

## AN IMPROVED BUTTERFLY-NET BASED DETECTION OF STONES IN KIDNEY IMAGES

Dr.P.Karthikeyan, Arun.Pa, EkanthNithish P.G.V and Sanjay M.S

Department of ECE

Velammal College of Engineering and Technology

Madurai- 625009

### ABSTRACT

Kidney stone disease is one of the major life threatening problem existing worldwide. It is important to recognize the true and exact location of kidney stone. The ultrasound images of kidney comprise speckle noise and are of low contrast which makes the identification of kidney abnormalities a challenging task. Thus, the doctors may find recognition of small stones difficult. Butterfly-net model is a low-complexity convolutional neural network with sparse cross channel connections and with pre-processing techniques for kidney stone detection. Integrating Serious problems are well approximated with network complexity based on frequency bandwidth rather than input dimension using a butterfly-net with a fully connected neural network. The proposed Butterfly-net obviously embraces the arranged construction in the Butterfly plot as the stacked convolutional layers. In the event that the boundaries are hard-coded as that in the Butterfly plot, Butterfly-net on the whole registers the Fourier coefficients of the info signal with ensured mathematical exactness.

**Keywords:** Butterfly-Net , Kidney Stone, Deep Learning.

### 1 Introduction

#### 1.1 OVERVIEW OF KIDNEY STONES

Kidney stones are mainly lodged in the kidney(s). Humans has been troubled by urinary stones since centuries dating back to 4000 B.C., and it is the most common disease of the urinary tract. Prevention of recurrence of renal stones remains a major concern in human health. Kidney stones have been related with an expanded gamble of constant kidney sicknesses, end-stage renal failure, cardiovascular diseases, diabetes, and hypertension. Kidney stones have been considered as a systemic illness associated to metabolic syndrome. Nephrolithiasis is responsible for 2 to 3% of end-stage renal cases if it is associated with nephrocalcinosis. The symptoms of a kidney stone varies based on whether it is in the kidney, ureter, or urinary bladder. Stone development does not create any symptoms at first. Later, the signs of the stone consist of renalcolic , flank pain (pain in the back side), hematuria (bloody urine), obstructive uropathy (urinary tract disease), urinary tract infections, blockage of urine flow, and hydronephrosis (dilation of the kidney).These conditions may result in nausea and vomiting with associated suffering from the stone event Kidney stones have been related with an expanded gamble of persistent kidney sicknesses, end-stage renal disappointment, cardiovascular diseases, diabetes, and hypertension.

#### 1.2 URINARY SYSTEMS AND STONE

The urinary filtrate is created within the glomerulus and passes into the tubules where the amount and content are altered by reabsorption or secretions. Most solute reabsorption occurs within the proximal tubules, whereas fine adjustments to urine composition occur within the distal tubule and collecting

ducts. The loop of Henle serves to concentrate urine composed of 95% water, 2.5% urea, 2.5% mixture of minerals, salts, hormones, and enzymes. In the proximal tubules, glucose, sodium, chloride, and water are reabsorbed and returned to the blood stream together with essential nutrients like amino acids, proteins, bicarbonate, calcium, phosphate, and potassium. within the distal tubule, the acid-base and salt balance of blood is regulated. the situation of stones may vary as indicated in Figure 1.1

The location of stones may vary as indicated in Figure 1.1

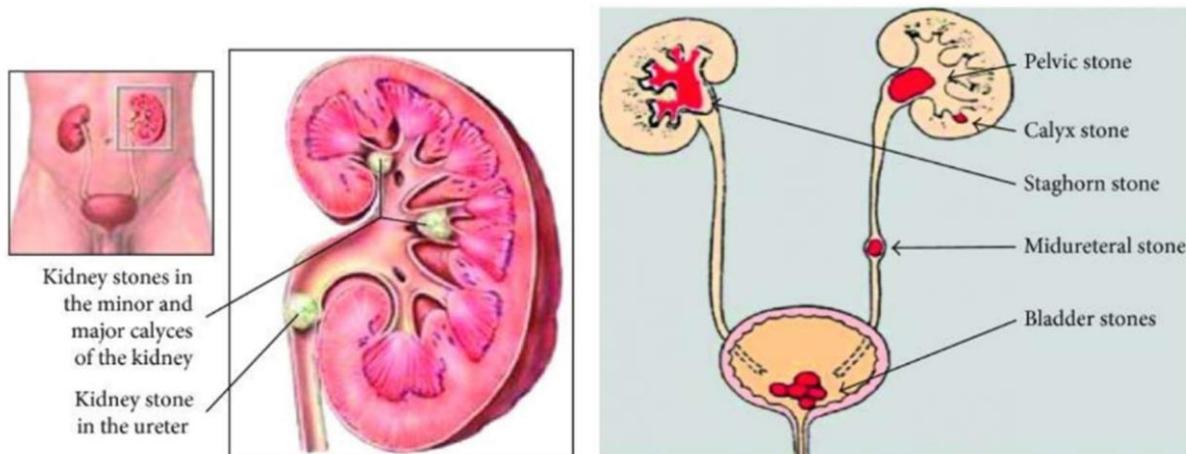


Figure 1.1: Kidney stone locations in the urinary system

## 2 DATASET

The data was provided by the Kaggle. The entire dataset was composed of 300 actual patient CT scans. Fifty percent of the dataset were used for the training set, and the remaining 50% scans were retained for testing predictions; no segmentation information is provided for this. We chose to use a validation set instead of cross-validation because cross-validation would necessitate us to multiply the computational time by the count of folds we had and train an independent ensemble of neural networks for each fold.

## 3 PROPOSED SYSTEM

We looked into several model architectures as potential solutions to this problem. Primarily, we looked at Butterfly-net configurations and decided on it as our final configuration after testing many variations. The below section introduces the proposed architecture in detail.

### 3.1 PREPROCESSING

The goal of pre-processing is to improve image quality so that we can analyse it more effectively. We can suppress unwanted distortions and enhance some features that are required for the application we're working on by pre-processing.

#### 3.1.1 GABOR FILTER

The Gabor filter is a bandpass filter that is used in image processing for feature extraction, texture analysis, edge detection, and other similar tasks. A Gabor filter can be thought of as a sinusoidal signal

modulated by a Gaussian wave with a specific frequency and orientation. The weights are provided by the gaussian component, while the directionality is provided by the sine component. When a Gabor filter is applied to an image, the response is strongest at the edges.

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \exp\left(i\left(2\pi \frac{x'}{\lambda} + \psi\right)\right)$$

Where,

$$x' = x \cos \theta + y \sin \theta, y' = -x \sin \theta + y \cos \theta$$

$\lambda$  - wavelength of the sine wave

$\theta$  represents orientation of the filter

$\psi$  is the phase offset

$\sigma$  is the S.D of Gaussian envelope

$\gamma$  is the spatial aspect ratio

### 3.1.2 LAPLACIAN FILTER

Laplacian filters are derivative filters that extract the image's vertical and horizontal edges. Because derivative filters are highly sensitive to noise, it is common practise to smooth the image prior to applying the Laplacian. This determines whether a change in adjacent pixel values is caused by an edge or by a continuous progression. Negative values in a cross pattern, centered within the array, are common in Laplacian filter kernels. The corners can be either zero or positive. The center value can be positive or negative.

### 3.1.3 HISTOGRAM EQUALIZATION

To boost contrast, histogram equalization is used. Histogram equalization is a sophisticated method for modifying an image's dynamic range and contrast by modifying the image's intensity histogram to have the desired shape. They are used to equalize all of an image's pixel values. The transformation produces a uniformly flattened histogram. Histogram equalization broadens the dynamic range of pixel values and produces an image with a flat histogram and high contrast.

### 3.2 BUTTERFLY-NET

In this project, the Butterfly-net is proposed to classify kidney stone using an image processing. This Butterfly-net model is a low-complexity CNN with structured and sparse cross channel connections,

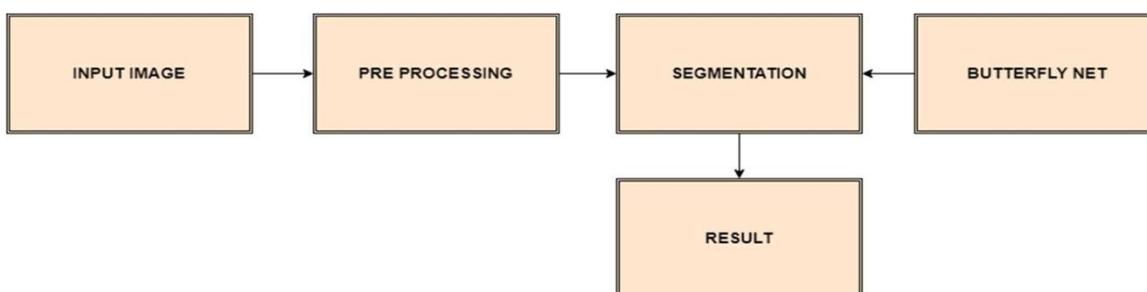


Figure 2.1: Proposed Block Diagram

jointly with a Butterfly initialization strategy for a family of networks. Hypothetical investigation of the estimate force of BN to the Fourier documentation of info information shows that the mistake rots

quickly as the profundity increments. When BN is combined with a fully connected neural network, a large class of problems is shown to be well approximated with network complexity depending on effective frequency bandwidth instead of the input dimension. Usual CNN is covered as a special case in our computation. Numerical experiments check the results on the approximation of Fourier kernels and energy functionals of equations of Poisson's.

Furthermore, all trials show that Butterfly initialization training beats random initialization training. Moreover, while increasing the number of cross-channel connections increases the parameter number, it does not improve post-training accuracy and is more susceptible to data distribution. This paper presents a particular architecture based on the Butterfly scheme, which was originally developed for the rapid calculation of Fourier integral operators and function transforms.

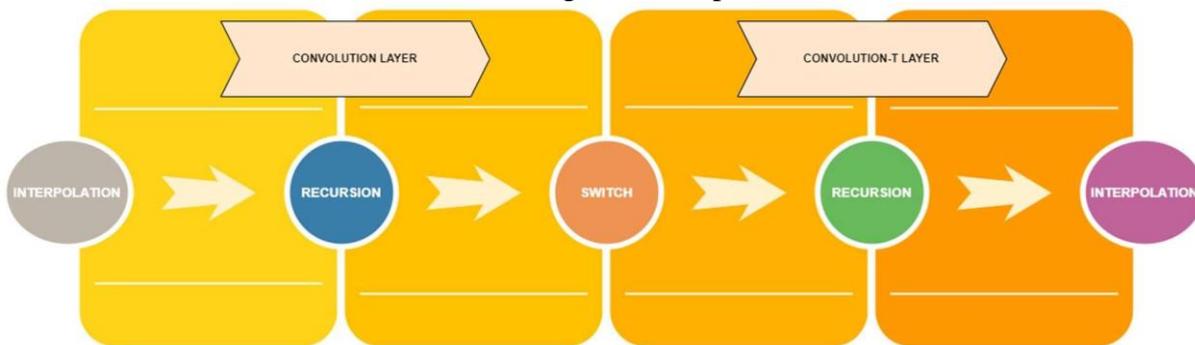


Figure 2.2 Butterfly net architecture

For Fourier kernels and Fourier integral operators, the technique is near optimal in terms of computational complexity. The suggested BN uses the Butterfly scheme's hierarchical structure as the stacked convolutional layers. The Fourier coefficients of the input signal are computed collectively by Butterfly-net with numerical precision, if the inputs are hardcoded as in the proposed scheme. In contrary to regular CNN, the Butterfly-net channels have unambiguous correspondences to the frequency bands, especially the placement in the spectral representation of the signal, and the cross-channel weights are sparsely connected. CNN is a unique Inflated-Butterfly-network. Butterfly-net is substantially lighter than Inflated Butterfly-net: the model complexity (in terms of parameter number) is  $O(K \log N)$ , and the calculation complexity is  $O(N \log N)$ , where  $K$  is the frequency bandwidth, and  $N$  is the length of the discrete I/P signal.

The complexity of the calculation is  $O(N \log N)$ , where  $N$  is the length of the discrete input signal and is the bandwidth. The Butterfly-net approximation error in modelling Fourier kernels is shown to exponentially diminish as network depth grows, which is statistically proven in Section 5. Butterfly-net has all of the approximation properties of the Fourier representation of input signals, thanks to efficient Fourier kernel approximation. In image and signal processing, it's especially useful for solving PDEs and (local) Fourier-based techniques. The accuracy assurance for Butterfly-net to represent Fourier kernels is logically guaranteed.

### 3.3 BUTTERFLY-NET ARCHITECTURE

1. Interpolation ( $l = 0$ ). Let  $m = N/(2^L)$  denotes the filter size, which relates to the no. of points in each  $B^L_j$ . A 1-dimensional convolution layer with filter size  $m$ , stride size  $m$  and output channel  $r$  is applied to  $x(:, 1)$  together with added ReLU activation and bias term. The weight tensor is denoted as  $M^{(0)}_{n,p,1}$ , where  $k = 1, \dots, r$  and  $q = 1, \dots, z$ . The layer is converted as I/P tensor  $x$  to an O/P tensor  $\lambda(0)$ . The below weight tensor is denoted as,

$$M_{n,p,1}^{(0)} \triangleq e^{-2\pi 2\xi_0 \cdot (t_q - t_k)} \mathcal{L}_k(t_q), 1 \leq k \leq r \text{ and } 1 \leq p \leq z$$

where  $\triangleq$  denotes extended assign operator as defined. This step interpolates function from uniform grid points to Chebyshev points. When the frequency domain of the input signal is not symmetric around origin, this step also extracts extra phase term.

2. Recursion ( $l = 1, \dots, L_t$ ): The input and output tensors at layer  $l$  are  $\lambda^{(l-1)}$  and  $\lambda^{(l)}$ . 1 (for  $L_t \geq 1 > L_{\min}$ ) or 2 (for  $L_{\min} \geq 1$ ) 1-D convolution layers having filter size of 2, stride 2 and output channel  $r$  are applied with ReLU activation and bias terms for each non-mixing channel at preceding layer. The weight tensors are denoted as  $M_{n,d,v}^{(\ell),i}$  where  $n, v = 1, \dots, r$  and  $d$ . The weight tensor is denoted using the notations as follows:

$$M_{n,d,v}^{(\ell),i} \triangleq e^{-2\pi i \xi_0^i \cdot (t_v^d - t_n)} \mathcal{L}_n(t_v^d)$$

where  $\xi_0^i$  denotes the centre of  $X_i^1$ ,  $t_v^d$  denotes the Chebyshev points in  $C_d = B_d^{L-1+1}$  and  $t_n$  denotes the Chebyshev points in  $B_0^{L-1}$ . Each  $\xi_0^i$  is the center of  $X_i^1$  corresponding to different frequency domain. Different frequency component in the input signal is now organized in different non-mixing channels. Later on, they will be converted independently which is connected to orthogonality of basis function in non overlapping frequency domain.

3. Switch ( $l = L_t$ ). This is a special layer of local operations. Denote the input tensor as  $\lambda^{(L_t)}(i, j, v)$  and the dense weights as  $M_{n,v}^{(L_t),i,j}$  for  $k, s = 1, \dots, r$  and  $i, j$  with range. For each  $i, j$ ,  $M^{(L_t)}$  is a  $r$  by  $r$  dense matrix. The operation at the layer is as follows,

$$\lambda^{(L_t)}(i, j, v) = \sum_{v=1}^r M_{n,v}^{(L_t),i,j} \lambda^{(L_t)}(i, j, v)$$

on every  $i$  and  $j$  pair. The O/P tensors are given a bias term and a ReLU layer. The dense weight tensors can be initialised as, following the notation in the dense weight tensors,

$$M_{n,v}^{(L_t),i,j} \triangleq e^{-2\pi i \xi_n^{L_t} \cdot t_v^{B_j^L \xi}}$$

where  $\xi_n^{L_t}$  and  $t_v^{B_j^L \xi}$  are Chebyshev points in  $X_i^{L_t}$  and  $B_j^L \xi$  respectively. For each  $i, j$ , the Fourier operator is used to the layer. Interpolation is applied again in frequency domains.

4. Recursion ( $l = L_t + 1, \dots, L_t + L_\xi$ ). The I/P and O/P tensors at layer  $l$  are  $\lambda^{(l-1)}$  and  $\lambda^{(l)}$ . The weight tensors are denoted as  $M_{n,d,v}^{(\ell),j}$ , where  $n, v = 1, \dots, r$  and  $d$ . One 1-dimensional convolution layer is performed for each non-mixing channel  $j$  as follows:

$$\lambda^{(\ell)}(i, j, n) = \sum_{v=1}^r \sum_{d=0,1} M_{n,d,v}^{(\ell),j,a} \lambda^{(\ell-1)}(\lfloor i/2 \rfloor, 2j + d, v)$$

where  $a$  is  $i \bmod 2$ . This convolution is called as  $t$ -convolution. The weight tensor can be initialised using the notations as follows:

$$M_{n,d,v}^{(\ell),j,a} \triangleq e^{-2\pi i (\xi_n^a - \xi_v) \cdot t_0^{2j+d}} \mathcal{L}_v(\xi_n^a)$$

where  $t_0^{2j+d}$  denotes the center of  $C = B_{2j+d}^{L-\ell+1}$ ,  $\xi_v$  denotes the Chebyshev points in  $X_0^{L-1}$ ,  $\xi_n^a$  denotes

the Chebyshev points in  $X_a^1$  for  $a = 0, 1$ . This is identical to third step. Various time components in the input signal are now structured in different non-mixing channels, rather than output being organised in non-mixing frequency domains. This is owing to the fact that time and frequency are complimentary.

5. Interpolation ( $l = L$ ).

The output channel size is  $m = \frac{K}{2^{L\xi+L_{min}}}$ , The number of points in each  $X_i^L$  correlates to this. A 1-D convolution layer having filter size of 1, stride size 1, input channel  $r$  and output channel  $m$  is applied to  $\lambda^{(L)}$  with added ReLU activation and bias terms.  $M_{q,0,n}^{(L)}$ , where  $q = 1, \dots, m$  and  $n = 1, \dots, r$  is the weight tensor. Following the notations in below equation, the weight tensor can be initialized as,

$$M_{q,0,n}^{(L)} \overset{\circ}{=} e^{-2\pi i(\xi_q - \xi_n) \cdot t_0} \mathcal{L}_n(\xi_q), 1 \leq q \leq m \text{ and } 1 \leq n \leq r.$$

This layer produce the O/P tensor denoted as  $x(i, q)$ , for  $i = 0, \dots, 2^L - 1$  being the index of data and  $p = 1, \dots, m$  being the index of channel. Reshaping  $x(v) = x(i, q)$  for  $v = q + i \cdot m$  yields a single vector O/P.

4 EVALUATION METRICS

Accuracy in classification problems is the number of correct predictions made by the model divided by the total number of predictions.  $[(TN+TP)/(TN+FP+TP+FN)]$

Recall is that the ability of a model to find all the relevant cases within a dataset.  $[TP/TP+FN]$

Precision is that the ability of a classification model to spot only the relevant data points.  $[TP/TP+FP]$

F1-Score is the harmonic mean of recall and precision taking both metrics into account in the following equation:  $F_1 = 2 \times \text{Precision} \times \text{Recall} / \text{Precision} + \text{Recall}$ .

5 RESULTS

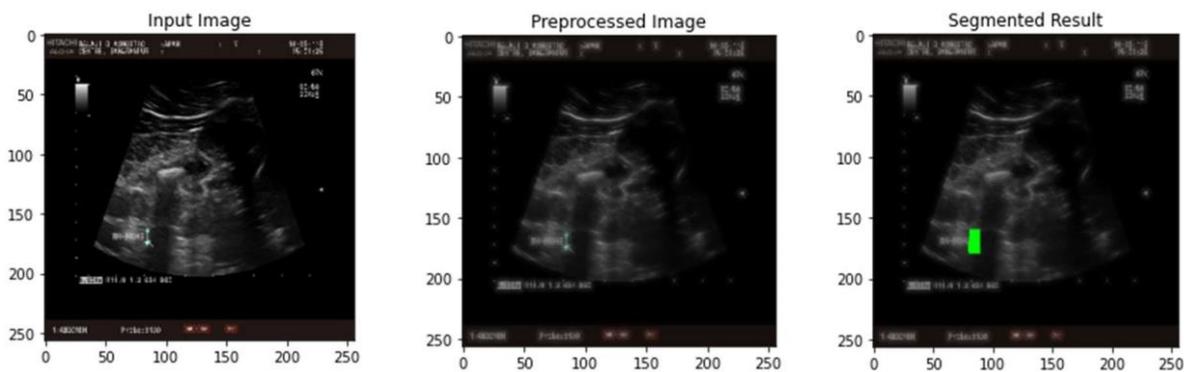


Figure 5 (a)

Figure 5 (b)

Figure 5(c)

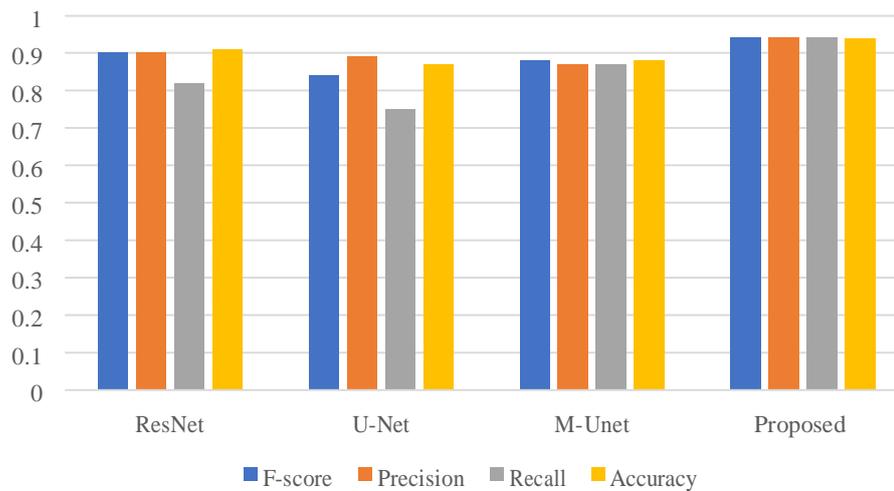
5.1 COMPARATIVE ANALYSIS

The table below shows the performance of our proposed system (Butterfly-Net). We have compared the performance of our proposed system(Butterfly-Net) with the following existing methods :

	ResNet	U-Net	M-Unet	Proposed
F-score	0.9	0.84	0.88	0.94
Precision	0.9	0.89	0.87	0.94
Recall	0.82	0.75	0.87	0.94
Accuracy	0.91	0.87	0.88	0.9383

### 5.1.1 Metric Analysis

Chart Title



### 5.1.2 Metric Analysis (Bar Chart)

## 6 SOFTWARE SPECIFICATION

### 6.1 SOFTWARE REQUIRED

- Python 3.7
- Jupyter Notebook

### 6.2 HARDWARE REQUIRED

- System : Windows
- Processor : Up to 1.5 GHz
- Memory : Up to 512 MB RAM

### 6.3 PYTHON

Python is a high level programming language that is widely used to solve complex problems in machine learning and deep learning, when comparing with other popular languages. Python and its libraries like Numpy, Pandas, Scipy, Matplotlib; frameworks like TensorFlow, Keras. These libraries and frameworks can be applied to solve complex real world problems.

## 7 Conclusion

The research work proposes a novel methodology to recognize the kidney stone from ultra sound images. The proposed strategy has been executed using deep learning approach to complete the classification step. The taken kidney snapshot can be subjected to the proposed set of rules in real time via interfacing it with scanning equipment to become aware of the affected location and for accurate classification of kidney stone. For accomplishing better accuracy, the proposed method uses butterfly net. This method is carried out in python. Comparing with grey other schemes, the proposed method achieves higher accuracy and efficiency.

## REFERENCES

- [1] J. Causey et al., "An Ensemble of U-Net Models for Kidney Tumor Segmentation with CT images," in *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, doi:10.1109/TCBB.2021.3085608.
- [2] Yingzhou Li, Xiuyuan Cheng, Jianfend Lu "Butterfly-Net: Optimal Function Representation Based on Convolutional Neural Networks" <https://doi.org/10.48550/arXiv.1805.07451>.
- [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] N. Thein, H. A. Nugroho, T. B. Adji and K. Hamamoto, "An image preprocessing method for kidney stone segmentation in CT scan images," 2018 International Conference on Computer Engineering, Network and Intelligent Multimedia (CENIM), 2018, pp. 147-150, doi: 10.1109/CENIM.2018.8710933.
- [5] MahdiMarsousi , Konstantinos N. Plataniotis and StergiosStergiopoulos "Shape-based kidney detection and segmentation in three-dimensional abdominal ultrasound images" DOI:10.1109/EMBC.2014.6944227.
- [6] 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.
- [7] N. Thein, K. Hamamoto, H. A. Nugroho and T. B. Adji, "A comparison of three preprocessing techniques for kidney stone segmentation in CT scan images," 2018 11th Biomedical Engineering International Conference (BMEiCON), 2018, pp. 1-5, doi: 10.1109/BMEiCON.2018.8609996.
- [8] T. Shah and S. Kadge, "Analysis and Identification of Renal Calculi in Computed Tomography Images," 2019 International Conference on Nascent 49 Technologies in Engineering (ICNTE), 2019, pp. 1-4, doi: 10.1109/ICNTE44896.2019.8945877.
- [9] H. Dave, V. Patel, J. N. Mehta, S. Degadwala and D. Vyas, "Regional Kidney Stone Detection and Classification In Ultrasound Images," 2021 Third International Conference on Inventive Research in Computing Applications (ICIRCA), 2021, pp. 1108-1112, doi: 10.1109/ICIRCA51532.2021.9545031.

- [10] S. M B and A. M R, "Kidney Stone Detection Using Digital Image Processing Techniques," 2021 Third International Conference on Inventive Research in 50 Computing Applications (ICIRCA), 2021, pp. 556561, doi: 10.1109/ICIRCA51532.2021.9544610.
- [11] L. Y. Myint, S. S. Maung and K. T. Zar, "Removal of Unwanted Object in 3D CT Kidney Stone Images and 3D Visualization," 2020 24th International Computer Science and Engineering Conference (ICSEC), 2020, pp. 1-5, doi: 10.1109/ICSEC51790.2020.9375155.
- [12] S. Rajput, A. Singh and R. Gupta, "Automated Kidney Stone Detection Using Image Processing Techniques," 2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 2021, pp. 1-5, doi: 10.1109/ICRITO51393.2021.9596175.
- [13] M. K. Shahina and H. S. Mahesh, "Renal Stone Detection and Analysis by Contour Based Algorithm," 2019 International Conference on Recent Advances in Energy-efficient Computing and Communication (ICRAECC), 2019, pp. 1-5, doi: 10.1109/ICRAECC43874.2019.8994967.
- [14] M. Ranjitha, "Extraction and dimensionality reduction of features for Renal Calculi detection and artifact differentiation from segmented ultrasound kidney images," 2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom), 2016, pp. 3087-3092.
- [15] Y. Li, E. Chouzenoux, B. Charmettant, B. Benatsou, J. -P. Lamarque and N. Lassau, "Lightweight U-Net For Lesion Segmentation In Ultrasound Images," 2021 IEEE 18th International Symposium on Biomedical Imaging (ISBI), 2021, pp. 611-615, doi: 10.1109/ISBI48211.2021.9434086.