VIT
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology & Engineering
Department of Computer Application
Fall Semester 2022 - 2023

SET CONFERENCE

First Review (7th December 2022)

Reg no: 22MCA0184, 22MCA0187
Student Name: RITESH KUMAR GUPTA, ABHISHEK DAS

Guide Name: Assistant Prof. YUVARANI S
Paper Name: Kidney Diseases Detection and Classification using CT Scanned Image.

Problem Description:

In the modern era, human life span is decreasing due to multiple causes. Among all the causes, eating unhealthy processed and packaged food is one of the major reasons. With this bad lifestyle a person gets numerous diseases. Kidney diseases such as Stone, Cyst, Tumor, etc are some of the many diseases that a person can be diagnosed with. The diseases related to the kidney are the most painful ones. Oftentimes, these are diagnosed very late which leads to further complications. Furthermore, in rural areas such expert doctors are not available who can determine whether a person is suffering from Kidney Stone, Cyst or Tumor. In most of the cases such patients die without any treatment as these are not detected at an early stage. This project looks to solve this problem by using Deep Learning concepts and image classification. For this purpose we have taken a dataset from popular platform Kaggle. We take Computed

Tomography (CT) scanned images and analyze the image balancing. If the data is imbalanced then we take the help of the SMOTE algorithm. It helps to even - out the data. After that we push it to (Convolutional Neural Network) CNN algorithm. It shortens the image matrix and flattens it, converting it into a neural network. After that it classifies the given image as Kidney Stone, Cyst and Tumor. This research work can be easily integrated with Hospital Diagnosis Departement allowing for better patient care and diagnosis.

Literature Survey

[1] explored methods for the detection of chronic Kidney disease. Adaptive Hybridized Deep Convolutional Neural Network (AHDCNN) was used to categorize different CKD. Different features are determined from the noise free data and fed into the classifier. Proposed approach is based on the method for deeper learning and ROIs given by radiologists.

[2] explored Lesion detection using CT images. Morphological cascade convolutional neural network (RCNN) was employed which gives highly precise and robust detection of kidney lesions. A six layer FPN is modified to generate different size feature maps for overall location and classification. However their model has shown low precision when there are several complicated goals such as polycystic kidney.

[3] worked on recognizing kidney stones from CT scan pictures. Firstly enhancement of input image is done by histogram equalization approach and for better visibility, Embossing technique is employed. Lastly, the convolution kernel and SVM image is classified. Obtained 98.71% accuracy on the basis of TP, TN, FP, FN parameters.

[4] proposed a smartphone based implementation(in the form of application(APP)) of kidney abnormalities(stones and cysts) detection using ultrasound images. For the detection of kidney abnormality Viola Jones

algorithm, texture feature extraction and SVM classifier is used. The result of the proposed algorithm gives the accuracy of 90.91% in detection of the abnormalities.

[5] introduced the classification of the Kidney disease using Ultrasound(US) image, namely into four classes : Normal, Cyst, Calculi and Tumor. Model used is Probabilistic Neural Network(PNN). Equal number of data is used for training and testing of the data which results in the accuracy of 92.99%, Sensitivity of 88.04% and Specificity of 97.33% is obtained. Drawback of the model is that an equal amount of data is used for training and testing, which can be improved by increasing the number of samples for training work.

[6] explored the classification of the Kidney tumor and pining out the tumor regions by processing the Computer Tomography(CT) scan images. They have used K-means segmentation and K-means clustering technique which gives the accuracy of 91.2 and dice coefficient of 87.7. Further improvement can be done by using new segmentation techniques.

Proposed work:

In this proposed research we have presented a deep learning model that classifies the type of Kidney disease using Computed Tomography (CT) scanned images. It takes CT scanned images as input, balances out the imbalanced data with the help of SMOTE algorithm and applies some deep learning algorithm to give an output. In addition, it classifies the image into kidney Stones, Cyst, Tumor or normal. Since there are a lot of rural areas where specialized doctors are not available, the patients are not able to diagnose which type of kidney disease they are suffering from. Consequently, most of the patients suffer a lot of pain and at the worst they die. To tackle this problem we need to implement some algorithm which will classify the image and print the output with minimum time and highest accuracy at the lowest cost possible.

Architecture Diagram

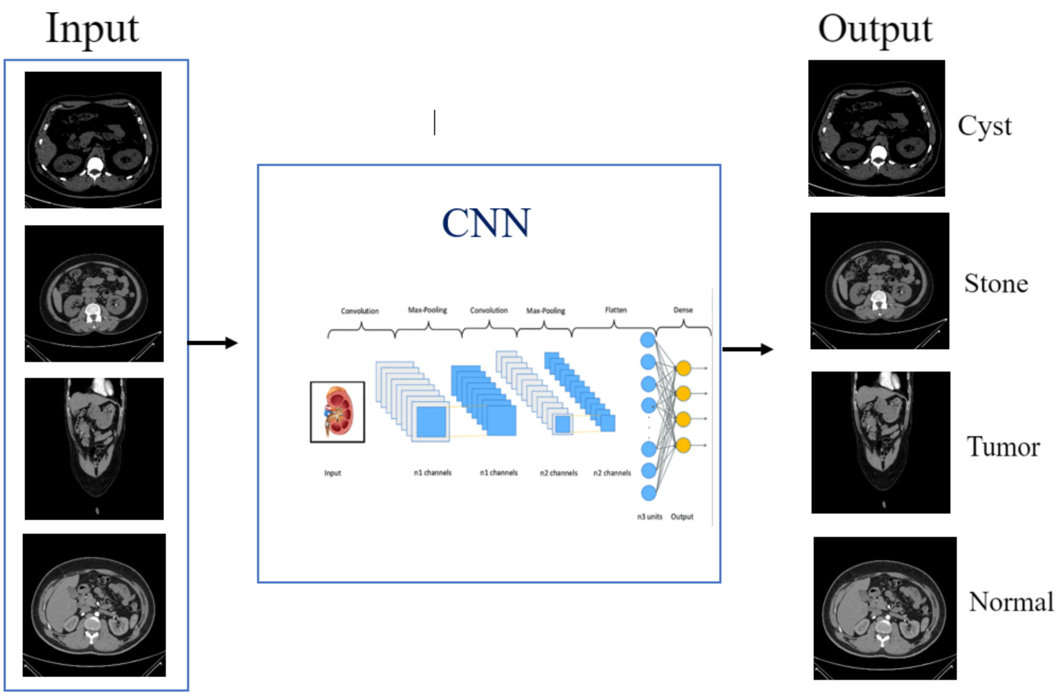

Fig 1: Architecture Design

Methodology and Module Description

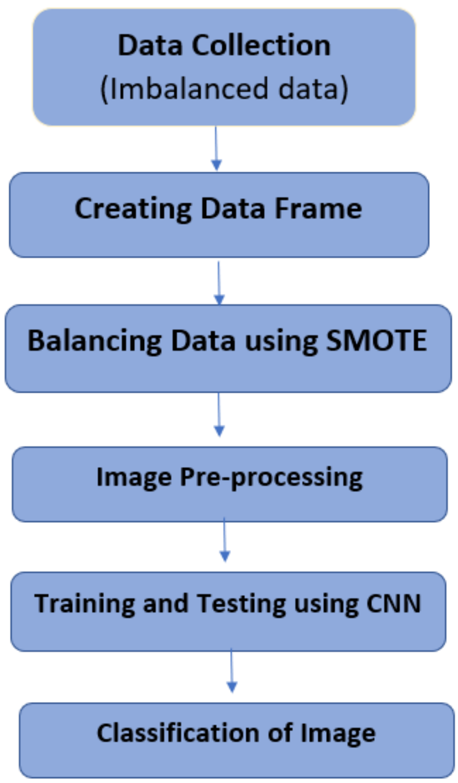

Fig 2: Methodology

Implemented Work (50%):

Working Code :-

1. **Accessing Dataset Path:**

```
In [1]: import numpy as np
        import pandas as pd
        from pathlib import Path
        import matplotlib.pyplot as plt
        import seaborn as sns
        from skimage.io import imread
        import cv2
```

Getting image Path

```
In [2]: # initializing path
        data_dir = Path('CT-KIDNEY-DATASET-Normal-Cyst-Tumor-Stone')
        train_dir = data_dir
```

```
In [3]: # Get the path to the normal and pneumonia sub-directories
        normal_cases_dir = train_dir / 'Normal'
        cyst_cases_dir = train_dir / 'Cyst'
        stone_cases_dir = train_dir / 'Stone'
        tumor_cases_dir = train_dir / 'Tumor'
```

```
In [4]: # Getting the list of all the images
        normal_cases = normal_cases_dir.glob('*.jpg')
        Cyst_cases = cyst_cases_dir.glob('*.jpg')
        Stone_cases = stone_cases_dir.glob('*.jpg')
        Tumor_cases = tumor_cases_dir.glob('*.jpg')
```

```
In [5]: # An empty list for inserting data into this list in (image_path, Label) format
        train_data = []
```

```
In [6]: # Labeling the Cyst case as 0
        for img in Cyst_cases:
            train_data.append((img, 0))

        # Labeling the Normal case as 1
        for img in normal_cases:
            train_data.append((img, 1))

        # Labeling the Stone case as 2
        for img in Stone_cases:
            train_data.append((img, 2))

        # Labeling the Tumor case as 3
        for img in Tumor_cases:
            train_data.append((img, 3))
```

```
In [7]: # Making a data frame using pandas (creating CSV file)
        train_data = pd.DataFrame(train_data, columns=['image', 'label'], index=None)
```

Fig 3: Accessing Dataset

Creating Data Frames:

DataFrame is a two-dimensional array with labeled axes (rows and columns). The term "DataFrame" refers to a common method of storing data that has both a row index and a column index.

Creating DataFrame ¶

In [5]:
```
# An empty list for inserting data into this list in (image_path, Label) format
train_data = []
```

In [6]:
```
# Labeling the Cyst case as 0
for img in Cyst_cases:
    train_data.append((img, 0))

# Labeling the Normal case as 1
for img in normal_cases:
    train_data.append((img, 1))

# Labeling the Stone case as 2
for img in Stone_cases:
    train_data.append((img, 2))

# Labeling the Tumor case as 3
for img in Tumor_cases:
    train_data.append((img, 3))
```

In [7]:
```
# Making a data frame using pandas (creating CSV file)
train_data = pd.DataFrame(train_data, columns=['image', 'label'], index=None)
```

In [8]:
```
# Select random data row from the dataframe and show and index it.
train_data = train_data.sample(frac=1.).reset_index(drop=True)
train_data.head()
```

Out[8]:

	image	label
0	CT-KIDNEY-DATASET-Normal-Cyst-Tumor-Stone\Norm...	1
1	CT-KIDNEY-DATASET-Normal-Cyst-Tumor-Stone\Norm...	1
2	CT-KIDNEY-DATASET-Normal-Cyst-Tumor-Stone\Norm...	1
3	CT-KIDNEY-DATASET-Normal-Cyst-Tumor-Stone\Norm...	1
4	CT-KIDNEY-DATASET-Normal-Cyst-Tumor-Stone\Norm...	1

In [9]:
```
# Returns all the unique values of Label in train_data
train_data['label'].unique()
```

Out[9]:
```
array([1, 2, 0, 3], dtype=int64)
```

In [10]:
```
# Returns number of rows and columns
train_data.shape
```

Out[10]:
```
(12446, 2)
```

Fig 4: Creating Dataframe

Imbalanced Dataset:

Imbalanced refers to a classification of data set with skewed class proportions. Majority classes are those that represent a sizable share of the data set. Minority classes are those that make up a smaller percentage.

Available Dataset

In [11]:
```
# Getting the count of each class (Normal, Cyst, Tumor, Stone)
cases_count = train_data['label'].value_counts()
cases_count
```

Out[11]:
```
1    5077
0    3709
3    2283
2    1377
Name: label, dtype: int64
```

In [12]:
```
# Plotting the Graph
plt.figure(figsize = (8,6))  # Size of graph
sns.barplot(x = cases_count.index, y = cases_count.values)
plt.title('Number of Cases', fontsize=14)
plt.xlabel('Case Type', fontsize = 12)
plt.ylabel('Count', fontsize = 12)
plt.xticks(range(len(cases_count.index)),['Cyst(0)', 'Normal(1)','Stone(2)','Tumor(3)'])
plt.show()
```

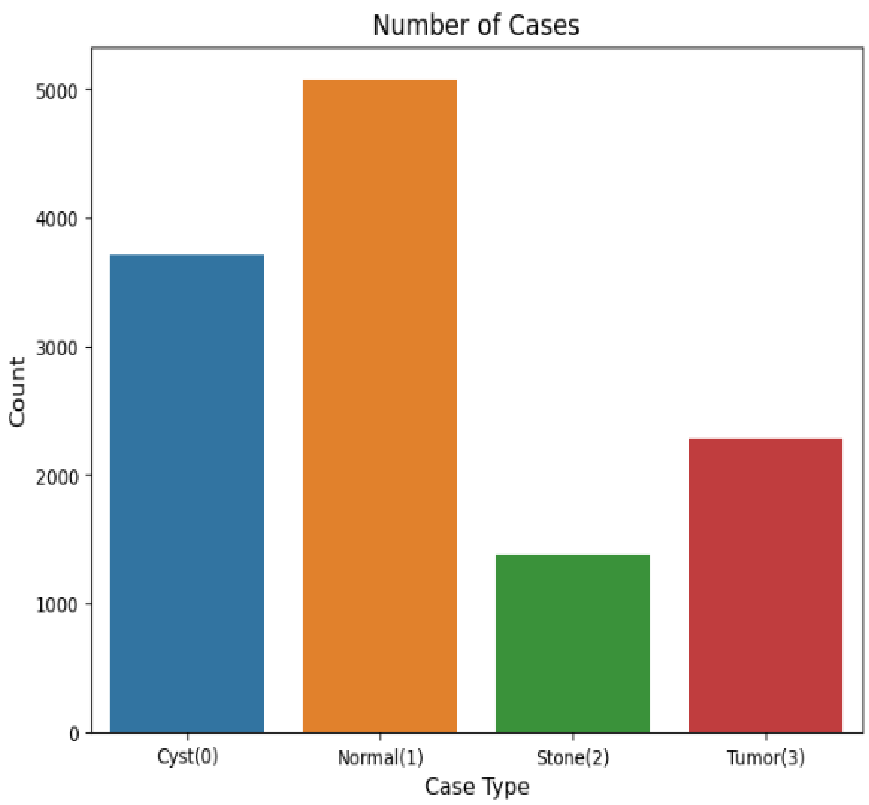

Fig 5: Imbalance Dataset

CT Scanned Image:

CT Scanned Images Available

In [13]:

```
# Getting few samples for both the classes

# Gets 5 data from each cases
Cyst_samples = (train_data[train_data['label'] == 0]['image'].iloc[:5]).tolist()
Normal_samples = (train_data[train_data['label'] == 1]['image'].iloc[:5]).tolist()
Stone_samples = (train_data[train_data['label'] == 2]['image'].iloc[:5]).tolist()
Tumor_samples = (train_data[train_data['label'] == 3]['image'].iloc[:5]).tolist()

# Combining data in one variable
samples = Cyst_samples + Normal_samples + Stone_samples + Tumor_samples

del Cyst_samples, Normal_samples, Stone_samples, Tumor_samples
```

In [14]:

```
# Displaying the picture
f, ax = plt.subplots(4, 5,figsize=(30,30)) # Initilizing the graph where image is to be display

for i in range(20):
    img = imread(samples[i]) # reading the image
    ax[i//5, i%5].imshow(img, cmap='gray') # displaying the image

    # putting title in the images
    if i<5:
        ax[i//5, i%5].set_title("Cyst_samples")
    elif i<10:
        ax[i//5, i%5].set_title("Normal_samples")
    elif i<15:
        ax[i//5, i%5].set_title("Stone_samples")
    elif i<20:
        ax[i//5, i%5].set_title("Tumor_samples")

    # removing the scales in the graph
    ax[i//5, i%5].axis('off')
    ax[i//5, i%5].set_aspect('auto')

plt.show()
```

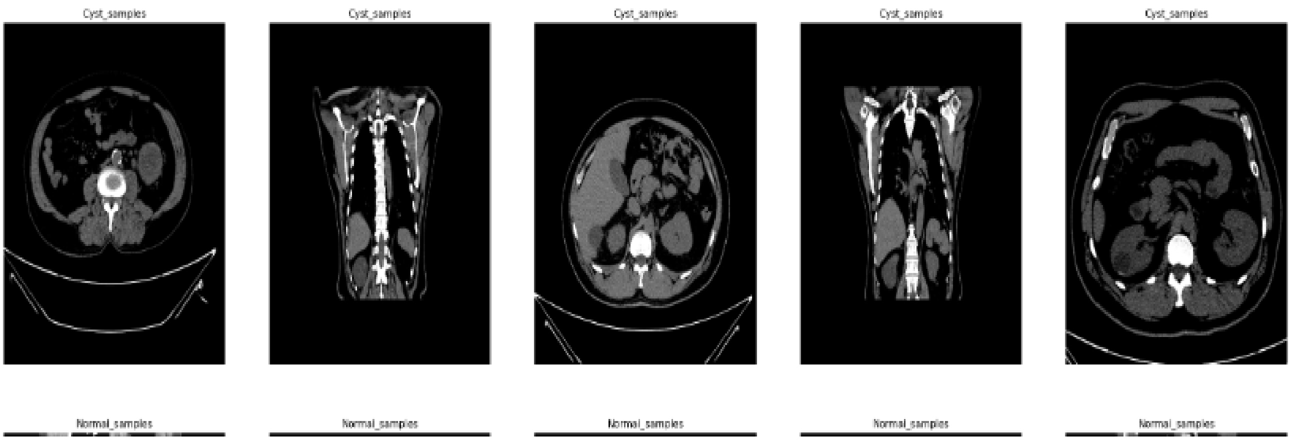

Fig 6: CT scanned Image

Balancing Dataset using SMOTE:

One of the most popular oversampling techniques to address the Imbalance issue is SMOTE (synthetic minority oversampling technique).

By increasing minority class samples at random and duplicating them, it attempts to even out the distribution of groups.

Solving Image Dataset Imblance Using SMOTE

```
In [30]: from imblearn.over_sampling import SMOTE
smt = SMOTE() # Initilizing The SMOTE class
train_rows = len(train_data1) # getting total number or rows
train_data1 = train_data1.reshape(train_rows,-1)  # Converting 4D array to 2D Array
train_data2, train_labels2 = smt.fit_resample(train_data1,train_labels1) # Balancing Image Dataset using SMOTE
```

Balanced Dataset after using SMOTE

```
In [49]: cases_count1 = train_labels2['label'].value_counts() # Counting values of diffrent image

#Plotting Graph for Label values
plt.figure(figsize=(8,6)) # Setting size of graph
sns.barplot(x=cases_count1.index, y=cases_count1.values)
plt.title('Number of cases', fontsize = 14)
plt.xlabel('Case Type',fontsize = 12)
plt.ylabel('Count', fontsize = 12)
plt.xticks(range(len(cases_count1.index)), ['Cyst(0)', 'Normal(1)', 'Stone(2)', 'Tumor(3)'])
plt.show()
```

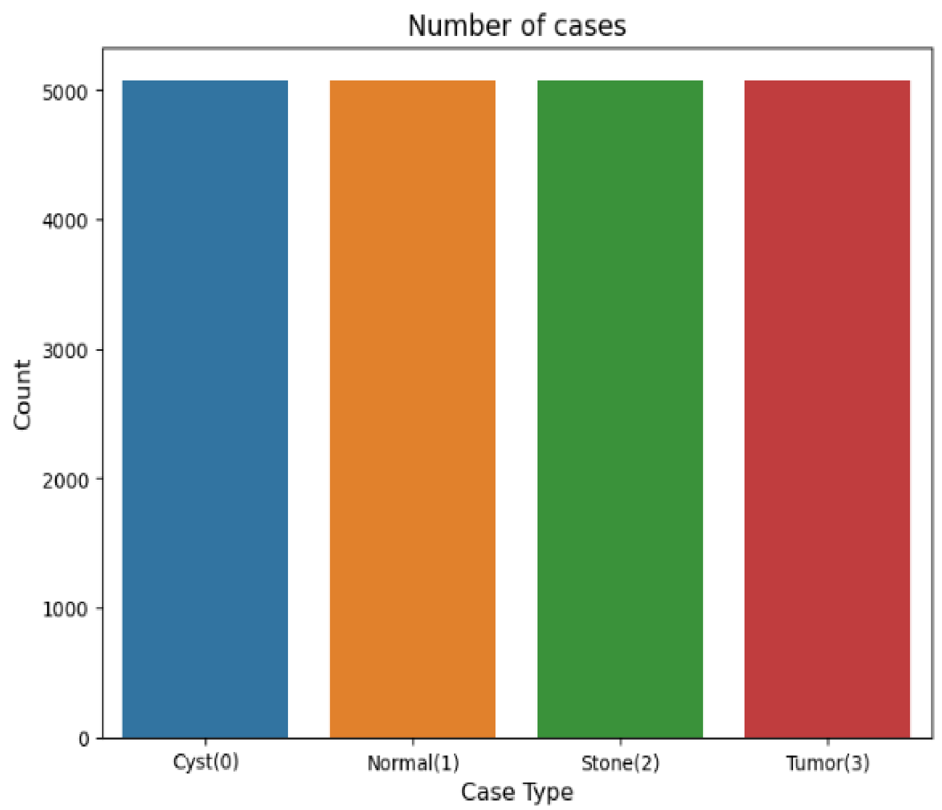

Fig 7: Balanced Dataset

References:

[1] G. Chen et al., "Prediction of Chronic Kidney Disease Using Adaptive Hybridized Deep Convolutional Neural Network on the Internet of Medical Things Platform," in IEEE Access, vol. 8, pp. 100497-100508, 2020, doi: 10.1109/ACCESS.2020.2995310.

[2] H. Zhang, Y. Chen, Y. Song, Z. Xiong, Y. Yang and Q. M. Jonathan Wu, "Automatic Kidney Lesion Detection for CT Images Using Morphological Cascade Convolutional Neural Networks," in IEEE Access, vol. 7, pp. 83001-83011, 2019, doi: 10.1109/ACCESS.2019.2924207.

[3] A. Soni and A. Rai, "Kidney Stone Recognition and Extraction using Directional Emboss & SVM from Computed Tomography Images," 2020 Third International Conference on Multimedia Processing, Communication & Information Technology (MPCIT), 2020, pp. 57-62, doi: 10.1109/MPCIT51588.2020.9350388.

[4] T. Mangayarkarasi and D. N. Jamal, "PNN-based analysis system to classify renal pathologies in Kidney Ultrasound Images," 2017 2nd International Conference on Computing and Communications Technologies (ICCCT), 2017, pp. 123-126, doi: 10.1109/ICCCT2.2017.7972258.

[5] P. Vaish, R. Bharath, P. Rajalakshmi and U. B. Desai, "Smartphone based automatic abnormality detection of kidney in ultrasound images," 2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom), 2016, pp. 1-6, doi: 10.1109/HealthCom.2016.7749492.

[6] N. R. Thomas and J. Anitha, "An Automated Kidney Tumor Detection Technique from Computer Tomography Images," 2022 International Conference on Computing, Communication, Security and Intelligent Systems (IC3SIS), 2022, pp. 1-6, doi: 10.1109/IC3SIS54991.2022.9885650.10

1.
2.

Signature of Students   Signature of Guide