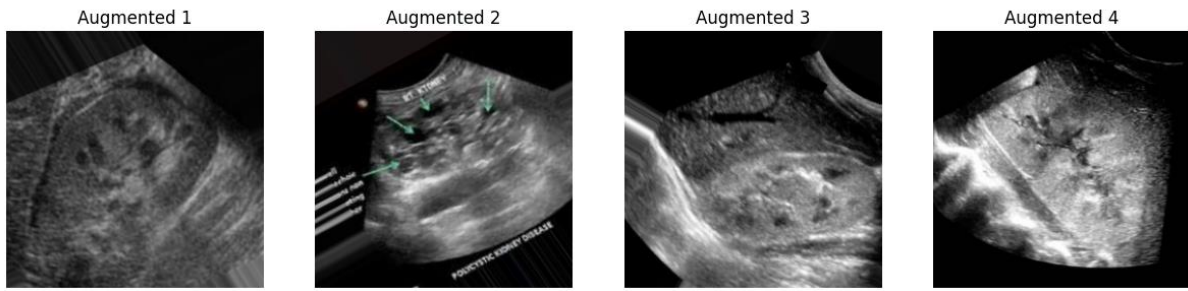


Fig. 6. Augmented normal images

I. Feature Extraction and Selection

After pre-processing, we performed feature extraction from the processed images. Features play an important role in ML classifier accuracy [18]. The model quality improves when appropriate features are provided to the ML model. In this research, feature extraction is performed using the GLCM method. FS is also conducted to minimize the feature dimensions, which helps to deduce the time and space needed to execute the ML model [19]. For FS, the PSO method is used. The theoretical concepts of GLCM and PSO are detailed in this section.

A. GLCM

The GLCM approach is a critical feature extraction technique that transforms gray values into texture information [20]. Haralick [21] proposed the texture analysis technique, which assigns a textural association to every pixel in an image. It captures structural details from the texture pattern, allowing it to be studied at various orientations and scales while enhancing efficiency and simplicity of implementation. Previous studies [22, 23] on GLCM texture analysis reveal that seven of fourteen indicators (contrast, dissimilarity, entropy, correlation, energy, Angular Second Moment (ASM), and homogeneity) are widely employed. The details of the features and their equations are provided below.

Contrast is used to quantify the amount of local texture changes in an image. As a result, the contrast is high when there are numerous variations and low when there are few variations. The Equation (1) gives the contrast formula.

$$\text{Contrast} = \sum_{i,j} (i - j)^2 p(i, j) \quad \text{Contrast} = \sum_{i,j} (i - j)^2 p(i, j) \quad [1]$$

The image has a high correlation if the grayscale values of various pixels in the image are linearly dependent on one another. A low correlation value denotes non-homogeneity, whereas a high correlation value suggests image uniformity. Equation (2) allows for its calculation.

$$\text{Correlation} = \sum_{i,j} \frac{(i - \mu_i)(j - \mu_j)p(i, j)}{\sigma_i \sigma_j} \quad \text{Correlation} = \sum_{i,j} \frac{(i - \mu_i)(j - \mu_j)p(i, j)}{\sigma_i \sigma_j} \quad [2]$$

The entropy of an image reveals how widely spaced its pixels are. To be more specific, an image's entropy and grayscale distribution grow more dispersed as its texture density increases. In contrast, areas of an image with lower entropy levels appear smoother. The entropy value is proportional to the data range being considered. Equation (3) can be used to compute it.

$$\text{Entropy} = -\sum_{i,j} p(i, j) \cdot \log_2 p(i, j) \quad \text{Entropy} = -\sum_{i,j} p(i, j) \cdot \log_2 p(i, j) \quad [3]$$

Homogeneity indicates how uniformly dispersed the image's pixels are and allows for comparison of their relative values. Furthermore, the following calculation yields this index, which is comparable to energy. Equation (4) gives the homogeneity formula.

$$\text{Homogeneity} = \sum_{i,j} \frac{p(i, j)}{1 + |i - j|} \quad \text{Homogeneity} = \sum_{i,j} \frac{p(i, j)}{1 + |i - j|} \quad [4]$$

ASM measures repeated pixel pairs, often known as textural consistency. In images, it detects texture defects. The angular second moment has a maximum value of one. A constant periodic form for the gray level distribution yields greater values. The ASM is calculated using Equation (5)

$$\text{ASM} = \sum_{i,j} p(i, j)^2 \quad \text{ASM} = \sum_{i,j} p(i, j)^2 \quad [5]$$

Energy is calculated by summing the squares of all GLCM elements. It assesses the homogeneity of texture in a picture and is regarded as the most acceptable measure for spotting anomalies. The Equation (6) gives the energy formula:

$$Energy = \sqrt{\sum_{i,j} p(i,j)^2} \quad Energy = \sqrt{\sum_{i,j} p(i,j)^2} \quad [6]$$

Dissimilarity is a linear measurement of an image's local variances and it is computed using Equation (7)

$$Dissimilarity = \sum_{i,j} |i - j| \cdot p(i,j) \quad Dissimilarity = \sum_{i,j} |i - j| \cdot p(i,j) \quad [7]$$

Where, $\sum_{i,j} p(i,j)$ is the sum of the normalized grayscale values at kernel coordinates ii and jj , where ii and jj are the image's row and column numbers, respectively, and the sum equals 1. Greyscale images with dimensions of 2444×244 were used to create the textural features. For each grayscale image in this research, a total of 28 features were created, with rotation angles 0° , 45° , 90° , and 135° .

B. PSO

Eberhart and Kennedy [24] developed PSO, a population-based methodology. PSO has been demonstrated to be an efficient and widely recognized search strategy [25, 26]. This strategy is appropriate for FS challenges because it is simple to encode features, provides a global search facility, is computationally reasonable, has few parameters, and is easy to apply. For the reasons stated above, PSO is used for FS. Figure 7 illustrates the PSO approach for FS.

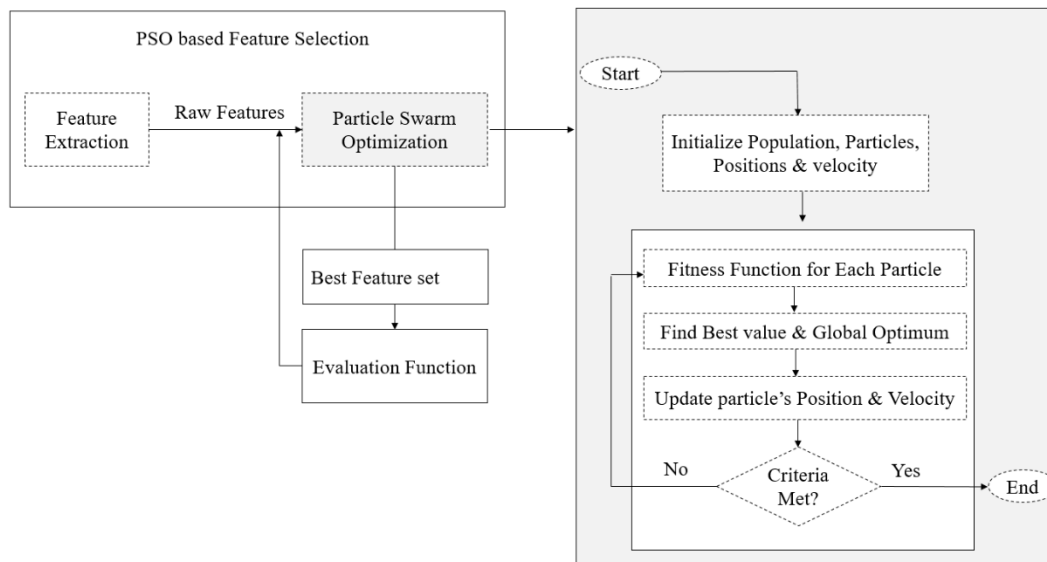


Fig. 7. Working of PSO for feature selection

Using PSO, the most important features were examined and selected from the search space. The particles in PSO form a population that represents potential solutions in the search space. The particle swarm is generated by randomly distributing 1's and 0's. If a particle's major component is one, it is picked; otherwise, it is rejected. Thus, each particle represents a separate subset of the basic components. Following a random initialization, the particle swarm adjusts its position and velocity as it wanders around the search space, looking for the best features. Equation (8) and (9) represents the particle's current position and velocity, respectively.

$$x_i = \{x_{i1}, x_{i2}, \dots, x_{iD}\} \quad x_i = \{x_{i1}, x_{i2}, \dots, x_{iD}\} \quad [8]$$

Where DD indicates the search space dimension.

$$v_i = \{v_{i1}, v_{i2}, \dots, v_{iD}\} \quad v_i = \{v_{i1}, v_{i2}, \dots, v_{iD}\} \quad [9]$$

The particle's position and velocity are determined using Equation (9) and (10).

$$\begin{aligned} v_{id}^{t+1} &= \omega * v_{id}^t + c_1 * r_{1i} * (P_{id} - x_{id}^t) + c_2 * r_{2i} * (P_{gd} - x_{id}^t) \\ v_{id}^{t+1} &= \omega * v_{id}^t + c_1 * r_{1i} * (P_{id} - x_{id}^t) + c_2 * r_{2i} * (P_{gd} - x_{id}^t) \end{aligned} \quad [10]$$

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1} \quad [11]$$

Where, tt is the process iteration, while dd is the search space dimension, ww represents the mass of inertia, and c_1c_1 and c_2c_2 are the acceleration constants. Both $r_{1i}r_{1i}$ and $r_{2i}r_{2i}$ are random numbers with a distribution of 0 to 1. $P_{id}P_{id}$ and $P_{gd}P_{gd}$ reflect the elements of $pbest$ and $gbest$ at dimension of dd . Until a stopping criterion [27] is reached (maximum iterations or an acceptable fitness value), each particle's position and velocity data are continually updated in search of the ideal set of features. The PSO algorithm employed was detailed.

PSO Pseudocode

Step 1. Randomly initialize each particle's (ii) position and velocity.

Step 2. Evaluate the particle fitness function

If the fitness of $x_i > pbest_i$

Update $pbest_i = x_i$

End

If the fitness of $pbest_i > gbest_i$

Update $gbest_i = pbest_i$

End

Step 3. Update the particle's (ii) velocity and position

Update the velocity

$$v_{id}^{t+1} = \omega * v_{id}^t + c_1 * r_{1i} * (P_{id} - x_{id}^t) + c_2 * r_{2i} * (P_{gd} - x_{id}^t)$$

$$v_{id}^{t+1} = \omega * v_{id}^t + c_1 * r_{1i} * (P_{id} - x_{id}^t) + c_2 * r_{2i} * (P_{gd} - x_{id}^t)$$

Update the position

$$x_{id}^{t+1} = x_{id}^t + v_{id}^{t+1}$$

Step 4. Check stopping criteria

If the stopping criterion is met,

Return $gbest$ and its fitness values

Else

Continue Steps 2 and 3

II. Artificial Intelligence Model

For kidney prediction from US images, two AI models, SVM and CNN, are employed. The workings and mathematical operations of both models are detailed in this section.

A. SVM

To classify data points, SVM, a supervised learning technique based on vector theory, represents them as vectors on a spatial grid [28]. Hyperplanes are employed to make decisions and categorize the data samples by separating the different types of data as far away as possible [29]. Hyperplanes are built and trained using labeled data points to help the model to classify the new data. SVMs are constructed by utilizing kernel-based techniques. In linear algebra, hyperplane learning capability is gained by taking the inner product of observations rather than applying them directly [30]. To calculate the inner product, sum all of the input pairs and multiply by their respective products. For example, if we had two input vectors (a, b) and (c, d) , the inner product could be calculated as $(a * c) + (b * d)$. We can forecast the inputs by calculating the dot product of the input (x) and the support vector (x_i) , which is determined by Equation (12):

$$f(x) = B_0 + \sum (a_i * (x, x_i)) \quad f(x) = B_0 + \sum (a_i * (x, x_i)) \quad [12]$$

Equation (1) uses all of the data's support vectors to generate the inner product of input $(x)(x)$, and the learning algorithm forecasts the coefficients of $B_0 B_0$ and $a_i a_i$ (input) during training. When data can't be segregated linearly, a transformation function ϕ based on dot products is needed to convert the input space into a feature space.

$$\left\{ \begin{aligned} &Max_a \left\{ \sum_{i,j=1}^m a_i - \frac{1}{2} \sum_{i,j=1}^m a_i a_j u_i u_j \phi(x_i) \cdot \phi(x_j) \right\} \\ &Max_a \left\{ \sum_{i,j=1}^m a_i - \frac{1}{2} \sum_{i,j=1}^m a_i a_j u_i u_j \phi(x_i) \cdot \phi(x_j) \right\} \end{aligned} \right\} \quad [13]$$

$$\{a_i \geq 0, i = 1, \dots, m\} \{a_i \geq 0, i = 1, \dots, m\} \quad [14]$$

$$\{\sum_{i=1}^m a_i u_{i=0}\} \{\sum_{i=1}^m a_i u_{i=0}\} \quad [15]$$

In which the transformed input xx from the ii -th element is indicated by $\phi(x_i)\phi(x_i)$. Since computing the scalar product is not practical, the kernel trick will be used instead. Equation (16) defines this according to Hilbert-Schmidt theory.

$$\langle \phi(x) \cdot \phi(x_i) \rangle = \sum_{i=1}^{\infty} \lambda_i \phi_i(x) \phi_i(y) = K(x, y) \langle \phi(x) \cdot \phi(x_i) \rangle = \sum_{i=1}^{\infty} \lambda_i \phi_i(x) \phi_i(y) = K(x, y) \quad [16]$$

Where, $\lambda_i \lambda_i$ represents the ii th element weighting coefficient.

It takes a while to train the model on non-linearly separable data using the basic SVM. Moreover, the traditional SVM classifier is not the best choice when handling large amounts of data due to its tendency to yield imprecise findings. As a dot product, the kernel can be computed by the Equation (17):

$$(x, x_i) = \text{sum}(x * x_i)(x, x_i) = \text{sum}(x * x_i) \quad [17]$$

The kernel is used to determine the degree of similarity, or distance, between the input data and support vectors. Some kernel types can handle higher-dimensional data and distinguish between classes that cannot be separated linearly using lines.

B. CNN

CNN is the most well-known and commonly utilized DL algorithm [31]. One significant enhancement above previous CNN versions is its capacity to discover relevant features automatically and without human assistance. CNNs have been widely used in various industries. A CNN consists of many layers which include Convolution Layers (CL), Pooling Layers (PL), and Fully Connected Layers (FCL) [32]. The CNN architecture is made up of multiple building elements organized into layers. Below, we'll go over each layer of the CNN architecture and what they accomplish.

Convolutional Layer: The CL is the foundation of CNNs. It consists of convolutional filters, commonly called as kernels. The feature map is generated by convolving the input image with these filters [33]. Each value in the kernel is called the kernel weight. During CNN training, the kernel initially receives a set of random numbers as weights. Various methods are used to initialize these weights. The kernel then learns to extract key features by adjusting the weights in each training iteration. We begin by outlining the CNN input format. CNNs accept multichannel images as input, whereas standard neural networks (NNs) take vector formats. To comprehend the convolution process, assume a 4×4 grayscale image with a 2×2 kernel and randomly initialized weights. The kernel first travels horizontally and vertically throughout the image. When the input image and kernel are dot-product, both of their values are multiplied and totaled to produce a single scalar value, which is computed synchronously. This process continues until sliding is no longer possible. It is essential to remember that the computed dot product values represents the final feature map.

Pooling Layer: The primary role of the PL is to subsample feature maps created by convolution processes [34]. In essence, this step reduces the sizes of feature maps from their original larger dimensions. It retains the most dominant data in the pooling operation. Similar to the convolution operation, before pooling, stride and kernel sizes are specified. Many pooling techniques are available and we have chosen the maximum pooling method [35]. While the PL aids in evaluating the presence of specific features in the input data, its primary focus is on accurately pinpointing those features, which may result in some loss of overall CNN performance. This constitutes the fundamental drawback of the PL.

Activation Function: Every activation function in a NN serves the same fundamental purpose: to transform input into output. It computes the weighted sum of neuron inputs and biases, determining the appropriate output for a given input and deciding whether to activate a neuron. Non-linear activation layers are positioned adjacent to each weighted layer in a CNN configuration. Apart from improving CNN learning of difficult tasks, the non-linearity of activation layers ensures that the mapping of input to output is also non-linear. Furthermore, the activation function has to be differentiable to allow error backpropagation during network training. In CNNs, the ReLU function [36] is used in input and hidden layers. This function

transforms input values into positive integers. ReLU offers computational efficiency over other functions. Equation (18) provides the mathematical expression.

$$f(x)_{ReLU} = \max(0, x) \quad f(x)_{ReLU} = \max(0, x) \quad [18]$$

Fully Connected Layer: The FCL is often found at the last of all CNN designs. In the FCL each neurons are linked to all the previous layer neurons. It acts as a CNN classifier, receiving input from the CL or PL preceding it. The feature maps are flattened to generate this vector-based input. The FCL determines the final CNN output. The output is generated using the sigmoid activation function. This function accepts real values as input and outputs integers ranging from 0 to 1. The sigmoid function's mathematical depiction of its S-shaped curve is provided by Equation (19).

$$f(x)_{\text{sigm}} = \frac{1}{1+e^{-x}} \quad f(x)_{\text{sigm}} = \frac{1}{1+e^{-x}} \quad [19]$$

The output layer is responsible for the final categorization in a CNN design. It uses cross-entropy loss functions to estimate the predicted error across all training data. This function highlights the disparity between expected and actual results. The next step involves improving it with the CNN learning algorithm.

Learning Algorithm: Gradient Descent reduces training error by iteratively adjusting network parameters after each training phase [37]. Specifically, it computes a first-order derivative on the network parameters to determine the objective function gradient and updates the parameters accordingly to minimize inaccuracies. Back-propagation [38] is employed to propagate the gradient at each neuron to all neurons in the next layer, facilitating parameter updates. Equation (20) numerically represents this operation.

$$w_{ij}^t = w_{ij}^{t-1} - \Delta w_{ij}^t, \quad \Delta w_{ij}^t = \eta * \frac{\partial E}{\partial w_{ij}} w_{ij}^t = w_{ij}^{t-1} - \Delta w_{ij}^t, \quad \Delta w_{ij}^t = \eta * \frac{\partial E}{\partial w_{ij}} \quad [20]$$

While w_{ij}^{t-1} represents the weight from the preceding training epoch, w_{ij}^t indicates the weight from the current training epoch. The E represents the prediction error, and the η represents the learning rate.

5. RESULTS AND DISCUSSION

The research aims to design an effective AI model to identify kidney disease using US images obtained from the hospital. The images are processed to improve quality. Next, to extract the features, GLCM is employed. From the GLCM features, the most important features are selected by the PSO algorithm. The GLCM and PSO features are given to the AI model to analyze the result. The outcome of the research is discussed in this section. The research was done on the Google Colab platform, using the T4 GPU hardware accelerator, and Python coding was used.

A. AI Model Outcome without Feature Selection

First, the GLCM features were given to the SVM and CNN models to predict kidney disease. The 560 feature samples of kidney images were used to train the SVM model. Then 140 features were tested. The outcome of the SVM model in confusion matrix format is given in Figure 8. The confusion matrix consists of TP, TN, FP, and FN. The figure gives a detailed count of correct and incorrect predictions of kidney and normal images. The SVM correctly identifies 76 samples as normal and 51 as diseased. The SVM wrongly identified 7 samples as normal and 6 as diseased. The TP, TN, FP, and FN obtained by the SVM model using GLCM features are 76, 51, 7, and 6.

		PREDICTED OUTCOME	
		0	1
ACTUAL OUTCOME	0	76	6
	1	7	51

Fig. 8. Confusion Matrix of SVM model on GLCM features

The same number of GLCM feature samples are given to the CNN model for training and testing. The CNN model is evaluated using accuracy and loss measures in the training phase. The accuracy and loss values attained in each epoch of the training phase are given in Figures 9 and 10. The testing phase outcome of the CNN model in confusion matrix format is given in Figure 11. The figure shows the CNN model's prediction quality in terms of correct and incorrect predictions. The CNN correctly identifies 80 and 54 samples as normal and diseased, respectively. The CNN wrongly identified 2 samples as diseased and 4 as normal. The TP, TN, FP, and FN obtained by the CNN model using GLCM features are 80, 54, 4, and 2.

CNN MODEL: LOSS PLOT

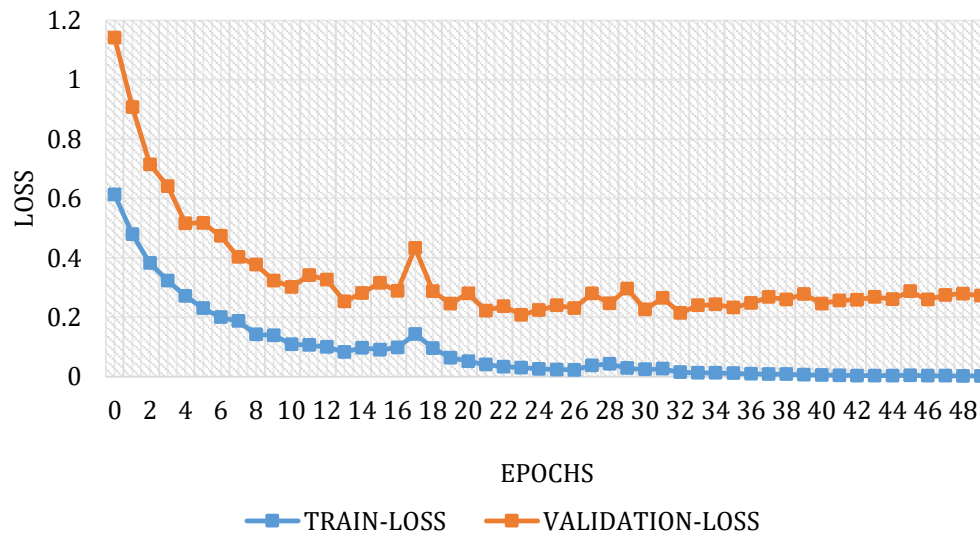


Fig. 9. Loss plot of CNN model on GLCM features

CNN MODEL: ACCURACY PLOT

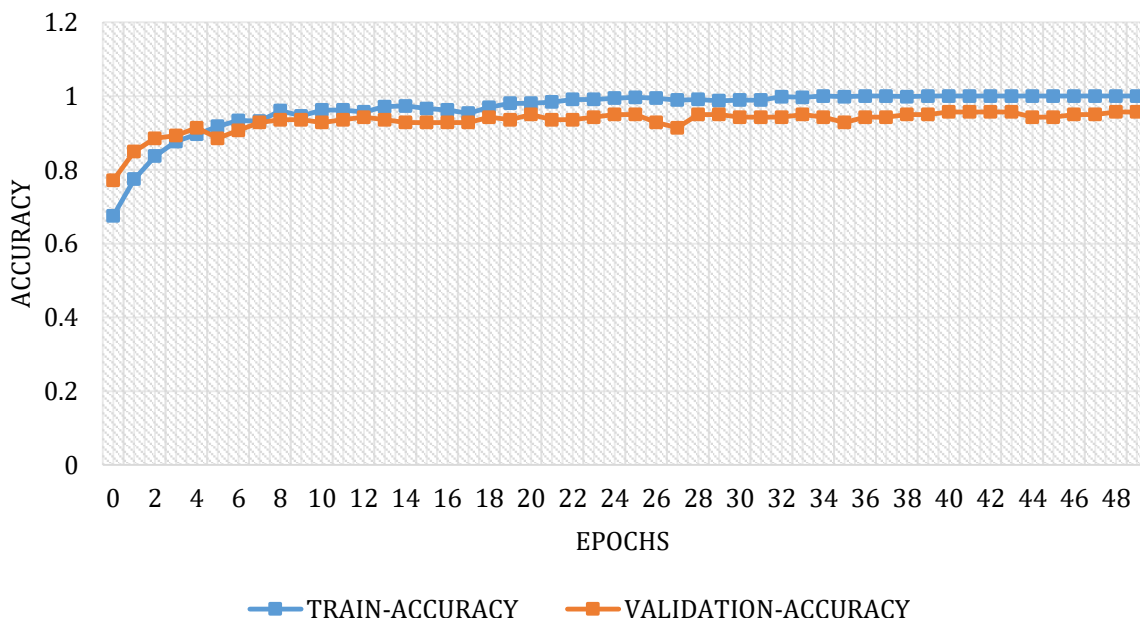


Fig. 10. Accuracy plot of CNN model on GLCM features

		PREDICTED OUTCOME	
		0	1
ACTUAL OUTCOME	0	80	2
	1	4	54

Fig. 11. Confusion Matrix of CNN model on GLCM features

B. AI Model Outcome with Feature Selection

Next, the PSO-selected features are given to the SVM and CNN models to evaluate the importance of features in disease prediction. The GLCM extracts 28 features from the US images of the kidney. The PSO identifies 26 features as more important from the 28 features. These 26 features are given to CNN and SVM for classification. The SVM data sample used for training and testing is the same as the previous case study: 340 for training and 70 for testing the models. By using the PSO features, the SVM model gives a TP of 76, correctly identifying 76 samples as normal, a TN of 55, correctly identifying 55 samples as diseased, an FP of 5, indicating the model wrongly identified 5 diseased images as normal, and an FN of 4, indicating the model wrongly identified 4 diseased images as normal. The confusion matrix of the SVM model using PSO features is given in Figure 12.

		PREDICTED OUTCOME	
		0	1
ACTUAL OUTCOME	0	76	4
	1	5	55

Fig. 12. Confusion Matrix of SVM model on PSO features

The 26 features from the PSO algorithm are given to CNN for classification. The CNN model is trained using 340 images with an epoch set to 50, using accuracy and loss as metrics. Figures 13 and 14 show the loss and accuracy plots of the CNN model using PSO features. The remaining 140 images are given to the CNN model for testing. The PSO-based CNN model correctly identified 75 and 63 US images as normal and diseased, respectively. The model did not make any wrong predictions on diseased images and made 2 wrong predictions on normal images. The CNN model gives TP, TN, FP, and FN of 75, 63, 2, and 0 using PSO features. The confusion matrix of the PSO-based CNN is given in Figure 15.

CNN-PSO MODEL: LOSS PLOT

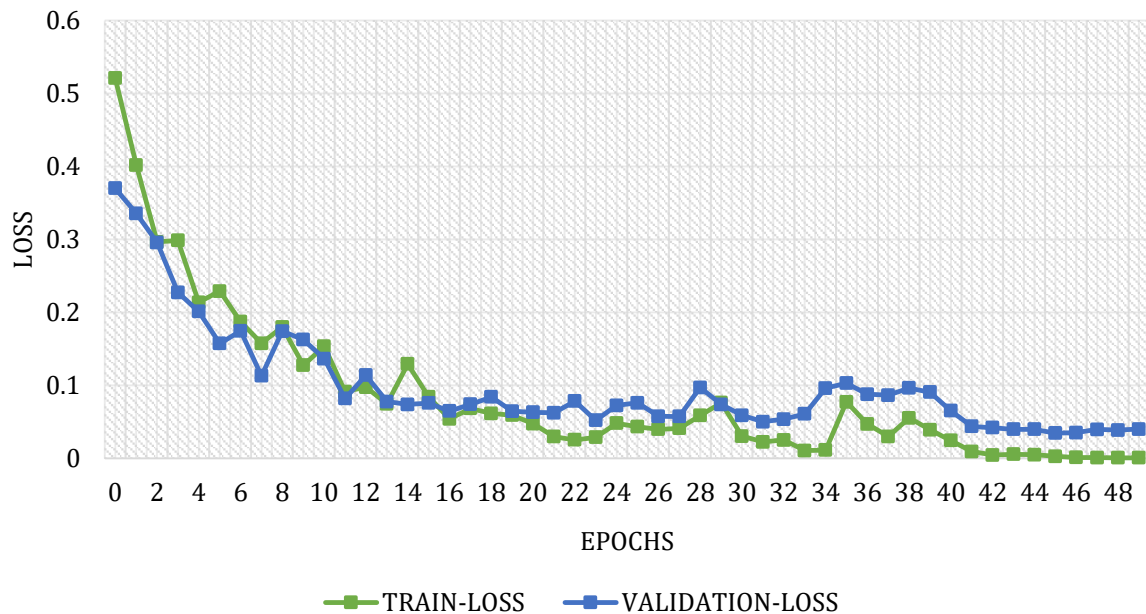


Fig. 13. Loss plot of CNN model on PSO features

CNN-PSO MODEL: ACCURACY PLOT

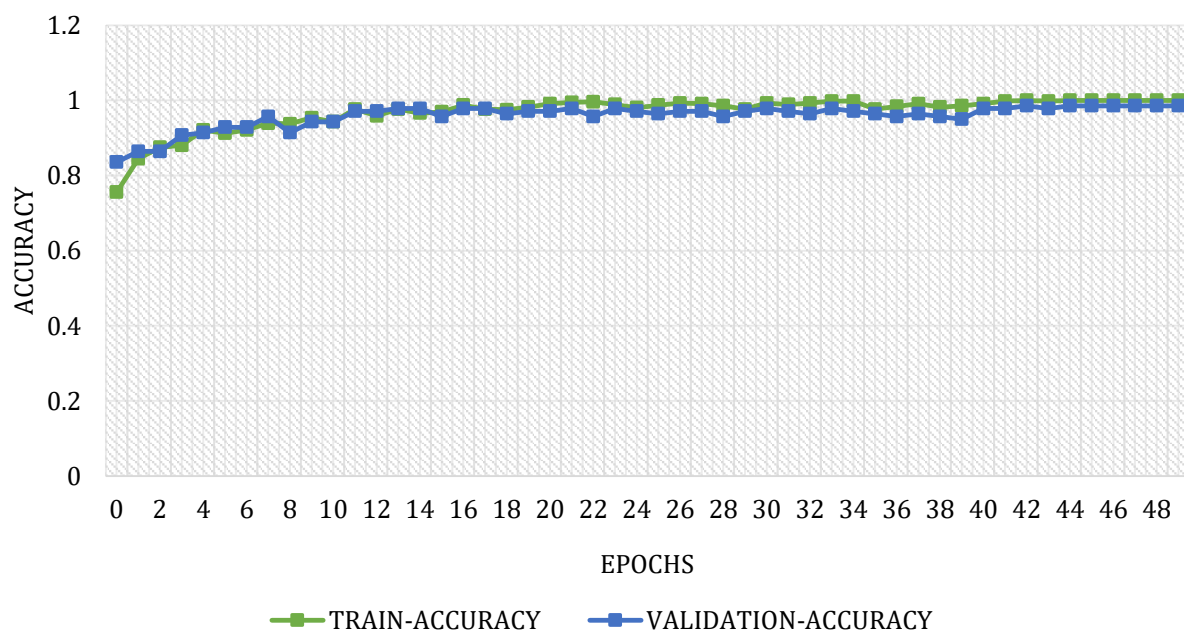


Fig. 14. Accuracy plot of CNN model on PSO features

		PREDICTED OUTCOME	
		0	1
ACTUAL OUTCOME	0	75	0
	1	2	63

Fig. 15. Confusion Matrix of CNN model on PSO features

By using the confusion matrices obtained by SVM and CNN models with GLCM and PSO features, performance metrics such as accuracy, precision, recall, and F1-score are calculated [39, 40]. The formulas used to calculate the metrics are given in Equations (21-24).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad [21]$$

$$Precision = \frac{TP}{TP+FP} \quad Precision = \frac{TP}{TP+FP} \quad [22]$$

$$Recall = \frac{TP}{TP+FN} \quad Recall = \frac{TP}{TP+FN} \quad [23]$$

$$F1 - Score = \frac{TP}{TP + \frac{1}{2}(FP+FN)} \quad F1 - Score = \frac{TP}{TP + \frac{1}{2}(FP+FN)} \quad [24]$$

Table 1 gives the performance measures of SVM and CNN models using GLCM and PSO features. The metrics of recall, precision, and F1-score are displayed separately for each target variable (normal and diseased). By comparing all the values, the CNN model gives the highest accuracy of 98.57% using PSO features. While other methods like SVM using GLCM and PSO features give 90.71% and 93.57% accuracy, the CNN using GLCM features gives 93.57% accuracy. For detecting healthy samples, the PSO-based CNN gives precision, recall, and F1-score of 97, 100, and 99, respectively. For diseased images, the metric values are 100, 97, and 98.

Table 1. Comparison of AI model by features applied

Model		Precision	Recall	F1-Score	Accuracy
GLCM-SVM	Healthy	92%	93%	92%	90.71%
	Disease	89%	88%	89%	
PSO-SVM	Healthy	94%	95%	94%	93.57%
	Disease	93%	92%	92%	
GLCM-CNN	Healthy	95%	98%	96%	95.71

	Disease	96%	93%	95%	
PSO-CNN	Healthy	97%	100%	99%	98.57%
	Disease	100%	97%	98%	

6. CONCLUSION

The research successfully designs an efficient methodology for kidney disease prediction from ultrasound images. The ultrasound images are acquired from health centers, and a database is created with the help of healthcare professionals. The data has some issues, such as different dimensions and limited samples. The images are pre-processed with resizing, grayscale conversion, and augmentation. Features from the processed images are extracted by GLCM. These GLCM features are further fine-tuned by PSO. The outcomes of both feature sets are given to the SVM and CNN models separately and analyzed. Both models show increased accuracy using PSO features compared to GLCM features. The outcomes of SVM and CNN using GLCM features are 90.71% and 95.71%, respectively. The outcomes of both models using PSO features are 93.57% and 98.57%, respectively. This demonstrates the power of optimized features in AI models. Comparing all four combinations, the CNN model with PSO features gives excellent metrics in recall, precision, and F1-score. The results show the promise of the designed methodology for kidney disease prediction from ultrasound images.

The limitation of the research is that the proposed method for kidney disease prediction is not accessible to the general public. To make the research more effective, in future studies, we plan to develop a user-friendly website for deploying the designed AI model, which will provide outputs within seconds.

REFERENCES

- [1] Tanner, George A. "Kidney function." Medical Physiology-Principles for Clinical Medicine. 3rd ed. Philadelphia: Lippincott Williams and Wilkins (2009): 391-418.
- [2] Ali, Ali Adeeb Hussein. "Overview of the vital roles of macro minerals in the human body." Journal of Trace Elements and Minerals 4 (2023): 100076.
- [3] Bonner, Ryan, Oltjon Albajrami, James Hudspeth, and Ashish Upadhyay. "Diabetic kidney disease." Primary Care: Clinics in Office Practice 47, no. 4 (2020): 645-659.
- [4] Yan, Ming-Tso, Chia-Ter Chao, and Shih-Hua Lin. "Chronic kidney disease: strategies to retard progression." International journal of molecular sciences 22, no. 18 (2021): 10084.
- [5] Alnazer, Israa, Pascal Bourdon, Thierry Urruty, Omar Falou, Mohamad Khalil, Ahmad Shahin, and Christine Fernandez-Maloigne. "Recent advances in medical image processing for the evaluation of chronic kidney disease." Medical Image Analysis 69 (2021): 101960.
- [6] Alkurdy, Noor Hamzah, Hadeel K. Aljobouri, and Zainab Kassim Wadi. "Ultrasound renal stone diagnosis based on convolutional neural network and vgg16 features." Int J Electr Comput Eng 13, no. 3 (2023): 3440-8.
- [7] George, Mino, and H. B. Anita. "Kidney Abnormalities Detection and Classification Using CNN-based Feature Extraction." In 2022 4th International Conference on Circuits, Control, Communication and Computing (I4C), pp. 359-362. IEEE, 2022.
- [8] Mittal, Harshit. "Kidney CT image analysis using CNN." Computer Science & Information Technology (CS & IT) ISSN (2023): 2231-5403.
- [9] Chen, Gongping, Jingjing Yin, Yu Dai, Jianxun Zhang, Xiaotao Yin, and Liang Cui. "A novel convolutional neural network for kidney ultrasound images segmentation." Computer methods and programs in biomedicine 218 (2022): 106712.
- [10] Chaki, Jyotismita, and Aysegül Ucar. "An efficient and robust approach using inductive transfer-based ensemble deep neural networks for kidney stone detection." IEEE Access (2024).
- [11] Nallarasana, V., Vijayakumar Ponnusamy, R. Lakshminarayanan, S. Vigneshwari, and R. Vinoth. "Prediction of Kidney Disease Utilizing a Hybrid Deep Learning Methodology." In 2024 2nd International Conference on Computer, Communication and Control (IC4), pp. 1-8. IEEE, 2024.
- [12] Saif, Dina, Amany M. Sarhan, and Nada M. Elshennawy. "Early prediction of chronic kidney disease based on ensemble of deep learning models and optimizers." Journal of electrical systems and information technology 11, no. 1 (2024): 17.
- [13] Preethi, I., K. Dharmarajan, Bhisham Sharma, Subrata Chowdhury, and Imad Ben Dhaou. "A Novel Method to Predict Chronic Kidney Disease using Optimized Deep Learning Algorithm." In 2024 21st Learning and

Technology Conference (L&T), pp. 313-318. IEEE, 2024.

- [14] Rahman, Md Mustafizur, Md Al-Amin, and Jahangir Hossain. "Machine learning models for chronic kidney disease diagnosis and prediction." *Biomedical Signal Processing and Control* 87 (2024): 105368.
- [15] Talebi, Hossein, and Peyman Milanfar. "Learning to resize images for computer vision tasks." In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 497-506. 2021.
- [16] Kanan, Christopher, and Garrison W. Cottrell. "Color-to-grayscale: does the method matter in image recognition?." *PloS one* 7, no. 1 (2012): e29740.
- [17] Moreno-Barea, Francisco J., José M. Jerez, and Leonardo Franco. "Improving classification accuracy using data augmentation on small data sets." *Expert Systems with Applications* 161 (2020): 113696.
- [18] Chen, Rung-Ching, Christine Dewi, Su-Wen Huang, and Rezzy Eko Caraka. "Selecting critical features for data classification based on machine learning methods." *Journal of Big Data* 7, no. 1 (2020): 52.
- [19] Dhal, Pradip, and Chandrashekhar Azad. "A comprehensive survey on feature selection in the various fields of machine learning." *Applied Intelligence* 52, no. 4 (2022): 4543-4581.
- [20] Mohanaiah, P., P. Sathyanarayana, and L. GuruKumar. "Image texture feature extraction using GLCM approach." *International journal of scientific and research publications* 3, no. 5 (2013): 1-5.
- [21] Haralick, Robert M., Karthikeyan Shanmugam, and Its' Hak Dinstein. "Textural features for image classification." *IEEE Transactions on systems, man, and cybernetics* 6 (1973): 610-621.
- [22] Zubair, Abdul Rasak, and Oluwaseun Adewunmi Alo. "Grey level co-occurrence matrix (GLCM) based second order statistics for image texture analysis." *arXiv preprint arXiv:2403.04038* (2024).
- [23] Özkan, Kürşad, Ahmet Mert, and Serkan Özdemir. "A new proposed GLCM texture feature: modified Rényi Deng entropy." *The Journal of Supercomputing* 79, no. 18 (2023): 21507-21527.
- [24] Eberhart, Russell, and James Kennedy. "A new optimizer using particle swarm theory." In *MHS'95. Proceedings of the sixth international symposium on micro machine and human science*, pp. 39-43. Ieee, 1995.
- [25] Freitas, Diogo, Luiz Guerreiro Lopes, and Fernando Morgado-Dias. "Particle swarm optimisation: a historical review up to the current developments." *Entropy* 22, no. 3 (2020): 362.
- [26] Houssein, Essam H., Ahmed G. Gad, Kashif Hussain, and Ponnuthurai Nagaratnam Suganthan. "Major advances in particle swarm optimization: theory, analysis, and application." *Swarm and Evolutionary Computation* 63 (2021): 100868.
- [27] Martí, Luis, Jesús García, Antonio Berlanga, and José M. Molina. "A stopping criterion for multi-objective optimization evolutionary algorithms." *Information Sciences* 367 (2016): 700-718.
- [28] Tzotsos, Angelos, and Demetre Argialas. "Support vector machine classification for object-based image analysis." *Object-based image analysis: Spatial concepts for knowledge-driven remote sensing applications* (2008): 663-677.
- [29] Xia, Shu-yin, Zhong-yang Xiong, Yue-guo Luo, and Li-mei Dong. "A method to improve support vector machine based on distance to hyperplane." *Optik* 126, no. 20 (2015): 2405-2410.
- [30] Chandra, Mayank Arya, and S. S. Bedi. "Survey on SVM and their application in image classification." *International Journal of Information Technology* 13, no. 5 (2021): 1-11.
- [31] Elngar, Ahmed A., Mohamed Arafa, Amar Fathy, Basma Moustafa, Omar Mahmoud, Mohamed Shaban, and Nehal Fawzy. "Image classification based on CNN: a survey." *Journal of Cybersecurity and Information Management* 6, no. 1 (2021): 18-50.
- [32] Shyam, Radhey. "Convolutional neural network and its architectures." *Journal of Computer Technology & Applications* 12, no. 2 (2021): 6-14p.
- [33] Kuo, Chien-Hao, Yang-Ho Chou, and Pao-Chi Chang. "Using deep convolutional neural networks for image retrieval." *Electronic Imaging* 28 (2016): 1-6.
- [34] Zhao, Qi, Shuchang Lyu, Boxue Zhang, and Wenquan Feng. "Multiactivation pooling method in convolutional neural networks for image recognition." *Wireless Communications and Mobile Computing* 2018, no. 1 (2018): 8196906.
- [35] Zafar, Afia, Muhammad Aamir, Nazri Mohd Naw, Ali Arshad, Saman Riaz, Abdulrahman Alruban, Ashit Kumar Dutta, and Sultan Almotairi. "A comparison of pooling methods for convolutional neural networks." *Applied Sciences* 12, no. 17 (2022): 8643.
- [36] Agarap, Abien Fred. "Deep learning using rectified linear units (relu)." *arXiv preprint arXiv:1803.08375* (2018).

- [37] Yang, Jing, and Guanci Yang. "Modified convolutional neural network based on dropout and the stochastic gradient descent optimizer." *Algorithms* 11, no. 3 (2018): 28.
 - [38] Yan, Ming, Junjie Chen, Xiangyu Zhang, Lin Tan, Gan Wang, and Zan Wang. "Exposing numerical bugs in deep learning via gradient back-propagation." In *Proceedings of the 29th ACM joint meeting on european software engineering conference and symposium on the foundations of software engineering*, pp. 627-638. 2021.
 - [39] Ferrer, Luciana. "Analysis and comparison of classification metrics." *arXiv preprint arXiv:2209.05355* (2022).
 - [40] Yacouby, Reda, and Dustin Axman. "Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models." In *Proceedings of the first workshop on evaluation and comparison of NLP systems*, pp. 79-91. 2020.
-