Swanirman Sunirmit Publications of Research-Special Issue of ICRTTEAS July 2024 | [2024-25]
ISSN [Online]: 2583-2654

# AI Kidney Cancer Diagnosis: A Deep Learning Approach Integrating CT scan and Blood Test Analysis

*Mrs. Vinutha B A, Hemanth G, Nadeem Pasha, Narendra C, Nelson Leo N*
*Department of CSE, Dr.T. Thimmaiah Institute of Technology, KGF, Karnataka, India*

***Abstract :*** An improved deep learning technique called Kidney Cancer Diagnosis System (KCDS) automatically classifies kidney tumors using DenseNet and ANN models. The KCDS system addresses critical issues in kidney cancer detection through a dual approach involving images and blood test samples. These techniques enhance the accuracy and efficiency of early-stage diagnosis, crucial for preventing the disease from reaching a life-threatening stage. The proposed KCDS system for analyzing CT scan images involves four stages. First, the system receives kidney CT scan images. Next, it preprocesses the data by converting BGR to RGB, normalizing pixel values, resizing images using CV2, and splitting the data into training (80%) and testing (20%) sets. In the third stage, the system trains the data using DenseNet. Finally, it tests the model on the test data and analyzes the results to determine if a tumor is present. The blood test sample analysis consists of six stages. First, the system receives a .csv file of blood test samples. Next, it preprocesses the data by analyzing features and dropping null values. In the third stage, label encoding converts categorical variables into numerical form, and feature selection is performed. Outliers are detected and removed using the IQR method. The data is then split, and ANN trains the dataset to predict kidney cancer. The final stage involves testing and validating the model's accuracy. This proposed application integrates these models, providing a user-friendly dual interface for efficient kidney cancer diagnosis, potentially saving numerous lives through timely intervention.



***Keywords: DenseNet (CNN), Artificial Neural Network (ANN), kidney cancer, classification of kidney images, prediction using dataset values.***

## I. INTRODUCTION

Kidney cancer, primarily arising as renal cell carcinoma, is a significant health challenge. The kidneys, located at the back of the abdomen and protected by the rib cage, filter blood and play a crucial role in vitamin D metabolism. When kidney cells multiply rapidly, they form tumors, with advanced stages often leading to metastasis. Kidney cancer is the 16th most common cause of cancer death, with higher survival rates in developed countries due to early detection. Annually, over 400,000 new cases are diagnosed, highlighting the need for early and accurate diagnosis.

Early detection and treatment are crucial for improving patient outcomes, making kidney segmentation in CT scans vital for computer-assisted diagnosis and treatment planning. Accurate segmentation provides structural information on kidney irregularities, aiding specialists in analyzing conditions like carcinoma. Image processing and machine learning techniques, particularly deep learning (DL), have become instrumental in developing computational methods for medical image interpretation. DL, especially Convolutional Neural Networks (CNNs), has proven highly effective, often matching or surpassing human performance in image analysis.

ISSN [Online]: 2583-2654

This proposal leverages advanced deep learning models to enhance kidney cancer detection. By employing state-of-the- art CNNs (DenseNet) for imaging and Artificial Neural Networks (ANNs) for blood sample analysis, the research aims to create a robust dual-modality diagnosis system. The advanced DenseNet model, with increased convolution layers, offers detailed classification of kidney images, while ANNs complement image-based detection through blood sample analysis. This integrated approach bridges imaging and data-driven diagnostics, aiming for early and accurate detection.

The development of a user-friendly application housing the backend code for predictive analysis ensures practical application in real-world medical scenarios. This study aspires to transform kidney cancer diagnosis, facilitating early detection, improving prognosis, and enhancing the quality of life for patients.

## II RELATED WORK

In [3] the authors have used Kits19 dataset it contains 300 CT-Scan images. Also, they have gone through image data augmentation which is used for improving of model's learning rate. The data augmentation methodology used are: Selective Data Augmentation, Augmentation Multiplier, Generate Augmented Data, Append Original Images They have got the training accuracy between 90-99% which is a best accuracy but they have taken unbalanced dataset and they have gone through many techniques for classification of tumor images and kidney images. The parameters used to check the performance in this study are cross folding's, dice score, cross-entropy. The dataset is imbalanced data, So it required lot of preprocessing.

In [4] the authors have gone through smaller image dataset with 1035 CT images from 308 patients. And in this study, they have using 3D images of kidney and classifying them into different types kidney cancer. In this study they are using CNN with less number of filters and not.

In [5] the authors have gone through all types of kidney classification through CNN and machine learning. In this study they have used a less number of convolution layers in CNN. And they have gone through all machine learning techniques like logistic regression (LR), decision tree (DT), LSTM, SVM such as GA G-SVM and SNAP-SVM , K- Nearest Neighbors Algorithm (KNN) and many other models, but they have not gone through Artificial Neural Network (ANN) which is very useful for improve the accuracy of the model.

In [6] the authors have gone through techniques like Histopathology and Proteomics which consist of visual analysis tools to identify the cellular patterns in kidney. But, in this study they have used Random Forest and VGG16- CNN. This study consists of less image datasets and it is also imbalanced data; the data is generated by CPTAC form the clear cell renal cell patients. Also, they have used Proteomics data encode molecular for diagnosis of cancer.

In [7] the authors have gone through kidney tumor segmentation using deep CNN with the KiTs19 dataset the e xtension they have used for deep CNN model are AlexNet, and U-Net 2D. They were evaluated with 210 CTs scan images and got 96% of accuracy which is good but they have two type extensions for the deep CNN model which is draw back in this study. The dataset used is consist of less images and this study is used for segmentation of images and there are no separate images like tumor and nontumor kidney images and it will not accurately gives whether it is tumor image or not.

In [8] the authors have considered the Renal cell carcinoma tissue to identify the kidney tumor. This

ISSN [Online]: 2583-2654

tissue is useful for identification kidney cancer which is happen when there is growth in the tissue. In this study, they have gone through deep CNN model, the dataset contains of 3D CNN images and data they have collected of 160 patients of TCIA database, and they have got 98% accuracy which is best but they have fixed with growth of the t issue to identify the kidney tumor is there are not.

In [9] the authors have proposed deep Convolutional Neural Network (CNN) model, known as the 3D-MS- RFCNN, is designed specifically for kidney tumor segmentation. It consists of two encoders and one decoder. The encoders are meticulously designed to capture features of images with low resolution by down-sampling the input image data 【9,10】. This approach enhances the model ' s ability to process and analyze the intricate details of kidney tumor images. Utilizing the KiTS dataset, the 3D-MS-RFCNN model has achieved an impressive 93% accuracy in segmenting kidney tumor images, demonstrating its high effectiveness and potential for significant impact in clinical applications and diagnostic processes.

## III PROPOSED SYSTEM

The proposed work we are going for the dual mode diagnosis frame work by integrating the DenseNet model and the ANN models into a unified dual mode diagnosis framework. We use DenseNet architecture of the CNN model to enhance classification accuracy of kidney images, distinguishing between Tumor and Normal tissues with greater precision. We implement and integrate ANN models for the analysis of blood sample dataset values. This will provide an additional modality for kidney cancer prediction, contributing to a more holistic diagnostic system. Design and implement a user-friendly application interface that incorporates the developed DenseNet and ANN models. In this application will serve as a practical tool for medical practitioners and users, providing easy access to diagnostic results and contributing to the seamless integration of the proposed models into real-world healthcare settings.

### *A. DenseNet Methodology*

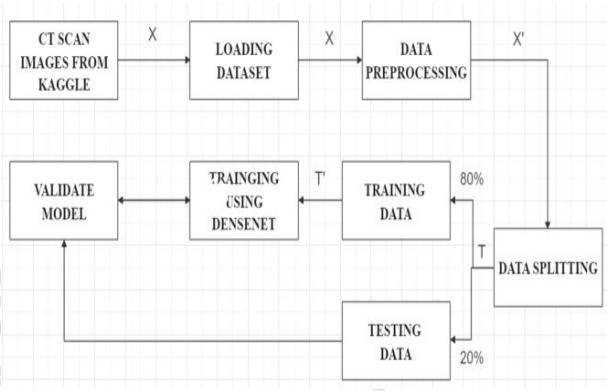


**Fig 1**

***Image Dataset:***

Collection of data containing the CT-scan images of the Kidney through the Kaggle. Our dataset contains 5077 normal and 2283 tumor class structured CT scan images gathered from Kaggle. Making it a comprehensive and structured resource for training and testing our model.

After collecting the required data sets we will load the data into our system.

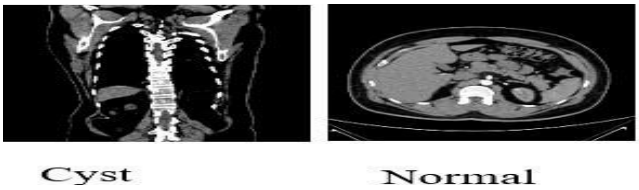


**Fig 2**

ISSN [Online]: 2583-2654

***Data processing stage:***

The pre-processing module in CT-scan images involves a series of steps to enhance the quality of the images, standardize their format, size and prepare them for the analysis.

The steps involved in pre-processing includes following techniques

### *1. BGR to RGB scale conversion*

The first step in pre-processing is converting the images from BGR to RGB scale. Converting images from BGR to RGB is required in deep learning to ensure that the images are in the correct format expected by the deep learning framework. If the images are not converted, the model may not perform as expected, as it would receive inputs in the incorrect color channel order. Therefore, converting images from BGR to RGB is a necessary preprocessing step when working with deep learning frameworks to ensure that the images are in the expected format for training or inference. Converting an image from BGR (Blue, Green, Red) to RGB (Red, Green, Blue) format involves rearranging the order of color channels. In Python, you can use OpenCV for this conversion. Converting them to RGB scale simplifies the data representation.

***Mathematically, this can be represented as:***

*(R,G,B) RGB = (B,G,R)BGR.....(1)*

So, if you have a pixel in BGR format like (100, 150, 200), its equivalent in RGB format wouldbe (200, 150, 100).

### *2. Resizing the images*

Resizing of images is done to ensure that all the CT scan images have the same dimensionality for standardized analysis and adopting the images to meet the requirement of the deep learning model which will improve the efficiency of the model. This can be done by computer vision libraries in python. X is the input image and X should be resized to a new size Xresized, the resizing process typically involves interpolation to determine pixel values in the resized image. X should be resized to a new width while maintaining the aspect ratio preservation. The aspect ratio is typically defined as the ratio of the width to the height of an image. The original width is denoted as W0 and the original height as H0. The aspect ratio (AR0) is given by:

*AR = W0 / H0 ..... (2)*

To preserve the aspect ratio when resizing, new width is denoted as Wn and to find the corresponding height Hn the mathematical formula is given by

*AR = W0 / H0 = Wn / Hn... ..... (3)*

***The above equation can be rearranged as***

*Hn or Xresized = Wn / AR..... (4)*

### *3. Normalization*

Normalization is done to improve the Visual interpretation of the CT scan images by adjusting the pixel values to a more intuitive scale. It will also mitigate the impact of noise or variation in Pixel values.

### *4. Min-Max Scaling*

Min-max scaling, also known as min-max normalization, is a common technique used to normalize data, including images. The purpose of normalization is to bring the values of different features into a similar scale, making it easier for machine learning algorithms to learn and converge efficiently.

We have used Min-Max scaling for the normalization where it will transform the pixel

ISSN [Online]: 2583-2654

values of an image to a specific range typically [0,1]. For images, min- max scaling is applied to each pixel value. Here's a brief overview:

The min-max scaling formula for a single pixel value X is as follows

$$Xnormalized = \frac{X-min}{max(X)-min(X)} \quad \ldots\ldots(5)$$

• X normalized: The normalized pixel value
• X: Original pixel Value
• min(X): The minimum pixel value in the image
• max(X): The maximum pixel value in the image

### *Process*

• Identify the minimum (min) and maximum (max) pixel values in the entire image.
• Choose the desired new range, often [0, 1] for image normalization.
• Apply the min-max scaling formula to each pixel in the image.

### *Data Splitting*

Data splitting is the act of partitioning available data into two portions usually for validation purpose. One portion of the data is used to develop predictive model and the otherto evaluate the models performance. Typically, when we separate a data set into training and testing set, most of the data(80%) is used for training and a smaller portion(20%) the data is testing.

### *Data Training (Classification)*

We use advanced convolution neural network model i.e. DenseNet architecture to train the images by classification. The goal of the DenseNet architecture model is to classify the images based on the shape and color features, facilitating more efficient and accurate analysis. DenseNet, short for Dense Convolutional Network, is a type of convolutional neural network (CNN) known for its densely connected layers. Unlike traditional CNN architectures, where each layer is connected only to the next layer in a feedforward manner, DenseNet connects each layer to every other layer in a feedforward fashion.

### *Test Dataset*

The test dataset is another subset of original data, which is independent of the training dataset. It's time to fit the first machine learning model into your data once you've cleaned it up, visualized it, and learnt more about it.

### *Classification result*

With the classified dataset (training dataset) the test data can be predicted for kidney cancer. And the corresponding positive and negative predictions with their probabilities are obtained. To generate prediction of kidney cancer, algorithms been developed and their accuracy was tested. After attaining results from various types of supervised learning and the output will be classification of kidney cancer based on the patients input.

### *DENSENET algorithm*

DenseNet, or Dense Convolutional Network, is a deep learning architecture renowned for its dense interconnections between layers, contrasting with traditional convolutional neural networks (CNNs) where each layer is connected only to the subsequent one. This dense connectivity facilitates feature reuse, reduces the number of parameters, and enhances gradient flow, making training more efficient. By directly connecting each layer to every other layer in a feed-forward fashion, DenseNet encourages the network to learn compact and discriminative representations. This architecture has proven effective in addressing the vanishing gradient to the problem, enabling the training of very deep networks. DenseNet has achieved state-

ISSN [Online]: 2583-2654

of-the-art performance in image classification and transfer learning tasks, making it a popular choice in image analysis applications.

***Step of the DenseNet algorithm***

***1. Input Data Preparation***

Preprocess the input images by resizing them to a fixed size, normalizing pixel values (typicallyto the range [0, 1] or [-1, 1]), and applying data augmentation techniques such as random rotation, flipping, and cropping to increase the diversity of training samples

***2.Initial Convolutional Layer***

Apply a convolutional layer with a certain number of filters (channels) to extract basic featuresfrom the input images. Optionally, followed by batch normalization and ReLU activation to normalizeand introduce non-linearity to the feature maps.

$$(I * W)(i, j) = \sum m \sum n\, I(I + m, j + n).\, W(m, n) \quad \ldots\ldots (6)$$

***3. Dense Block Formation***

Start with an initial input feature map. Concatenate feature maps from all preceding layers within the dense block. Pass the concatenated feature maps through a series of convolutional layers (typically 3x3 filters). Optionally, employ bottleneck layers (1x1 convolutions) to reduce the number of channels and computational cost. Use batch normalization and ReLU activation after each convolutional layer.

***4. Transition Layer***

After each dense block, introduce a transition layer to down sample feature maps and reduce their spatial dimensions. Typically consists of a 1x1 convolutional layer followed by average pooling to halve the spatial dimensions. Batch normalization and ReLU activation may also be applied.

***5. Repeat Dense Blocks and Transition Layers***

Stack multiple dense blocks and transition layers to increase the depth of the network. The number of dense blocks and transition layers can vary based on the desired model architecture and complexity.

***6. Pooling (Max Pooling)***

Use pooling layers to down sample spatial dimensions and reduce computational complexity

***Mathematically is given by:***

$$MaxPooling(i,j)=max(ActivationMap(2i,2j),ActivationMap(2i,2j+1),ActivationMap(2i+1,2j),ActivationMap(2i+1,2j+1)) \quad (7)$$

***7. Fully Connected Layer***

Add a fully connected layer to map the features obtained from global average pooling to the output classes. Optionally, apply dropout regularization to prevent overfitting. Finally, apply a softmax activation function to obtain class probabilities for multi-class classification tasks.

***8. Training***

Use a training dataset with labeled samples to train the DenseNet model. Minimize a loss function (e.g., categorical cross-entropy) using an optimization algorithm (e.g., stochastic gradient descent, Adam) to update the model parameters. Monitor training metrics (e.g., loss, accuracy) on a separate validation set to avoid overfitting.

***9. Evaluation***

Assess the performance of the trained model on a held-out test set to evaluate its generalizationability. Compute evaluation metrics such as accuracy, precision, recall, and F1- score to quantify the model's performance.

***10. 1nference***

Deploy the trained DenseNet model to make predictions on new, unseen data. Pass input images

ISSN [Online]: 2583-2654

through the network, and use the output probabilities to make predictions about the corresponding classes.

This detailed step-by-step process illustrates the complete pipeline of training, evaluation, and inference using the DenseNet algorithm for image classification tasks. Each step contributes to building a robust and effective deep learning model for real-world applications.

### *B. ANN Methodology*

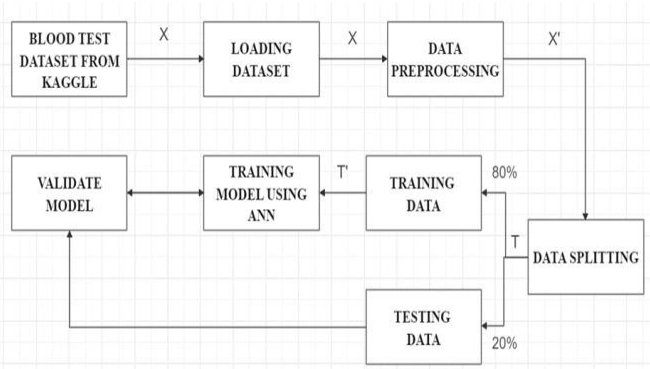


**Fig 3**

***Data Collection and loading***

Collection of the data containing the blood test samples of the patients having kidney cancer and normal through Kaggle. Our data set contains structured data of 400 records of patients with 26 attributes, which contains WBC count, RBC cou(n8t), Platelets count and many other symptoms or medical factors. After collecting the required data set, we will load the data into our system.

***Data Preprocessing***

The Data preprocessing in terms of CSV file (Structured data) involves a series of steps to remove redundancy and the null values, selection of required features and handling the outliers.

***The steps involved in preprocessing includes following techniques:***

• Handling of Null values in the dataset by dropping them as there are limited numbers of Null values.
• Label encoding will be done to covert the categorical variables into numerical form. This will be done manually
• Feature Selection: Required features or the attributes will be selected which are important for the analysis and the remaining will be dropped.
• Handling outlier by IQR: The Inter Quartile Range (IQR) method is a technique used for identifying and handling outliers in the dataset, There outliers may result in some errors and decrease in the Model performance. Here are the steps involved:-

#### *1. Calculate the IQR*

IQR is the range between first quartile (Q1) and the third quartile (Q3), It is calculated as follows

$$IQR = Q3 - Q1 \quad \text{.....(8)}$$

#### *2. Identity Outlier*

Values that fell below Q1 – 1.5 X IQR or above Q3 +1.5 X IQR are considered as outlier

#### *3. Handling Outlier*

Removing or excluding the outlier points from the data set in our case.

***Data Splitting***

Data splitting is the act of partitioning available data into two portions usually for validation purpose, one portion of the data is used to develop predictive performance model and the other to evaluate the model performance. Typically when we separate a data set into training and testing at most of the data(80%) is used for training and a smaller portion(20%) the data is used for testing.

ISSN [Online]: 2583-2654

***Model Training***

Here for the purpose of training the model we have used Artificial Neural Network. Which is a new approach for analyzing the kidney cancer detection with the help of blood test samples, First 80% of the numerical data is used for ANN model or for the purpose of Training which will result in the classisification whether the patient is affected or not. ANN are computational models inspired by the structure and functioning of the human brain. They consist of interconnected nodes or "neurons" organized in layers. The input layer receives data, which is then processed through hidden layers, and finally, an output layer produces the result. Each connection between neurons has an associated weight, which adjusts during training to optimize predictions. ANNs excel at learning complex patterns in data, making them widely used in tasks like image recognition, natural language processing, and regression analysis. They possess the ability to generalize from training data to make predictions on unseen data. However, ANNs can be computationally intensive and require substantial data for effective training. Despite their complexity, they are a cornerstone of modern machine learning and have led to significant advancements in various fields.

An artificial neural network is a supervised learning algorithm which means that we provide it the input data containing the independent variables and the output data that contains the dependent variable. In the beginning, the ANN makes some random predictions, these predictions are compared with the correct output and the error(the difference between the predicted values and the actual values) is calculated. The function that finds the difference between the actual value and the propagated values is called the cost function. The cost here refers to the error. Our objective is to minimize the cost function. Training a neural network basically refers to minimizing the cost function. We will see how we can perform this task.A neural network executes in two phases: Feed Forward phase and Back Propagation phase.

***Let us discuss both these steps in detail.***

In the feed-forward phase of ANN, predictions are made based on the values in the input nodes and the weights. If you look at the neural network in the above figure, you will see that we have three features in the dataset: X1, X2, and X3, therefore we have three nodes in the first layer, also known as the input layer. The weights of a neural network are basically the strings that we have to adjust in order to be able to correctly predict our output.

For now, just remember that for each input feature, we have one weight.

***The following are the steps that execute during the feedforward phase of ANN:***

***Step 1: Calculate the dot product between inputs and weights***

The nodes in the input layer are connected with the output layer via three weight parameters. In the output layer, the values in the input nodes are multiplied with their corresponding weights and are added together. Finally, the bias term b is added to the sum. Suppose if we have a person who has input values (0,0,0), the sum of the products of input nodes and weights will be zero. In that case, the output will always be zero no matter how much we train the algorithms. Therefore, in order to be able to make predictions, even if we do not have any non-zero information about the person, we need a bias term. The bias term is necessary to make a robust neural network.

Mathematically, the summation of dot product:
$X.W = x1.w1 + x2.w2 + x3.w3 + b$ ..... (9)

ISSN [Online]: 2583-2654

***Step 2: Pass the summation of dot products (X.W) through an activation function***

The dot product XW can produce any set of values. However, in our output, we have the values in the form of 1 and 0. We want our output to be in the same format. To do so we need an Activation Function, which restricts the input values between 0 and 1. So of course we'll go ahead with Sigmoid activation function. The sigmoid function returns 0.5 when the input is 0. It returns a value close to 1 if the input is a large positive number. In the case of negative input, the sigmoid function outputs a value close to zero.

Therefore, it is especially used for models where we have to predict the probability as an output. Since probability of anything exists only between the range of 0 and 1, sigmoid is the right choice for our problem.

In the above figure z is the summation of dot product
X.W Mathematically, the sigmoid activation function is:

$$f(x) = \frac{1}{1+e-(x)} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (10)$$

First, we have to find the dot product of the input features (matrix of independent variables) with the weights. Next, pass the summation of dot products through an activation function. The result of the activation function is basically the predicted output for the input features.

***Back Propagation:***

In the beginning, before you do any training, the neural network makes random predictions. We then compare the predicted output of the neural network with the actual output. Next, we update the weights and the bias in such a manner that our predicted output comes closer to the actual output. In this phase, we train our algorithm. Let's take a look at the steps involved in the backpropagation phase.

***Step 1: Calculate the cost***

The first step in this phase is to find the cost of the predictions. The cost of the prediction can be calculated by finding the difference between th e predicted output values and the actual output values. If the difference is large then cost will also be large.

We will use the mean squared error or MSE cost function. A cost function is a function that finds the cost of the given output predictions.

***Step 2: Minimize the cost***

Our ultimate goal is to fine-tune the weights of our neural network in such a way that the cost is minimized the minimum. If you notice carefully, you'll get to know that we can only control the weights and the bias. Everything else is beyond our control. We cannot control the inputs, we cannot control the dot products, and we cannot manipulate the sigmoid function.

In order to minimize the cost, we need to find the weight and bias values for which the cost function returns the smallest value possible. The smaller the cost, the more correct our predictions are.

To find the minima of a function, we can use the gradient descent algorithm. The gradient descent can be mathematically represented as:

∂Error is the cost function. The above equation tells us to find the partial derivative of the cost function with respect to each weight and bias and subtract

ISSN [Online]: 2583-2654

the result from the existing weights to get new weights.

In the above equation a is called the learning rate, which is multiplied by the derivative. The learning rate decides how fast our algorithm learns. We need to repeat the execution of gradient descent for all the weights and biases until the cost is minimized and for which the cost function returns a value close to zero.

## IV. RESULT AND ANALYSIS

The proposed deep learning models ANN and CNN model, when examined the two models then training accuracy is 99% and 96%, Shown in Fig. 4 & 5 We have got improved accuracy compared of [3] and [4]. Red colour.

The validation accuracy of the two models is also improved when compared to [3] and [4] they are 97% and 99% , Shown in Fig. 4 & 5. The table I will gives accuracy values of models.

The training loss and validation loss of proposed ANN model is shown in Fig.6. validation loss= 0.0163, training loss= 0.037. Red colour refers to training and blue colour refers to validation.

**TABLE I ACCURACY TABLE**

| S.No | Accuracy of proposed models. | | |
|---|---|---|---|
| | *Models* | *Accuracy of training* | *Accuracy of validation* |
| 1 | CNN | 96.4% | 99.6% |
| 2 | ANN | 99% | 97% |

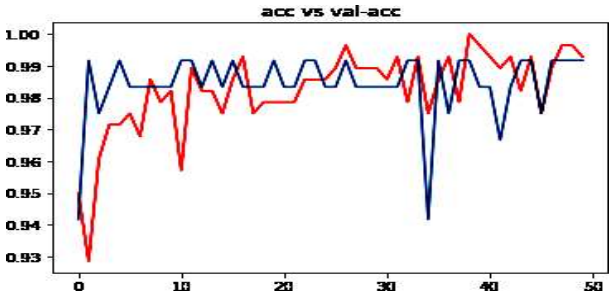


**Fig.4. Training and Validation Accuracy of Proposed ANN model**

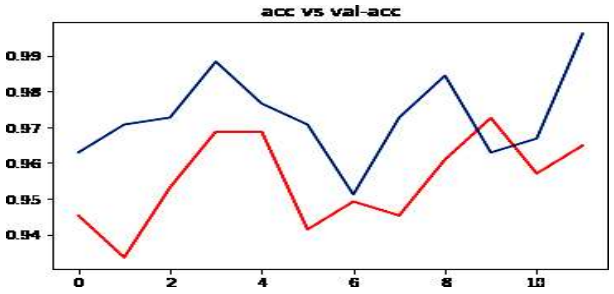


**Fig.5. Training and validation accuracy of Proposed CNN model**

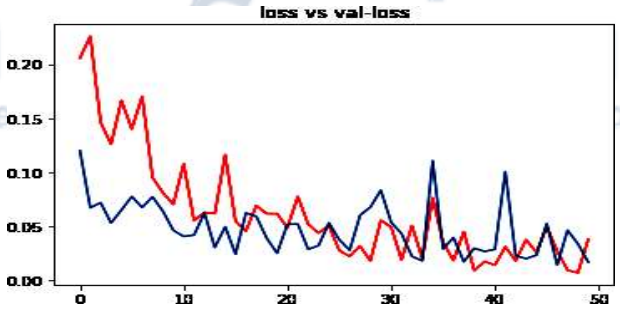


**Fig.6. Training loss and Validation loss of Proposed ANN model.**

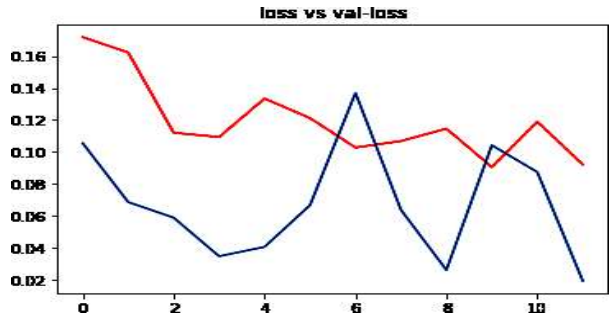


**Fig.7. Training loss and Validation Loss of CNN model**

ISSN [Online]: 2583-2654

Fig.8. illustrates the Confusion matrix for training data of the Proposed ANN model. A confusion matrix is used to show how the classification of model works. A confusion matrix will summarizes the results of the proposed classification model. From the confusion matrix we can say the proposed model detects the kidney cancer accurately. Fig.9. illustrates the Confusion matrix for validation data of the Proposed ANN model.

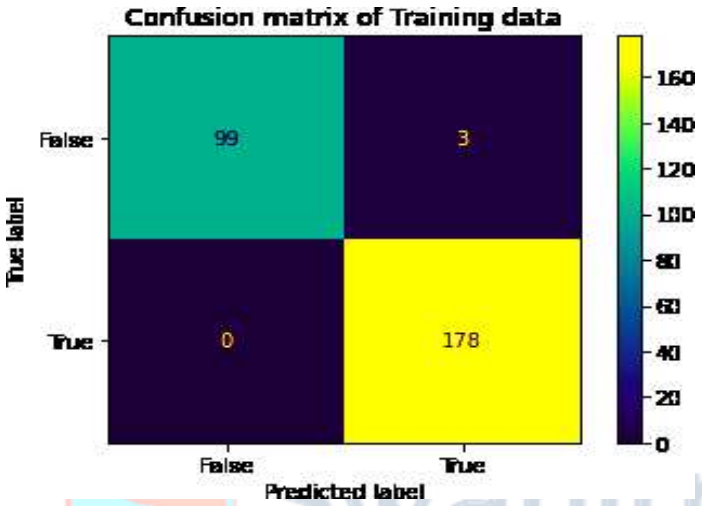


**Fig.8. Confusion matrix of Proposed ANN model.**

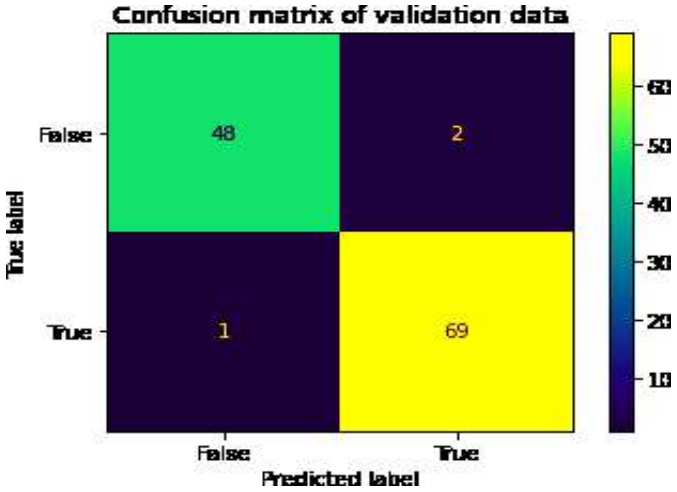


**Fig.9. Confusion matrix of Proposed ANN model.**

The Accuracy of the proposed models are better than [3] & [4] which is achieved by changing the architecture of the models that is by adding more convolution layer to the CNN, and by adding more hidden layer to ANN model, and increasing neurons in the architecture, by using correct preprocesssing of dataset, and by using large dataset, and by increasing more number epoches for training the model.

The performance of the proposed models is also better then [3] and [4]. Performance of CNN model is improved by hyper parameter tuning such as patience, verbose, delta(learning rate). In proposed model patience = 8, verbose= 1, delta = 0.01. Performance of ANN model is improved by using 'Relu' activation function in the hidden layers , and by using 'adam' optimizer for training the model.

***Web App – build & deployed on pycharm cloud***

Using trained CNN model, a Flask App was constructed that takes input from the website and gives them to the trained model. Then, the model predicts the output which is again rendered to the website where it is displayed to the user. This Flask app is deployed over a cloud platform - IDE - PyCharm Community Edition.

## V CONCLUSION

The research shows the proposed model for kidney cancer detection. At first, it shows the implementation of the overall proposed model and its layers. Proposed research "Kidney cancer detection using deep learning models" will help in predicting the kidney cancer disease. In this research, the training models like ANN, CNN are included. In the proposed research classification of kidney cancer is performed through images and test sample value, so if the user enters kidney cancer test sample values as input the model will predict the kidney cancer is there or not. If the user enters CT scan image of kidney in the web page, then CNN model will detect

the cancer is there are not. ANN model is trained and saved for further use. The training and validation accuracy for the both the models are shown in the Table I. Proposed research also helps in the situation when the patient is trying to know the kidney cancer is there or not with the kidney cancer test samples available or with the Kidney CT scan images.

***REFERENCES***

*[1] Matta Preethi Reddy, MD Farhan Mohiuddin, Shruthi Budde, Gajula Jayanth, Ch Rajendra Prasad, Srikanth Yalabaka. "A Deep Learning Model for Traffic Sign Detection and Recognition using Convolution Neural Network", Published: 24 June 2022.*

*[2] Kollem, S., Prasad, C. R., Ajayan, J., Malathy, V., & Subbarao, A. (2022). " Brain tumor MRI image segmentation using an optimized multi-kernel FCM method with a pre-processing stage. Multimedia Tools and Applications", 1-30.*

*[3] Akram Z*, Kareem MS, Mughal B, Ahmed Z, and Aziz S. " Cancerous Tumor Segmentation of Kidney Images and Prediction of T umor Using Medical Image Segmentation and Deep Learning Techniques", Published : 09 Mar 2021.*

*[4] Kwang-Hyun Uhm, Seung-Won Jung, Moon Hyung Choi, Hong-Kyu Shin, Jae-Ik Yoo, Se Won Oh, Jee Young Kim, Hyun Gi Kim, Young Joon Lee, Seo Yeon Youn, Sung-Hoo Hong, Sung-Jea Ko. " Deep learning for end-to-end kidney cancer diagnosis on multi-phase abdominal computed tomography", Published: 18 June 2021.*

*[5] Maha gharaibeh, Dalia Alz'bi, Malak Abdullah, Ismail Hmeidi, Mohammad Rustom AI Nasar, "Radiology Imaging Scans for Early Diagnosis of Kidney Tumors: A Review of Data Analytics-Based Machine Learning and Deep Learning Approaches", Doi: 10.3390/bdcc6010029, Published: 8 March 2022.*

*[6] Kollem, Sreedhar, et al. "Image denoising for magnetic resonance imaging medical images using improved generalized cross - validation based on the diffusivity function." International Journal of Imaging Systems and Technology 32.4 (2022): 1263-1285.*

*[7] Ehwa Yang, Chan Kyo Kim, Yi Guan, Bang-Bon Koo, Jae-Hun Kim, "3D multi-scale residual fully convolutional neural network for segmentation of extremelylarge-sized25kidneytumor", https://doi.org/10.1016/j.cmpb.2022.106616Version of Record 10 January 2022.*

*[8] Luana Batista da Cruz, José Denes Lima Araújo, Jonnison Lima Ferreira, Joao Otavio Bandeira Diniz, Aristofanes Correa Silva, João Dallyson Sousa de Almeida, Anselmo Cardoso de Paiva, Marcelo Gattass, "Kidney segmentation from computed tomography images using deep neural network",Doi: 10.1016/j.compbiomed.2020.10390621, published: 6 July2020*

*[9] Seokmin Han, Sung Il Hwang, Hak Jong Lee, "The classification of renal cancer in 3- phase CT images using a deep learning method", Published: 16 May 2019.*

*[10] Duggirala SR, Kollem S, Rama Linga Reddy K (2021) An optimized SVM-based possibilistic fuzzy c-means clustering algorithm for tumor segmentation. Multimed Tools Appl 80(1):409–437*

*[11] Fuzhe Ma a , Tao Sun a , Lingyun Liu b,∗ , Hongyu Jing c, "Detection and diagnosis of chronic kidney disease using deep learning-based heterogeneous modified artificial neural network", Published: 24 April 2020.*

*[12] Suresh, G., Prasad, C. R., & Kollem, S. (2022, May). Telugu Optical Character Recognition Using Deep Learning. In 2022 3rd International Conference for Emerging Technology (INCET) (pp. 1 - 6). IEEE.*

*[13] Reddy, Matta Preethi, et al. "A Deep Learning Model for Traffic Sign Detection and Recognition using Convolution Neural Network." 2022 2nd International Conference on Intelligent Technologies (CONIT). IEEE, 2022.*