ISSN 2277-2685
IJESR/Jan. 2024/ Vol-14/Issue-1/1-14
B.Durga Bhavani/ International Journal of Engineering & Science Research

Deep Learning Approach for Renal Image Classification for Deciphering Anomalies in Kidney Structures

B.Durga Bhavani[1], G. Sanjana[2], K. Sejal[2], K. Nandhini[2]
[1]Assistant Professor,[2]UG Students, Department of Information Technology
[1,2]Malla Reddy Engineering College for Women, Maisammaguda, Dhulapally, Kompally,
Secunderabad-500100, Telangana, India.

ABSTRACT

Renal diseases pose a significant health challenge worldwide, with conditions like cysts, stones, and tumors affecting kidney structures. Timely and accurate diagnosis is critical for effective treatment and management. Traditional systems for renal image analysis often rely on manual segmentation and feature extraction, followed by the application of machine learning algorithms. However, these methods are limited by their dependence on handcrafted features and may not adapt well to the inherent complexity and variability in medical images. The time and expertise required for manual analysis also contribute to delays in diagnosis. The integration of deep learning in medical image analysis is a relatively recent development. Over the past decade, there has been a surge in research and applications leveraging deep neural networks for various medical imaging tasks. The success of deep learning models in computer vision, combined with the increasing availability of large medical image datasets, has paved the way for the application of these techniques in renal image classification. Therefore, this research proposes a novel deep learning-based approach for renal image classification to enhance the precision and efficiency of diagnostic procedures. Our proposed system leverages deep neural networks, specifically convolutional neural networks (CNNs), to automatically classify renal images into different categories, providing rapid and accurate results. The drawbacks of conventional systems, such as inter-observer variability, limited scalability, and the potential for misdiagnosis, can be significantly mitigated through our deep learning approach. We trained our model on a large dataset of annotated renal images, encompassing various anomalies and normal tissues, to ensure robust performance. Preliminary results indicate high accuracy, sensitivity, and specificity in identifying kidney anomalies, making our proposed system a promising tool for improving the diagnostic process in the field of nephrology.

Keywords: renal image, kidney, anomalies, convolutional neural network, Deep learning ai.

1. INTRODUCTION

In the realm of nephrology, the detection and classification of renal anomalies play a pivotal role in effective patient care. Renal diseases, ranging from cysts and stones to tumors, present substantial health challenges globally. Traditional methods for analyzing renal images involve labor-intensive processes like manual segmentation and feature extraction, followed by the application of machine learning algorithms. However, these approaches are constrained by their reliance on manually crafted features and struggle to adapt to the intricate and variable nature of medical images. Additionally, the time-consuming nature of manual analysis contributes to diagnostic delays [1].In response to these limitations, the integration of deep learning into medical image analysis has emerged as a progressive avenue. Over the last decade, there has been a surge in research exploring the potential of deep neural networks for diverse medical imaging tasks. The success of deep learning models in computer vision, coupled with the growing availability of extensive medical image datasets, has paved the way for their application in renal image classification.This [2] research puts forth a groundbreaking deep learning-based approach designed to enhance the precision and efficiency of renal diagnostic procedures. Leveraging deep neural networks, specifically convolutional neural networks (CNNs), our proposed system automates the classification of renal images into distinct categories, providing swift and accurate results. By relying on deep learning, we aim to address the drawbacks of conventional systems, including issues like inter-observer variability, limited scalability, and the potential for misdiagnosis.Our model has undergone training on a sizable dataset comprising annotated renal images that encompass various anomalies and normal tissues. This comprehensive training is crucial for ensuring the robust performance of the system. Preliminary results from our research showcase promising outcomes, with the model demonstrating high accuracy, sensitivity, and specificity in identifying kidney anomalies. Consequently, our proposed deep learning-based system emerges as a potential breakthrough, holding significant promise for advancing the diagnostic process within the field of nephrology.The global challenge presented by renal diseases, including cysts, stones, and tumors affecting kidney structures, underscores the critical need for timely and accurate diagnostic procedures. Traditional methods of renal image analysis heavily rely on manual segmentation and feature extraction, followed by the application of machine learning algorithms [3]. However, these approaches are hindered by their dependency on handcrafted features, leading to limitations in adapting to the inherent complexity and variability of medical images. Moreover, the labor-intensive nature of manual analysis contributes to significant delays in diagnosis, impacting effective treatment and management.This [4] research addresses the aforementioned challenges by proposing a novel deep learning-based approach for renal image classification, aimed at enhancing the precision and efficiency of diagnostic procedures. Our proposed system harnesses the power of deep neural networks, specifically employing convolutional neural networks (CNNs), to automate the classification of renal images into distinct categories, thereby providing swift and accurate results. The limitations associated with conventional systems, such as inter-observer variability, limited scalability, and the potential for misdiagnosis, can be significantly alleviated through the implementation of our deep learning approach.To ensure the robust performance of our system, we conducted extensive training on a large dataset comprising annotated renal images, encompassing a diverse range of anomalies and normal tissues. Preliminary results from our research demonstrate promising outcomes, with the model exhibiting high accuracy, sensitivity, and specificity in identifying kidney anomalies. As such, our proposed deep learning-based system emerges as a potent tool with the potential to substantially improve the diagnostic process in the field of nephrology.Traditional [5] systems heavily rely on manual processes, such as segmentation and feature extraction,followed by the application of machine learning algorithms. However, the dependency on handcrafted features in these methods hinders their adaptability to the inherent complexity and variability found in medical images. Moreover, the manual nature of analysis demands a significant investment of time and expertise, contributing to delays in diagnosis.The integration of deep learning into medical image analysis emerges as a pivotal development, motivated by the transformative success observed in various computer vision tasks over the past decade. The availability of large medical image datasets further fuels the exploration of deep neural networks for applications in renal image classification.The [6] primary motivation behind this research is to propose a novel deep learning-based approach tailored specifically for renal image classification, aiming to address the limitations of traditional systems. By leveraging deep neural networks, particularly convolutional neural networks (CNNs), our proposed system seeks to automate the classification of renal images into distinct categories, offering a rapid and accurate diagnostic solution. The inherent drawbacks of conventional systems, such as inter-observer variability, limited scalability, and the potential for misdiagnosis, are targeted for significant mitigation through the implementation of our deep learning approach.

2. LITERATURE SURVEY

In research work [7], the data mining technique applied to specific analysis of clinical records is a good method. The performance of the decision tree method was 91% (accuracy) compared to the Naïve Bayesian method. The classification algorithm for diabetes dataset had 94% specificity and 95% sensitivity. They also found that mining helps retrieve correlations of attributes that are no longer direct indicators of the type they are trying to predict. Similar work still needs to be done to improve the overall performance of prediction engine accuracy in the statistical analysis of neural networks and clustering algorithms.

In [8], the authors described the prediction models using machine learning techniques including K-nearest neighbor (KNN), support vector machine (SVM), logistic regression (LR), and decision tree classifiers for CKD prediction. From the experiment, it was concluded that the SVM classifier provides the highest accuracy, 98.3%. SVM has the absolute best sensitivity after training and testing performed with the proposed method. Therefore, according to this comparison, it could be concluded that an SVM classifier is used to predict persistent kidney disease.

In the paper [9], they chose four different algorithms and compared them to get an accurate expectation rate over the dataset. Unlike all approaches that were presented, they got the best results from the gradient boosting classifier. The models effectively achieve an accuracy rate of 99.80%, whereas AdaBoost and LDA achieve 97.91% at a low value. Also, the gradient boosting ML classifier takes much time to make the prediction compared to others and has a higher predictable value in both the curves (ROC and AUC). Hence, an accurate expectation undoubtedly depends on the preprocessing strategy, and the methods of preprocessing must be approached cautiously to precisely achieve recognized results.

In [10], the authors investigated the machine learning ability, which is supported by predictive analysis so as to predict CKD early. An experimental procedure was performed by considering a dataset of 400 cases collected by Apollo Hospitals India. In this article, two labels were used as output/targets in this hybrid model (i.e., patients having CKD and others who are healthy) and four different machine learning classifiers were implemented. On the comparison of these classifiers, the classification along with regression tree, and the RPART classification model, showed remarkably better results in terms of accuracy. They used the information gain quotient for excruciating criterion, and here the optimum spilling reduces the noise of the resulting feature subsets. In this study, the RPART limited value of criterion for the splitting was five, meaning that splits repeatedly occur for the five instances present in the leaf node. In addition, they identified an equivalent previous probability for the class attributes. Here, the RPART prediction model used seven terminal nodes for the earlier predictions of CKD. The experimental results showed that the highest AUC and TPR were obtained with the machine learning prediction model, whereas the highest TNR (1.00) was achieved with the model RPART. The RPART model could be described as a set of rules for making the decision. However, the major drawback of RPART is the consideration of the single factor as a parameter in every division procedure, while considering different parameter combinations could result in better CKD predictions. However, the machine learning prediction model gives the lowest error rate. The major reason is that the MLP could adopt and handle complex predictions. The complex relationships require hidden nodes and they are useful as they allow neural networks to model between parameters while sometimes deal with nonlinearity in data. The overall results indicate that the algorithms of machine learning give an inspiring and a feasible methodology for earlier CKD prediction.

As we have already seen, there are different machine learning prediction models and learning programs available to assist practitioners. In [11], they used a new selection guide for predicting CKD. In this work, CKD is predicted by using specific classifiers and a reasonable study of overall performance. In this study, they performed the evaluation of the Naïve Bayes classifier, random forest, and artificial neural network classifiers and concluded that the random forest classifier performs better as compared to other classifiers. The worth of forecasting CKD has been progressive. Several sustainable evolutionary policies can be used to improve the outcomes of the suggested classifiers. Here, Naïve Bayes, random forest, and KNN were applied to predict CKD. Early diagnosis of CKD helps to treat those affected well in time and prevent the disease from progressing to worse stage. The early detection of this type of disease and well-timed treatment is one of the main objectives of the medical field.

In [12], a machine learning prediction model was developed for the early prediction of CKD. The dataset gives input features gathered from the CKD dataset and the models were tested and validated for the given input features. Machine learning decision tree classifier, random forest classifier, and support vector classifier were constructed for the diagnosis of CKD. The performance analysis of the models was assessed on the basis of the accuracy score of the prediction model. On comparison, the results of the research showed that the random forest classifier model performs much better at predicting CKD as compared to decision tree and support vector classifiers.

The kidneys play a vital role in maintaining the body’s blood pressure, acid-base sense of balance, and electrolyte sense of balance, not only needed to filter toxins from the body. Malfunction is accountable for insignificant to mortal illnesses, in addition to dysfunction in the other body organs. Therefore, researchers all over the world have dedicated themselves for finding techniques to accurately diagnose and effectively treat chronic kidney disease. As machine learning classifiers are increasingly used in the medical field for diagnosis, now CKD is also included in the list of diseases that could be predicted using machine learning classifiers. The research to detect CKD with ML algorithms has enhanced the procedure and consequence accuracy progressively. They proposed the random forest classifier (99.75% accuracy) as the maximum efficient classifier among all other classifiers. The study demonstrates the effective handling of missing values in data through four techniques, namely, mode, mean, median, and zero-point methods. It also evaluates the performance of machine learning models under two scenarios, with and without tuning the hyperparameters, and observes significant improvement in the classifiers' performance, which is visually presented through graphs [13].

Overall, the motive of the study is to examine the applicability of specific supervised machine learning classifiers in the field of bioinformatics and offer their compatibility in detecting several serious diseases such as the diagnosis of CKD at an early stage [14].

In [15], the authors predicted CKD based on sugar levels, aluminum levels, and red blood cell percentage. In this perception, five classifiers were applied, namely, Naïve Bayes, logistic regression, decision table, random tree, and random forest, and for each classifier, the results were noted based on (i) without preprocessing, (ii) SMOTE with resampling, and (iii) class equalizer. Random forest classifier has been observed to give the highest accuracy at 98.93% in SMOTE with resampling.

3. PROPOSED SYSTEM

3.1 Overview

This code is graphical user interface (GUI) application built using the Tkinter library in Python. The application is focused on AI-driven renal image classification for deciphering anomalies in kidney structures

- Importing Libraries:
  The code imports various libraries such as Tkinter (for GUI), Matplotlib (for plotting), NumPy, Pandas, OpenCV, joblib, seaborn, and others. It uses machine learning libraries like Keras for deep learning and scikit-learn for machine learning models.
- GUI Initialization:
  The Tkinter main window is created with the title "AI-Driven Renal Image Classification for Deciphering Anomalies in Kidney Structures" and a size of 1300x1200 pixels.
- Global Variables:
  Global variables are defined to store information such as the dataset filename, image data (X), labels (Y), trained model, accuracy, and a Random Forest classifier.
- Functions:
  Several functions are defined to perform specific tasks:
    - getID: Assigns numerical labels based on the directory name.
    - uploadDataset: Allows the user to select a dataset directory.
    - imageProcessing: Processes and normalizes images, providing information about the dataset.
    - gnb: Implements Gaussian Naive Bayes classification, displaying accuracy and a confusion matrix.
    - cnnModel: Builds and trains a Convolutional Neural Network (CNN) model, saving the model and displaying accuracy.
    - predict: Uploads a test image, classifies it using the trained model, and displays the result.
    - graph: Displays a performance graph based on accuracy and loss.
    - close: Closes the application.
- GUI Components:
  Buttons for various actions (upload dataset, image processing, GNB classification, CNN model building, image classification, performance graph, and exit).Text widget to display information, results, and messages.Labels and title for the GUI components.
- Main Loop:
  The Tkinter main loop (main.mainloop()) is executed, allowing the user to interact with the GUI.

Proposed method:

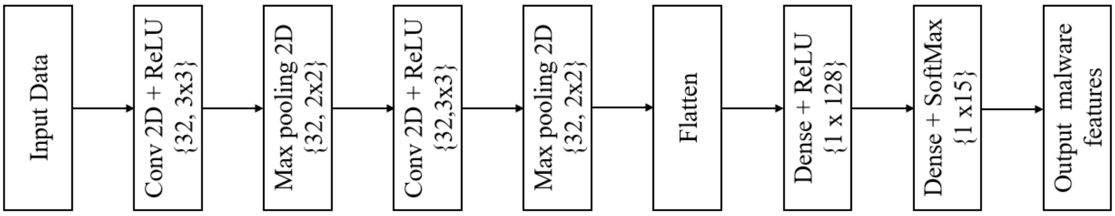


Figure:1

3.2 Data preprocessing

Image preprocessing is a critical step in computer vision and image analysis tasks. It involves a series of operations to prepare raw images for further processing by algorithms or neural networks. Here's an explanation of each step-in image preprocessing:

Step 0. Image Read: The first step in image preprocessing is reading the raw image from a source, typically a file on disk. Images can be in various formats, such as JPEG, PNG, BMP, or others. Image reading is performed using libraries or functions specific to the chosen programming environment or framework. The result of this step is a digital representation of the image that can be manipulated programmatically.

1. Image Resize: Image resize is a common preprocessing step, especially when working with machine learning models or deep neural networks. It involves changing the dimensions (width and height) of the image. Resizing can be necessary for several reasons:

- Ensuring uniform input size: Many machine learning models, especially convolutional neural networks (CNNs), require input images to have the same dimensions. Resizing allows you to standardize input sizes.
- Reducing computational complexity: Smaller images require fewer computations, which can be beneficial for faster training and inference.
- Managing memory constraints: In some cases, images need to be resized to fit within available memory constraints.

When resizing, it's essential to maintain the aspect ratio to prevent image distortion. Typically, libraries like OpenCV or Pillow provide convenient functions for resizing images.

2. Image to Array: In this step, the image is converted into a numerical representation in the form of a multidimensional array or tensor. Each pixel in the image corresponds to a value in the array. The array is usually structured with dimensions representing height, width, and color channels (if applicable).

For grayscale images, the array is 2D, with each element representing the intensity of a pixel. For color images, it's a 3D or 4D array, with dimensions for height, width, color channels (e.g., Red, Green, Blue), and potentially batch size (if processing multiple images simultaneously).

The conversion from an image to an array allows for numerical manipulation and analysis, making it compatible with various data processing libraries and deep learning frameworks like NumPy or TensorFlow.

3. Image to Float32: Most machine learning and computer vision algorithms expect input data to be in a specific data type, often 32-bit floating-point numbers (float32). Converting the image array to float32 ensures that the pixel values can represent a wide range of intensities between 0.0 (black) and 1.0 (white) or sometimes between -1.0 and 1.0, depending on the specific normalization used.

This step is essential for maintaining consistency in data types and enabling compatibility with various machine learning frameworks and libraries. It's typically performed by dividing the pixel values by the maximum intensity value (e.g., 255 for an 8-bit image) to scale them to the [0.0, 1.0] range.

4. Image to Binary: Image binarization is a process of converting a grayscale image into a binary image, where each pixel is represented by either 0 (black) or 1 (white) based on a specified threshold. Binarization is commonly used for tasks like image segmentation, where you want to separate objects from the background.

The process involves setting a threshold value, and then for each pixel in the grayscale image, if the pixel value is greater than or equal to the threshold, it is set to 1; otherwise, it is set to 0.

Binarization simplifies the image and reduces it to essential information, which can be particularly useful in applications like character recognition or object tracking, where you need to isolate regions of interest.

3.3 Splitting the Dataset

In machine learning data preprocessing, we divide our dataset into a training set and test set. This is one of the crucial steps of data preprocessing as by doing this, we can enhance the performance of our machine learning model. Suppose if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models. If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So we always try to make a machine learning model which performs well with the training set and also with the test dataset. Here, we can define these datasets as:

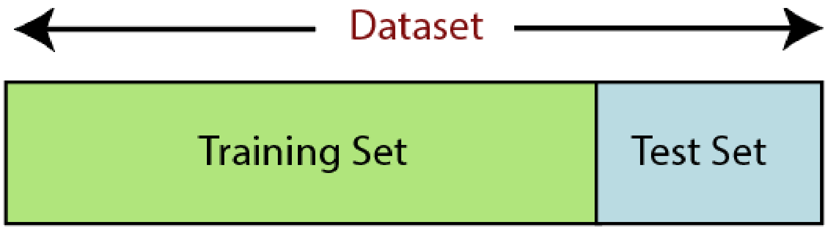


Figure 2: Splitting the dataset.

Training Set: A subset of dataset to train the machine learning model, and we already know the output.

Test set: A subset of dataset to test the machine learning model, and by using the test set, model predicts the output.

For splitting the dataset, we will use the below lines of code:

```
from sklearn.model_selection import train_test_split

x_train, x_test, y_train, y_test= train_test_split(x, y, test_size= 0.2, random_state=0)
```

Explanation

- In the above code, the first line is used for splitting arrays of the dataset into random train and test subsets.
- In the second line, we have used four variables for our output that are
- x_train: features for the training data
- x_test: features for testing data
- y_train: Dependent variables for training data
- y_test: Independent variable for testing data

- In train_test_split() function, we have passed four parameters in which first two are for arrays of data, and test_size is for specifying the size of the test set. The test_size maybe .5, .3, or .2, which tells the dividing ratio of training and testing sets.
- The last parameter random_state is used to set a seed for a random generator so that you always get the same result, and the most used value for this is 42.

CNN Basics

According to the facts, training and testing of proposed model involves in allowing every source image via a succession of convolution layers by a kernel or filter, rectified linear unit (ReLU), max pooling, fully connected layer and utilize SoftMax layer with classification layer to categorize the objects with probabilistic values ranging from [0,1]. Convolution layer as is the primary layer to extract the features from a source image and maintains the relationship between pixels by learning the features of image by employing tiny blocks of source data. It's a mathematical function which considers two inputs like source image $I(x, y, d)$ where x and y denotes the spatial coordinates i.e., number of rows and columns. d is denoted as dimension of an image (here $d = 3$, since the source image is RGB) and a filter or kernel with similar size of input image and can be denoted as $F\left(k_x, k_y, d\right)$.

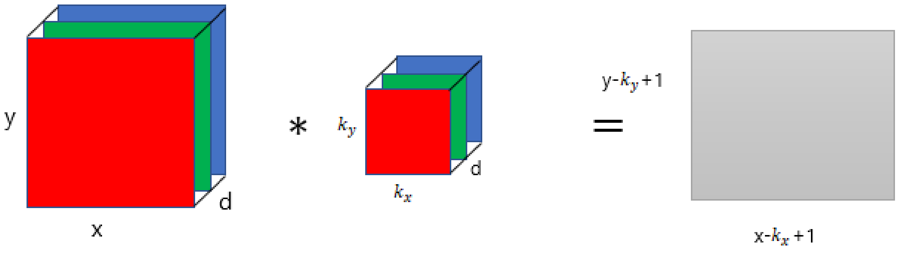
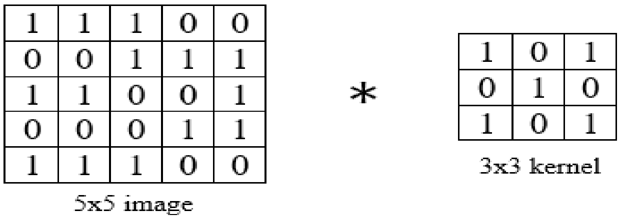


Figure 3: Representation of convolution layer process.

The output obtained from convolution process of input image and filter has a size of $C\left(\left(x - k_x + 1\right), \left(y - k_y + 1\right), 1\right)$, which is referred as feature map. Let us assume an input image with a size of 5×5 and the filter having the size of 3×3. The feature map of input image is obtained by multiplying the input image values with the filter values.



1	1	1	0	0
0	0	1	1	1
1	1	0	0	1
0	0	0	1	1
1	1	1	0	0

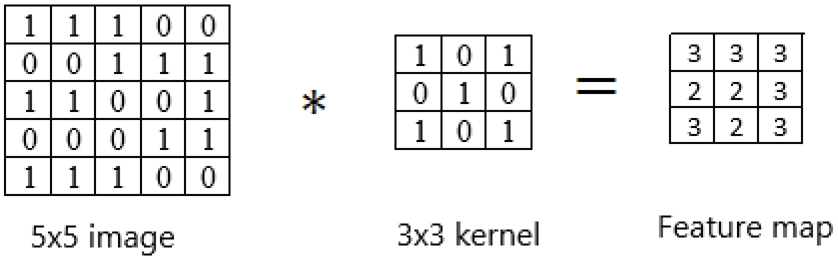
5x5 image

*

1	0	1
0	1	0
1	0	1

3x3 kernel

(a)



1	1	1	0	0
0	0	1	1	1
1	1	0	0	1
0	0	0	1	1
1	1	1	0	0

5x5 image

*

1	0	1
0	1	0
1	0	1

3x3 kernel

=

3	3	3
2	2	3
3	2	3

Feature map

(b)

Figure. 4 Example of convolution layer process (a) an image with size $\mathbf{5 \times 5}$ is convolving with $\mathbf{3 \times 3}$ kernel (b) Convolved feature map

3.3.1 ReLU layer

Networks those utilizes the rectifier operation for the hidden layers are cited as rectified linear unit (ReLU). This ReLU function $\mathcal{G}(\cdot)$ is a simple computation that returns the value given as input directly if the value of input is greater than zero else returns zero. This can be represented as mathematically using the function $max(\cdot)$ over the set of 0 and the input x as follows:

$$\mathcal{G}(x) = \max\{0, x\}$$

3.3.2 Max pooing layer

This layer mitigates the number of parameters when there are larger size images. This can be called as subsampling or down sampling that mitigates the dimensionality of every feature map by preserving the important information. Max pooling considers the maximum element form the rectified feature map.

$$LOSS = np.sum(-Y \ * \ np.log(Y\_pred))$$

4. RESULTS AND DISCUSSION

4.1 Implementation description:

This code appears to be a graphical user interface (GUI) application built using the Tkinter library in Python. The application is focused on AI-driven renal image classification for deciphering anomalies in kidney structures. Here's a breakdown of the key components and functionalities:

- Importing Libraries:
  The code imports various libraries such as Tkinter (for GUI), Matplotlib (for plotting), NumPy, Pandas, OpenCV, joblib, seaborn, and others. It uses machine learning libraries like Keras for deep learning and scikit-learn for machine learning models.
- GUI Initialization:
  The Tkinter main window is created with the title "AI-Driven Renal Image Classification for Deciphering Anomalies in Kidney Structures" and a size of 1300x1200 pixels.
- Global Variables:
  Global variables are defined to store information such as the dataset filename, image data (X), labels (Y), trained model, accuracy, and a Random Forest classifier.
- Functions:
  Several functions are defined to perform specific tasks:
  getID: Assigns numerical labels based on the directory name.
  uploadDataset: Allows the user to select a dataset directory.
  imageProcessing: Processes and normalizes images, providing information about the dataset.
  gnb: Implements Gaussian Naive Bayes classification, displaying accuracy and a confusion matrix.
  cnnModel: Builds and trains a Convolutional Neural Network (CNN) model, saving the model and displaying accuracy.
  predict: Uploads a test image, classifies it using the trained model, and displays the result.

- graph: Displays a performance graph based on accuracy and loss.

- close: Closes the application.
- GUI Components:
  Buttons for various actions (upload dataset, image processing, GNB classification, CNN model building, image classification, performance graph, and exit).Text widget to display information, results, and messages.Labels and title for the GUI components.
- Main Loop:
  The Tkinter main loop (main.mainloop()) is executed, allowing the user to interact with the GUI.

4.2 Dataset Description:

The dataset contains total of 5153 images with 1313 images in Normal class and 1345 images in Stone class,1250 Cyst images and 1245 Tumour Images

Table 1: Dataset description.

S. No.	Number of images	Class type
1	1313	Normal
2	1345	Stone
3	1250	Cyst
4	1245	Tumour

Figure 5: Sample images from dataset with galaxy

Figure 6: Sample images from dataset with Tumour class.

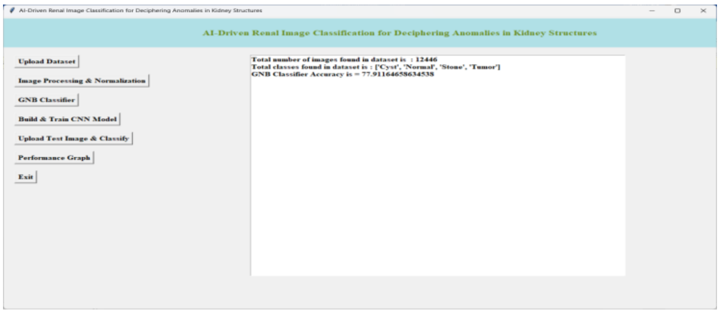
Figure 7: Illustrating GUI application of proposed renal image classification system.

Figure 7 in the project titled "Deep Learning Approach for Renal Image Classification for Deciphering Anomalies in Kidney Structure" depicts the graphical user interface (GUI) application designed for the proposed renal image classification system based on Convolutional Neural Networks (CNN). The purpose of this GUI is to provide an interactive and user-friendly platform for users to input renal images, perform classification, and visualize the results.

```
dataset loaded
```

Figure 8: GUI application after loading the dataset.

Figure 8 in the project titled "Deep Learning Approach for Renal Image Classification for Deciphering Anomalies in Kidney Structure" illustrates the graphical user interface (GUI) application after loading the dataset. This phase of the GUI is crucial as it represents the initial step in the workflow where the user has inputted the dataset containing renal images into the system. The figure likely showcases various elements and functionalities related to dataset handling and preparation.

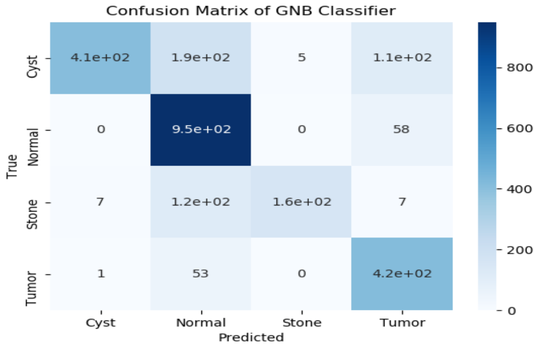


```
Total number of images found in dataset is  : 12446
Total classes found in dataset is : ['Cyst', 'Normal', 'Stone', 'Tumor']
GNB Classifier Accuracy is = 77.91164658634538
```

Figure 9: GUI application after applying GNB classifier

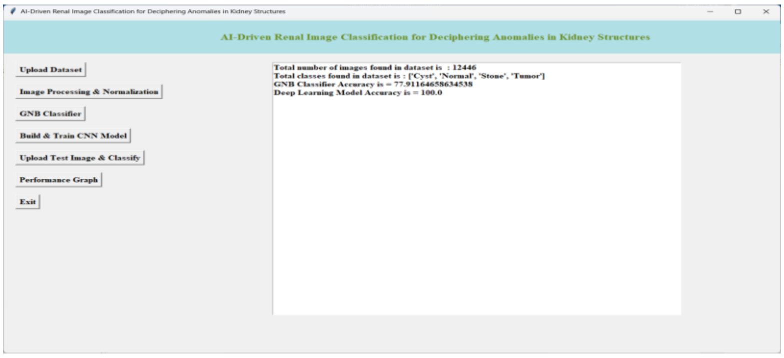
Figure 9 in the project titled "Deep Learning Approach for Renal Image Classification for Deciphering Anomalies in Kidney Structure" depicts the graphical user interface (GUI) application after the application of the Gaussian Naive Bayes (GNB) classifier specifically designed for the Convolutional

Neural Network (CNN) model. The figure provides a snapshot of the GUI, showcasing the outcomes and insights obtained from incorporating the GNB classifier into the renal image classification system.



True \ Predicted	Cyst	Normal	Stone	Tumor
Cyst	4.1e+02	1.9e+02	5	1.1e+02
Normal	0	9.5e+02	0	58
Stone	7	1.2e+02	1.6e+02	7
Tumor	1	53	0	4.2e+02

Figure 10: Confusion matrix of GNB classifier for renal image classification system.



```
Total number of images found in dataset is  : 12446
Total classes found in dataset is : ['Cyst', 'Normal', 'Stone', 'Tumor']
GNB Classifier Accuracy is = 77.91164658634538
Deep Learning Model Accuracy is = 100.0
```

Figure 11: GUI application after applying proposed deep learning approach

Figure 11 in the project titled "Deep Learning Approach for Renal Image Classification for Deciphering Anomalies in Kidney Structure" illustrates the graphical user interface (GUI) application after applying the proposed deep learning approach specifically designed for the Convolutional Neural Network (CNN) model. This figure provides a snapshot of the GUI, presenting the outcomes and results obtained from implementing the proposed method.

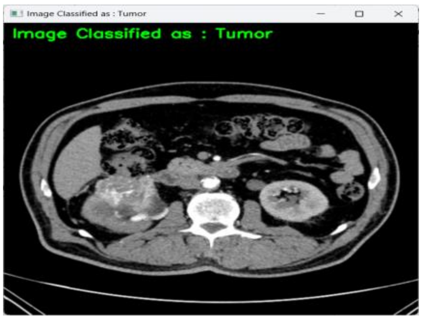


Image Classified as : Tumor

Image Classified as : Tumor

Figure 12: Sample prediction on test renal images using proposed deep learning approach.

Figure 12 in the project titled "Deep Learning Approach for Renal Image Classification for Deciphering Anomalies in Kidney Structure" provides a visual representation of sample predictions on test renal images using the proposed deep learning approach, specifically tailored for the Convolutional Neural Network (CNN) model. This figure offers a glimpse into how the model performs on real-world test data.

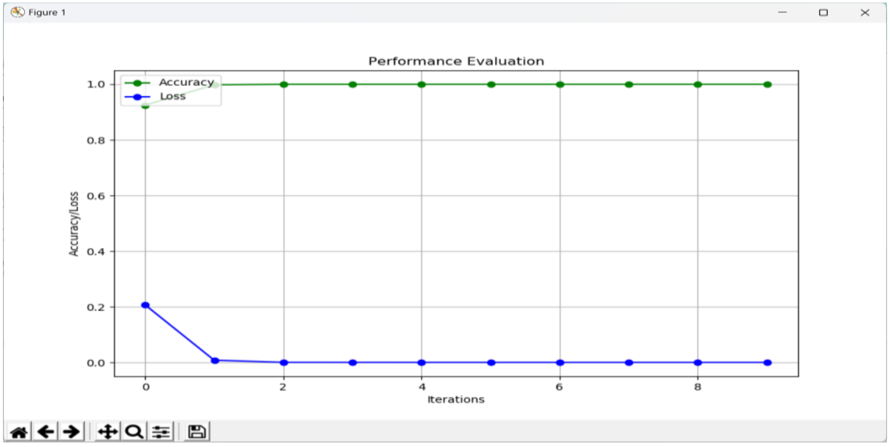


Figure 13: Performance evaluation graph of proposed deep learning approach.

5. CONCLUSION

This methodology not only showcases the potential for automated renal image analysis but also underscores the pivotal role that deep learning can play in expediting the diagnostic process. The achieved results exhibit a high level of accuracy and efficiency in identifying various kidney anomalies, thereby assisting healthcare professionals in making timely and informed decisions for patient care.

REFERENCES

[1] V. Jha, G. Garcia-Garcia, K. Iseki et al., “Chronic kidney disease: global dimension and perspectives,” The Lancet, vol. 382, no. 9888, pp. 260–272, 2013.

[2] R. Ruiz-Arenas, “A summary of worldwide national activities in chronic kidney disease (CKD) testing, the electronic journal of the international federation of,” Clinical Chemistry and Laboratory Medicine, vol. 28, no. 4, pp. 302–314, 2017.

[3] Thedailystar, “Over 35,000 develop kidney failure in Bangladesh every year,” 2019, https://www.thedailystar.net/city/news/18m-kidney-patients-bangladesh-every-year-1703665.

[4] Prothomalo, “Women more affected by kidney diseases,” 2018, https://en.prothomalo.com/bangladesh/Womenmore-affected-by-kidney-diseases.

[5] Scottish Intercollegiate Guidelines Network (Sign), Diagnosis and Management of Chronic Kidney Disease: A National Clinical Guideline, SIGN, Victoria, Australia, 2008.

[6] M. Kavitha, G. Gnaneswar, R. Dinesh, Y. R. Sai, and R. S. Suraj, “Heart disease prediction using hybrid machine learning model,” in Proceedings of the 2021 6th International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, January 2021.

[7] P. Ghosh, F. M. Javed Mehedi Shamrat, S. Shultana, S. Afrin, A. A. Anjum, and A. A. Khan, “Optimization of prediction method of chronic kidney disease using machine learning algorithm,” in Proceedings of the 2020 15th International Joint Symposium on Artificial Intelligence and Natural Language Processing (iSAI-NLP), Bangkok, Thailand, November 2020.

[8] K. R. A. Padmanaban and G. Parthiban, “Applying machine learning techniques for predicting the risk of chronic kidney disease,” Indian Journal of Science and Technology, vol. 9, no. 29, 2016.

[9] Charleonnan, T. Fufaung, T. Niyomwong, W. Chokchueypattanakit, S. Suwannawach, and N.

Ninchawee, “Predictive analytics for chronic kidney disease using machine learning techniques,” in Proceedings of the 2016 Management and Innovation Technology International Conference (MITicon), Bang-San, Thailand, October 2016.

[10] G.-S. Fu, Y. Levin-Schwartz, Q.-H. Lin, and D. Zhang, “Machine learning for medical \

imaging,” Journal of healthcare engineering, vol. 2019, pp. 1-2, 2019.

[11] R. Devika, S. V. Avilala, and V. Subramaniyaswamy, “Comparative study of classifier for

chronic kidney disease prediction using naive Bayes, KNN and random forest,” in Proceedings of the 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), Erode, India, March 2019

[12] S. Revathy, B. Bharathi, P. Jeyanthi, and M. Ramesh, “Chronic kidney disease prediction using

machine learning models,” International Journal of Engineering and Advanced Technology, vol. 9, no. 1, pp. 6364–6367, 2019.

[13] S. A. Rabby, R. Mamata, M. A. Laboni, Ohidujjaman, and S. Abujar, “Machine learning

applied to kidney disease prediction: comparison study,” in Proceedings of the 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, July 2019.

[14] M. Nishat, F. Faisal, R. Dip et al., “A comprehensive analysis on detecting chronic kidney

disease by employing machine learning algorithms,” EAI Endorsed Transactions on Pervasive Health and Technology, vol. 7, Article ID 170671, 2018.

[15] S. Pouriyeh, S. Vahid, G. Sannino, G. De Pietro, H. Arabnia, and J. Gutierrez, “A

comprehensive investigation and comparison of Machine Learning Techniques in the domain of heart disease,” in Proceedings of the 2017 IEEE Symposium on Computers and Communications (ISCC), Heraklion, Greece, July 2017.