

Kidney Disease detection and classification from CT Images using Watershed Segmentation and Deep Learning.

by

Mohammad Sakib Hossain

18341001

S.M. Nazmul Hassan

18301171

Md. Nakib rahaman

18301203

Mohammad Al-Amin

18301259

Rakib Hossain

18301187

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science

Department of Computer Science and Engineering
Brac University
September 2022

© 2022. Brac University
All rights reserved.

Declaration

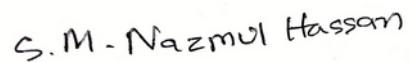
It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:



Mohammad Sakib Hossain
18341001



S.M. Nazmul Hassan
18301171



Md. Nakib Rahaman
18301203



Mohammad Al-Amin
18301259



Rakib Hossain
18301187

Approval

The thesis/project titled “Kidney Disease detection and classification from CT Images using Watershed Segmentation and Deep Learning.” submitted by

1. Mohammad Sakib Hossain (18341001)
2. Md. Nakib Rahaman (18301203)
3. S.M. Nazmul Hassan (18301171)
4. Mohammad Al-Amin (18301259)
5. Rakib Hossain (18301187)

Of Summer, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on September 25, 2015.

Examining Committee:

Supervisor:
(Member)

Dr. Muhammad Iqbal Hossain
Associate Professor
Department of Computer Science and Engineering
BRAC University

Co-Supervisor:
(Member)

Ms. Arnisha Khondaker
Lecturer
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Md. Golam Rabiul Alam
Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Ms. Sadia Hamid Kazi
Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

Chronic kidney disease, often called chronic kidney failure, is a steady decline of renal function. Some of the most common reasons for kidney failure are cyst, stone and tumor. There may be no symptoms of chronic renal disease in its first stages. However, It's possible to have kidney disease and not know it until it's too late. Fortunately various neural networks have been shown to be beneficial in early disease prediction as machine learning and computer science has progressed. In this paper, we have used 5 CNN classification methods that are based on watershed segmentation and make use of deep neural networks (DNN) to classify 4 types (cyst,normal,stone,tumor) of kidney CT images. There are two stages to our work. We have first segmented the region of choice in CT images by watershed algorithm. The segmented kidney data was then used to train a variety of classification networks, which includes EAnet and the transfer learning based pre-trained neural networks: ResNet50, VGG19, InceptionV3, and SqueezeNet. Our models were trained using the CT Kidney Normal Cyst Tumor and Stone dataset that was made public on Kaggle. Finally, EAnet, SqueezeNet, VGG19, InceptionV3, and ResNet50 achieved 83.6%,97.3%,99.9%,98.8% and 87.9% of accuracy, respectively, on the test set of classification models. We observed that the modified VGG19 model had the highest sensitivity and specificity as well as the best overall accuracy.

Keywords: Watershed Algorithm ; InceptionV3; Squeezenet; ResNet50; VGG19; Transfer Learning; EAnet.

Dedication

We appreciativenessly and warmly regard that we are dedicating our to our family members and well wishers, who have supported us throughout the whole process of this work.

Acknowledgement

Firstly, thanks to the Almighty Allah for allowing us to finish our thesis without any serious setbacks. Second, we'd want to express our gratitude to our supervisor, Dr. Muhammad Iqbal Hossain sir, for all of his assistance and guidance throughout this project. When we needed help, he was there for us. Finally, we'd like to thank our co-supervisor, Ms. Arnisha Khondaker, for all of her help and support throughout this thesis process. They were there for us the whole way through, offering guidance and encouragement at every stage of the thesis process. We owe Md Nazmul Islam and Mehedi Hasan a debt of gratitude for providing us with their dataset. Finally, we owe a debt of gratitude to our parents, without whose unwavering encouragement none of this would have been possible. Thanks to their encouragement and prayers, we're almost ready to walk across the stage.

Table of Contents

Declaration	i
Approval	ii
Ethics Statement	iv
Abstract	iv
Dedication	v
Acknowledgment	vi
Table of Contents	vii
List of Figures	ix
List of Tables	x
Nomenclature	xii
1 Introduction	1
1.1 Problem Description	2
1.2 Research Objectives	2
1.3 Thesis Overview and Orientation	3
2 Related Work	4
3 Proposed Model	12
3.1 Data Collection	13
3.2 Data Preprocessing	13
3.2.1 Data Spiting	13
3.2.2 Data Resizing	14
3.2.3 Segmentation using Watershed Algorithm	14
3.2.4 Data Normalization	16
3.2.5 Data Augmentation	16
3.3 Classification	17
3.4 Transfer Learning	17
3.4.1 VGG19	19
3.4.2 ResNet50	20
3.4.3 InceptionV3	22

3.4.4	Squeezenet	23
3.5	EAnet	24
4	Result and Analysis	27
4.1	Experimental Evaluation	27
4.1.1	Accuracy	27
4.1.2	Loss	27
4.1.3	Sensitivity, Precision and F1 Score	28
4.1.4	ROC Curve and AUC	28
4.2	Implementation	28
4.3	Analysis of Results	29
4.3.1	ResNet50	29
4.3.2	VGG19	31
4.3.3	InceptionV3	33
4.3.4	EAnet	35
4.3.5	SqueezeNet	37
4.4	Discussion of models	39
5	Conclusion	41
	Bibliography	45

List of Figures

3.1	Workflow.	12
3.2	Distribution of CT KIDNEY DATASET.	13
3.3	The flowchart of data segmentation using marker based Watershed segmentation.	15
3.4	CT images after applying Watershed segmentation.	16
3.5	Augmented images of an image of each class.	17
3.6	Fixed Feature Extraction of Transfer Learning.	18
3.7	The Model Architecture of VGG19.	20
3.8	The Model Architecture of Resnet50.	21
3.9	The Model Architecture of InceptionV3.	22
3.10	The Model Architecture of Squeezenet.	24
3.11	EAnet’s External Attention.	25
4.1	Train and Validation Accuracy graph of ResNet50.	29
4.2	Train and Validation Loss of ResNet50.	29
4.3	ROC Curve of ResNet50.	30
4.4	Confusion Martrix of ResNet50.	30
4.5	Train and Validation Accuracy graph of VGG19.	31
4.6	Train and Validation Loss of VGG19.	31
4.7	ROC Curve of VGG19.	32
4.8	Confusion Martrix of VGG19.	32
4.9	Train and Validation Accuracy graph of InceptionV3.	33
4.10	Train and Validation Loss of InceptionV3.	33
4.11	ROC Curve of InceptionV3.	34
4.12	Confusion Martrix of InceptionV3.	34
4.13	Train and Validation Accuracy graph of EAnet.	35
4.14	Train and Validation Loss of EAnet.	35
4.15	ROC Curve of EAnet.	36
4.16	Confusion Martrix of EAnet.	36
4.17	Train and Validation Accuracy graph of SqueezeNet.	37
4.18	Train and Validation Loss of SqueezeNet.	37
4.19	ROC Curve of SqueezeNet.	38
4.20	Confusion Martrix of SqueezeNet.	38

List of Tables

3.1	Dataset distribution after splitting.	14
4.1	Classification report of all models.	39

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

Adam Adaptive Moment estimation

AUC Area under the Curve

BiLSTM Bidirectional Long Short-Term Memory

CCT Compact Convolutional Transformers

CGAN Conditional Generative Adversarial Networks

CKD Chronic Kidney Disease

CNN Convolutional Neural Network

CT Computed Tomography

DL Deep Learning

DNN Deep Neural Network

DSC Dice Similarity Coefficient

EHR Electronic Health Record

FC Fully Connected Layer

FPR False Positive Rate

IoMT Internet of Medical Things

ML Machine Learning

MRI Magnetic Resonance Inaging

MRI Magnetic Resonance Inaging

NN Neural Network

RCC Renal Cell Carcinoma

RCC Renal Cell Carcinoma

ReLU The rectified linear activation function

ResNet Residual Network

RMSProp Root Mean Squared Propagation

ROC Receiver Operating Characteristic

SGD Stochastic Gradient Descent

SVM Support vector Machine

TCIA The Cancer Imaging Archive

TPR True Positive Rate

VGG19 Very Deep Convolutional Networks for Large-Scale Image Recognition-16

VGG19 Very Deep Convolutional Networks for Large-Scale Image Recognition-19

AHDCNN An Adaptive Hybridized Deep Convolutional Neural Network

Chapter 1

Introduction

Kidney disease is nowadays increasing at an alarming rate. In the whole world it is about 10% people who have been suffering with chronic kidney diseases[7]. Among the leading causes of death it has been ranked as 16th. Nevertheless, there is a high chance that the ranking of this disease will be in the 5th position around 2040. Cyst, Stone in the kidney is mainly malfunctioning the kidney and giving one's painful death. A cyst in the kidney is basically a pocket filled with fluid that exists on the kidney's surface and also a thin wall is bounded by it [18]. On the contrary, kidney stone is a crystal concentration in kidneys by which 12% people around the world get affected [13]. A higher risk of kidney failure leading to dialysis or transplantation has been linked to it. Kidney stones have a multifaceted etiology. Calcium Oxalate which is the most typical kidney stone develops at Randall's plaque on the surfaces of the renal papillaries. The mechanism of stone development is a complicated process that occurs from a number of physicochemical processes, such as supersaturation, nucleation, growth, aggregation, and retention of urinary stone ingredients inside tubular cells. This culminates in the creation of stones. These processes are governed by a discrepancy between the components that promote and inhibit crystallization in urine. Additionally, It should be focused on particle retention encouraged by the cellular damage on renal papillary surfaces. Oxalate exposure induces a signaling cascade that ultimately leads to apoptosis via p38 mitogen-activated protein kinase pathways in renal epithelial cells. There is currently no effective medication to treat kidney stones or stop them from returning. Therefore, research is needed to better understand the biology of kidney stone development in order to treat urolithiasis with novel medications. As a result, the purpose of this study was to consolidate current knowledge on the pathophysiology, etiology, and prevention of kidney stones. Kidney tumor, or renal cell carcinoma (RCC), is one of the top 10 most common tumors worldwide. The course of the illness and cardiovascular morbidity and death can be avoided if chronic renal disease is diagnosed early. Therefore, early detection techniques in public health are gaining more attention. Pathology tests are frequently used with imaging techniques including X-rays, computed tomography (CT), B-ultrasound devices (US), and MRIs to identify renal problems. The CT scanner uses X-ray beams to scan the required area of the human body and create a cross-sectional picture that shows the area in three dimensions [5]. CT scans of the kidneys are excellent research tools because they give both slice-by-slice pictures and three-dimensional data. Renal failure can develop if kidney problems like cysts, stones, and tumors aren't identified and treated quickly. Consequently, it would

seem that identifying renal illnesses such as kidney cysts, stones, and tumors early on is a crucial step in avoiding kidney failure [37]. However, urologists are in abundance, whereas nephrologists and radiologists are in short supply. Whereas there is just about one nephrologist for every million people in South Asia, there are almost exactly 25.3 nephrologists for every million people in Europe [35].

1.1 Problem Description

As we know ultrasound (UT), MRI, or CT scan should be performed before and also blood and urine tests for diagnosis for Kidney disease which make it a very lengthy process . The human body has a role to make an adjustment of various hormones including creatinine, albumin, and eGFR (estimated) at a constant level. If this hormone production level reaches to an inconsistent level we will find it after a urine test. But a lot of things may change the degree of precision required for Kidney disease diagnosis. Hence, the patient is given a blood test, which is the following trustworthy test. Regardless of the testing method used on the blood sample, it is required for the Donor patient to fast for at least half day before giving the blood sample, or abide by specified dietary and pharmaceutical restrictions. Diagnostic procedures for kidney disease that are increasingly complex also increase the chance of human error.

Various imaging modalities, including ultrasound (UT), (MRI) magnetic resonance imaging , and (CT) computed tomography , have been incorporated into image-based systems for the accurate detection of kidney disorders. Clinic-performed CT scan is the first step. Printing the report takes time.The doctor and radiologist collaborate to identify the illness once the patient gets the CT scan data. Even though it varies depending on the circumstance, the whole procedure takes many important days from the patient's life. The patient's condition worsens day by day. When the kidneys aren't working correctly, the person feels excruciating agony. Kidney function is crucial for maintaining health, living a full life and also to manage proper biological processes. Kidney tumor, cyst and stone removal may be a long procedure, and in certain cases, the cancer has spread which tends to be misdiagnosed or just recently diagnosed. Approximately 50% of individuals are diagnosed with the illness after it has progressed significantly.In this essay, we'll be finding kidney tumor, cyst and stone using CT pictures which will be time and money saving and also produces more precision and accuracy.

1.2 Research Objectives

The goal of this effort is to utilize AI to detect kidney tumors, cysts, and stones without the intervention of a human observer. The time it takes from receiving the results of a CT scan to beginning effective therapy would be cut down significantly with the streamlined system.

The following is a list of the objectives of this study:

- To get a clear insight of CT reports and the operation of DNN.

- To gain an in-depth understanding of how WaterShed Segmentation may be used to CT scans to segment the kidney images and identify tumors, cysts, and stones.
- To conduct an analysis on the models that were used.
- To make suggestions for improving the overall performance of the models.

1.3 Thesis Overview and Orientation

The following sections break down the work we've done into chapters and provide an in-depth analysis of our methodology. In chapter 2, we addressed similar prior research on neural networks and renal disease prediction done by other researchers and how they used their research in their investigations. The methodology we have presented for this study is then discussed at length in chapter 3. We then presented an overview of our system and how we preprocessed the data. Our segmentation techniques and a summary of the CNN models are also included in this chapter. The subsequent chapter, 4, is devoted to the analysis and interpretation of our study's findings. In this chapter, we also see the training models' and graphs' final products. In the final chapter, we summarized our findings and spoke about directions for moving forward.

Chapter 2

Related Work

In the paper by AV Gregory and DA Anaam (2021) [31] reconstructed the goal of detecting or locating object and the segmentation job which is instance-based into a problem (semantic segmentation) , and created the initially 3rd dimension semantic instance cyst segmentation method in MR images for kidney. Using a 3rd dimension watershed method on a 3D Euclidean distance map of generated cyst segmentation from the MRI pictures,from the manual cyst tracings they have built an initial instance of cyst segmentation. One label cyst segmentation was performed using a previously established automated semantic cyst segmentation technique to speed up the cyst segmentation procedure for the validation and training sets. The 3rd dimension watershed procedure explained in the prior section was applied to that resulting segmentation. They employed the U-Net architecture and four-fold cross-validation sets for the training and validation phases.

Research utilizing radiological imaging scans which is for the early diagnosis of kidney tumors is discussed in the publication [39] by Maha Gharaibeh and their co-authors. This review covers related research in the areas of DL(deep learning), data analytics, and ML (machine learning). The various kidney disorders and cancers were discussed. Radiologic imaging techniques for the kidney were also discussed and evaluated, including ultrasound, magnetic resonance imaging, computed tomography, and angiography (cta). In addition, they discussed various early detection methods for kidney tumors and made comparisons using ML(machine learning) and DL(deep learning). From what they've gathered, deep learning appears to be more effective than ML(machine learning) in analyzing medical photos. Nonetheless, they all used previous models rather than developing their own, and they failed to fill the void with an original CNN model for KT diagnosis.

Cong Chen, Longfei Ma, Yan Jia, and Panli Zuo [19] applied a 3D Mask RCNN with some modifications to detect and segment both kidney and tumor at the same time in their data. U-Net was then used to further optimize the kidney and tumor segmentation results before they were output. Their RCNN network is able to propagate contextual information to higher resolution layers because of a significant adjustment they made In the up sampling component of the design, a high number of feature channels are included.. They were able to segment the kidney to a Dice Similarity Coefficient (DSC) of 96% and the tumor to a DSC of 43%.

In the paper by Krishnamoorthy Somasundaram [34], Both feature extraction and classification for The Kidney stones detection are handled by a combination of the Visual Geometry Group network 19 (VGN-19) and the Binary Support Vector Machine classifier (SVM) . Alexnet, GoogleNet, ResNet, and SqueezeNet were among the previous DL methodological advances they examined. When compared to Alexnet, GoogleNet, ResNet, and SqueezeNet, their Hybrid Deep VGN-19 and Binary SVM (HDVS) performed better in the early stage of recognizing Kidney stones across all metrics studied (Recall, precision, accuracy, specificity, and F1 score).

Extracting kidney tumor information from abdominal MR images is the focus of the work by Sitanaboina S L Parvathi and Harikiran Jonnadula [41], where they propose a semantic segmentation model for kidney tumors based on CU-Net and Mask R-CNN. The customized U-Net model was initially applied for kidney segmentation, and then the Mask R-CNN model was applied to highlight tumors from the segmented kidney pictures. They have created a model that can take advantage of both global location and context simultaneously. It outperforms competing segmentation algorithms despite using less training samples. When compared to other single-model entrants, Mask R-CNN consistently achieves better results across the board.

In their publication, S. Sudharson and Priyanka Kokil [29] discuss automatic classification of B-mode kidney ultrasound pictures using an ensemble of deep neural networks (DNNs) trained with transfer learning. They have employed numerous types including ResNet-101, pre-trained DNNs, MobileNet-v2 and ShuffleNet and have relied on the voting method (major) to make their combined and final predictions. Using the knowledge gained from pre-trained DNNs, they developed a technique to combine many DNNs into a single model, which can then be used as a more effective classification model.

Alnazer et al.(2020) [30] have evaluated all the methodologies and procedures for calculating and predicting CKD using various Diagnostic instruments including MRI, Ultrasound, and CT scans. Following a review of the available methods, they discussed the potential applications of artificial intelligence to renal segmentation. First, they have compiled a summary of the many medical imaging techniques used to detect CKD (chronic kidney disease) . Next, they demonstrated AI's capacity to guide renal function assessment beyond simple classification and into accurate disease detection. Researchers can utilize a variety of Segmentation approaches, such as DeepMedic, ScaleNet, VNet, and HighRes3dNet, etc., to improve CKD prediction and further the state of modern medicine.

The use of multi-resolution 3D V-net networks for segmenting kidney and renal cancers in CT scans is explored by Mu et al. (2019)[21]. In particular, their proposed VB-Net model is a hybrid of the two options they advise adopting. With a high rate of accuracy for kidney segmentation in disease diagnosis, their technique minimizes the overall burden on the GPU. Their suggested model can precisely identify the organ boundaries in CT scans and pinpoint their precise locations. They have used the CT scans from the KiTS Challenge Dataset for both training and testing purposes. After the data was prepared, VB-Net was utilized to successfully segment the

organs. While encouraging, there were still issues with irregularly dense tumors or cysts despite their efforts. Optimizing the post-processing algorithm to apply fixes was the solution they found.

Multiple diseases can be predicted using EHR data, as Ren et al.(2018)[23] have discussed. In most cases, an EHR will consist of two distinct modules. These include both concrete evidence and a written account. Predicting renal illness in hypertensive individuals is an underexplored area, and the paper’s authors propose studying it via a neural network model. In the beginning, we model the prediction problem as a simple task of yes/no classification. A hybrid NN can then be utilized to get a complete picture of HER. Both BiLSTM and Autoencoder networks have been used to predict kidney illness.

Dziekiewicz and Markiewicz(2018)[15] describe an innovative method for identifying kidneys in CT scans. After doing shape analysis on the object’s boundaries, they calculated the geometrical coefficients needed for ultimate detection. Approximately 84% of cases can be correctly classified as border detection of the kidney. Canny’s edge identification algorithm, which calculates geometric forms using a coefficient, is used in this investigation. The ability to detect more organs is yet another strength of their approach. Their technique paves the path for a speedy and reliable approach of diagnosing kidney neoplasia. In their study, they discussed renal algorithm recognition, brightness correction, image segmentation, and the derivation of geometric coefficients. They have employed 59 CT scans from a reliable source in their analysis.

For establishing segmentation of kidneys and renal tumors that is clear and unambiguous, manual delineation approaches are frequently time-consuming, prone to mistake, and need specialist expertise. They explained in their study [22] that a fully Convolutional Neural Network (CNN) that is aware of its bounds across the whole network is necessary for kidneys and kidney tumors effective semantic segmentation from 3D CT images of arterial phase abdominal. A segmentation network having encoder-decoder was taken into consideration for extracting tumor information by creation of a specialized network edge branch and edge-aware loss terms. The primary segmentation branch of our network is joined by a second border stream that analyzes feature maps at the border level. A 176x176x176 crop is used as the encoder’s input, and it is first sent through a 3x3x3 convolution with filters number of sixteen. Before being fused in the attention gated layer and transmitted via a ReLU activation function, each input is processed through a 3x3x3 convolutional layer with the same number of feature maps. In order to incorporate the highlighted boundary information into the mainstream-learned feature maps, the fusion module was developed. The last layer consists of a 1x1x1 convolutional layer accompanied by a channel-wise sigmoid activation function, and its input is the product of the outputs of the various branches, which have been multiplied together element-wise. Based on their own data split, they trained and assessed their model on the KiTS 2019 Kidney Tumor Segmentation challenge (as part of MICCAI) and for the kidney it had obtained 0.970 dice and tumor segmentation and for tumor segmentation it had 0.834 dice. MICCAI hosted the challenge.

In the work of [25], they said that the presence of kidney cancer at stage 1 is a neces-

sary prerequisite for therapy. Typically, accurate estimation of the stage of cancer is difficult for clinicians which results in improper treatment. DLNN (Deep Learning Neural Network) was used for kidney cancer predicting Stage 1 and also it boosts the accuracy of predicting higher stages also. Around 227 patients having kidney cancer (various stages) from the TCIA database had been trained, evaluated and validated. The 3D CT images were edited to exclude the kidney malignancies. Division of datasets into training (48%), test (42%), and validation (10%) sets. Within the TensorFlow platform, the deep learning network structure known as Inception V3 was utilized for this particular project. The Inception V3 Deep Learning Neural Network with Transfer Learning was trained with kidney cancer training photos that had their boundaries reduced. The Inception V3 neural network was pre-trained with the use of the ImageNet dataset, which contains more than one million photographs of natural scenes. In order to get an idea of how accurate the categorization was, the area under the ROC curve (AUC) was calculated. After some adjustments, the data that was supplied to the network after it had been trained to differentiate between the one thousand classes in ImageNet proved to be useful for classifying the various stages of kidney cancer as well. Carcinoma of the kidney The transfer learning approach involves retaining the knowledge learned while working through one issue and then applying that knowledge, after receiving more instruction, to a second problem. For the training data, they achieved an AUC of 0.97, for the validation data, 0.91, and for the test data, 0.90.

According to research published in [28], the 2019 Coronavirus Disease Outbreak (COVID-19) is the most infectious coronavirus ever documented (SARS-CoV-2). When applied to the identification of COVID-19, the use of AI methods, in particular deep learning, will be of assistance in the early discovery of this virus, which will, in turn, raise the likelihood of a speedy recovery for patients all over the world. As a result, this will relieve some of the stress on healthcare systems everywhere. Using CGAN (Conditional Generative Adversarial Nets) against traditional data augmentation methods is the model for detection of COVID-19. In this inquiry, in total five separate deep CNN-based models had been chosen for investigating the detection of patients having coronavirus infection utilizing chest CT radiography digital pictures. These models are VGGNet16, AlexNet, VGGNet19, GoogleNet, and ResNet50. The models are compared against each other to determine which one performs the best. When conventional data augmentations are used in tandem with CGAN, classification performance is enhanced across the board for the specified deep transfer models. The proposed architecture is made up of two significant parts: the first part is the data augmentation, which makes use of techniques of augmentation which is very much standard and CGAN, and the second part is the DTL model. With an accuracy of testing of 82.91%, sensitivity of 77.66%, and specificity of 87.6%, the results demonstrate that ResNet50 is the best deep learning model for detecting COVID-19 from a restricted chest CT dataset using conventional data augmentation. This conclusion was reached as a result of the findings that demonstrate that the best DL model is ResNet50.

The Research[4] said in the discipline of mathematical morphology, the watershed transform is the preferred technique for picture segmentation. In this article, numerous descriptions of the watershed transform and their associated consecutive

algorithms are described. Additionally, a variety of difficulties that can lead to misunderstandings in the literature are talked about. Differentiating definition, algorithm specification, and algorithm execution are all highlighted as being important. Several cases are shown to show how watershed transformations might vary depending on their concept and/or execution. The paper's second section provides a broad overview of methods for accelerating sequential watershed algorithms in a parallel setting. One possibility for a definition of a watershed that applies to the continuous situation is to use distance functions. One might arrive at a different definition of the term depending on the distance function that was utilized. It becomes problematic when we try to apply the aforesaid method to photos that are not lower complete. In these examples, there is no vacant interior on the plateaus. When the foregoing definitions are applied strictly, we find that the topographical distance between pixels located in the interior of a plateau is exactly zero. This necessitates establishing a further ordering relation between these types of pixels. Standard practice calls for finding the geodesic distance to the plateau's base. To put this into a formal framework, we first transform the image to a lower complete image, and then we apply the definitions from above. Plateaus, also known as zones of continuous gray value, can be a challenge to deal with when working with digital photographs since they can cover a significant amount of an image's surface area. These plateaus present an issue when extrapolating the concept of a continuous watershed positioned on topographical distances to pixel-based images. Differentiating between distributed memory and shared memory architectures, as well as other strategies for parallel implementation, was covered. Due to its sequential structure, the watershed approach via immersion is challenging to parallelize. Transforming this method into a components graph allows for a parallel implementation. Thanks to its distance-based concept, parallel implementations can take several forms. The majority of them employ either (ordered) queues, recursive raster scanning, a form of the union-find method, or some combination of the three, but the queues are the most frequent. The most essential thing that can be learned from this study is that there will always be a portion of the watershed transformation that is conducted on a global scale. As a consequence of this, only minimal benefits in performance can be anticipated when conducting this transformation in parallel.

In the work of [6], they said mathematical morphology, the watershed transform is a well-liked segmentation technique. Simply put, the watersheds are the lines that divide the "lakes" (technically termed catchment basins) that emerge when rain gradually falls on the ground, and the image may be thought of as a topographic relief in which the height of each point is directly connected to its gray level. With the watershed transform, the gradient of the original picture is used to determine where the catchment basin boundaries should be placed. Some of the drawbacks of watershed segmentation are over segmentation, sensitivity to noise, poor detection of significant areas with low contrast boundaries, poor detection of thin structures. They introduce an innovative tweak to the watershed transform that vastly enhances its performance by including functions based on past knowledge. In addition, they advocate for the use of a statistical atlas to introduce previously acquired wisdom into watershed segmentation. In particular, they demonstrate in excruciating detail how the recently disclosed method was utilized for knee cartilage segmentation as well as white matter/gray matter segmentation, two famously challenging applica-

tions for medical picture segmentation. When applied to individual photos, they will be paying special attention to the topographical distance-based methods. To do this, it is necessary to define the lower slope and the lower neighbors. In order to compute the watershed transform, it is required to construct a steepest slope connection between voxels, and here is where the lower slope comes into play. As an alternative to using the actual image data, the watershed transform for segmentation is typically computed using the absolute value of the gradient, which has high degrees through the illustration of objects. Gradient estimation in the middle of the pixels reduces the original quality of the picture and could cause problems when segmenting very tiny areas. One of its most significant drawbacks is the oversegmentation that frequently occurs after a change in the watershed. The homotopy of the function upon which the technique is employed is usually require adjusted in order to predetermine the number of areas produced by the watersheds approach as well as their estimated position. This is done in order to fulfill the goal of predetermining the amount of watersheds. Geodesic reconstruction is a mathematical morphological approach that is utilized in order to bring about the alteration that is desired. The function is changed by this operation to the point where the minima can be imposed on it by another function (the marker function). They begin by assuming that the entities in the photo had normal distributions, and afterwards they calculate the mean and variance of those distribution functions using a set of seed voxels that are individual to each class. This allows them to determine whether or not the assumption was correct. It is recurrently necessary to take exact calculations of the thickness and volume of a patient’s cartilage in order to acquire an accurate diagnosis and evaluate how well a patient is responding to treatment. This is done so that the physician can determine how well the patient is responding to treatment. In order to determine the accuracy and dependability of magnetic resonance imaging (MR) as a diagnostic tool for diseases affecting cartilage, a lot of studies have been carried out in recent years. Direct two-dimensional observations on individual slices are not feasible since their locations and alignments cannot be duplicated accurately in longitudinal examinations. This makes it impossible to replicate their positions. Because of this, it is not possible to use these slices to evaluate the changes that take place in cartilage over the duration of time. There was a 94.56% success rate in the white matter, compared to an 89.02% success rate in the gray matter.

In the work [38] it is presented that For the best possible care and to lessen the risk of developing serious kidney stone disease, early diagnosis and size measurement of renal calculi are crucial. Volumetric measures of kidney stones are more useful, according to previous research. Compared to linear measures , systems based on deep learning that use of abdominal non-contrast computed tomography (CT) images might be helpful by eliminating the requirement for manual stone volume detection, you may decrease workload. Here the researcher utilized a dataset consisting of 89 CTC scans without kidney stones and 91 CT colonography (CTC) images with kidney stones that had been manually tagged. Half of the data was utilized for training and testing in order to compare to earlier research. As an external validation set, 6185 CTC scans from a different institution’s patients were employed. These scans included patient-level labeling. Following segmentation of the kidneys using a 3D U-Net model, we performed territory, gradient-based anisotropic denoising, and thresholding. Then, a 13-layer CNN classifier was utilized to distinguish between

regions containing kidney stones and those that were false positives. The method improved upon previous work that obtained a sensitivity of 0.52 on a challenging test set of low-dose CT containing multiple small stones, achieving a sensitivity of 0.86 at 0.5 false positives each image. $R2 = 0.95$ indicates a strong relationship between machine and human estimations of stone volume. On an external validation set, the system for patient categorization acquired a value of 0.95 for the area under the receiver operating characteristic (sensitivity = 0.88, specificity = 0.91 at the Youden point). The presence of tiny atherosclerotic plaques in the renal sinus that resembled the appearance of kidney stones was a typical source of false positive results. These plaques were mistaken for kidney stones.

The paper [40] argues that the shortage of nephrologists around the world, the prevalence of kidney failure in the general population, and the advent of AI based systems with necessary development that could automatically identify kidney problems. The main purpose of the study is to amplify the state of the art in artificial intelligence by constructing a kidney illness diagnosis system using AI. The three main types of renal disorders studied were kidney stones, cysts, and malignancies. In all, 12,446 CT scans of the complete abdomen and urograms were gathered and annotated for the purpose of the study. Data from an exploratory examination of the collected pictures revealed that the images from each grouping shared a similar mean color distribution. We also constructed six machine learning models, three of which are based on state-of-the-art Vision transformers (the EANet, CCT, and Swin transformers), and the other three on established deep learning methods (VGG16, ResNet and Inception v3 that were modified in the final layers). Although the VGG16 and CCT models did exceptionally well, the swin transformer had the highest accuracy, at 99.30 percent. When comparing accuracy, recall, and the F1 score, the Swin transformer comes out on top and a training model which can be trained quickest. This study also revealed the inner workings of VGG16, ResNet50, and InceptionV3 in spotting the necessary anatomical anomalies. Their Swin transformer-based model, which has a higher degree of accuracy than the VGG16-based model, may be useful in the identification of kidney tumors, cysts, and stones, in our opinion.

According to the research published in [36], kidney stones are a frequent cause of visits to emergency rooms due to excruciating discomfort. Kidney stones can be diagnosed with a number of different imaging methods. Images like these require the expertise of trained professionals in order to be properly interpreted and diagnosed. Practical methods that can aid clinicians in their diagnosis include computer-aided diagnosis systems. This research proposes employing the deep learning (DL) technique, which has lately seen substantial advancement in the field of artificial intelligence, to automate the detection of kidney stones (stone present/not present) using (CT) coronal computed tomography images. Different cross-sectional CT pictures were taken for each subject, for a grand total of 1799 images. Using CT scans, our built automatic model detected kidney stones with a 96.82% accuracy rate. To our surprise, we found that our algorithm is able to reliably identify kidney stones of varying sizes. The improved performance of our DL model on a bigger dataset of 433 participants indicates it is suitable for clinical use. This research demonstrates that the currently fashionable DL approaches can be used to tackle additional dif-

difficult issues in urology. We have implemented f XResNet-50 for binary classification.

In the work [24] it is presented that Chronic kidney disease is a serious, lifelong illness that may be brought on by kidney damage or renal disease. Kidney cancer is one of the most deadly diseases now under study and has a significant impact on patient categorization and survival. Early detection and appropriate treatment may halt or postpone the progression of this chronic condition until the point at which kidney transplantation or dialysis are the only options left for preserving the patient's life. It has become urgently necessary to create automated technologies that can correctly detect kidney cancer subtypes in recent years. An Adaptive Hybridized Deep Convolutional Neural Network (AHDCNN) for the early diagnosis of Kidney illness has been suggested in this study to evaluate the capability of various deep learning approaches. The purpose of this study was to investigate the capabilities of these various approaches. The usefulness of categorization technologies is directly proportional to the functions performed by the data sets. An algorithm that is based on CNN has been created in order to accomplish the goal of improving the accuracy of classification systems by reducing the number of feature dimensions. Using these top-level features, it is possible to create a supervised tissue classifier that is capable of distinguishing between the two different types of tissue. The experimental process on the Internet of Medical Things platform (IoMT) has reached the conclusion, with the assistance of predictive analytics, that advancements in machine learning offer a promising framework for the recognition of intelligent solutions to demonstrate their predictive capability outside of the realm of kidney disease. This was determined after the process was conducted on a live patient.

Chapter 3

Proposed Model

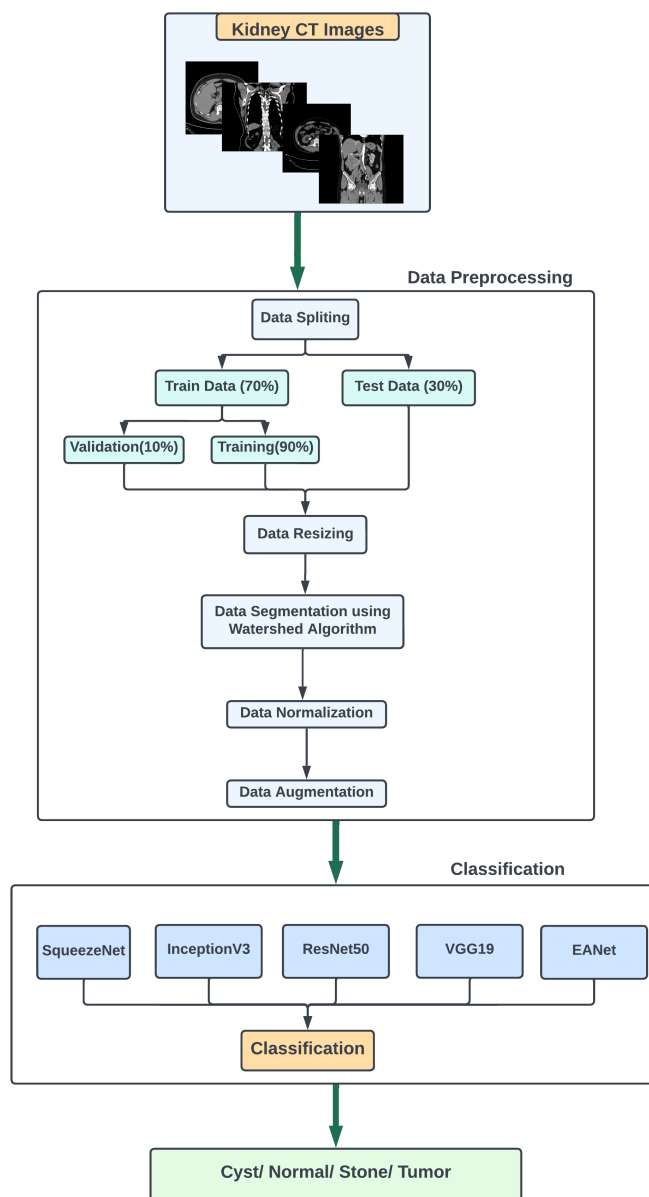


Figure 3.1: Workflow.

This section will illustrate how we have executed our classification experiment of Kidney Cyst, Normal, Stone or Tumor. The above diagram 3.1 provides all the steps that we have implemented to classify kidney disease using watershed segmentation algorithm and classification models. First, we have collected the specified data from the CT KIDNEY DATASET: “Normal-Cyst-Tumor and Stone” from Kaggle. The CT kidney images we have collected are in jpg format. After collecting the required data, we have split the data into two portions: Train(70%) and Test(30%). Then we have resized all the CT images into specific dimensions for feeding into the deep convolutional neural networks for classification tasks. After resizing, we have implemented the marker based watershed segmentation algorithm to segment the Kidney CT images. Moreover, then we have applied data normalization and data augmentation to our segmented images. Finally, we feed our segmented images into ResNet50, VGG19, InceptionV3, SqueezeNet and EAnet for classification in which the CT kidney images of test subjects are used to get the desired result, therefore weather the kidney has Cyst, Stone, Tumor or not.

3.1 Data Collection

The source for our dataset, the “CT KIDNEY DATASET: Normal-Cyst-Tumor and Stone” was from Kaggle [33]. After downloading the dataset from kaggle we found that there are in total 12,446 CT Kidney images and these images are categorized into 4 different labels(folders) and the images are in jpeg format. There are 3709 images of Stone class, 5077 images of Normal class, 1377 of Stone class and 2283 of Tumor Class. The proposition of each class of the dataset is shown in figure 3.2. For our research we have used all the 4 class images of a total 12,446 images.

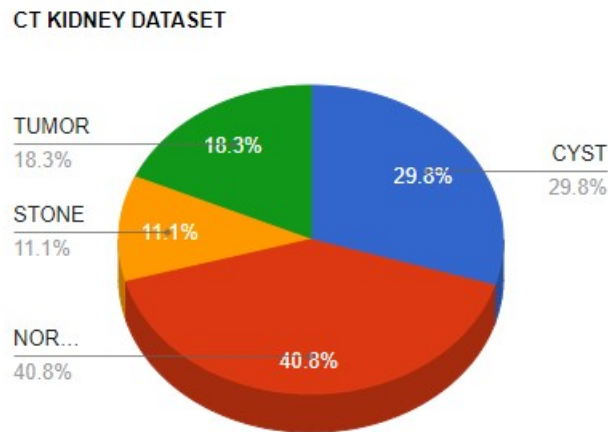


Figure 3.2: Distribution of CT KIDNEY DATASET.

3.2 Data Preprocessing

3.2.1 Data Spiting

The splitting of the data is the first step in the data preprocessing part. Since there are in total 12446 CT kidney images of jpeg format, first we have split all the images

into two sets: Train(70%) and Test(30%), further we have divided the train set into portions: Training(60%) and Validation(10%). As a result, the final distribution of training, validation and testing set is given in the table below:

	Cyst	Normal	Stone	Tumor	Tumor
Training	2342	3181	870	1447	7840
Validation	256	368	105	143	872
Testing	1111	1528	402	693	3734

Table 3.1: Dataset distribution after splitting.

3.2.2 Data Resizing

In our experiment, we have used all image type data and in our CT KIDNEY DATASET, all the images are of different shapes, moreover image resizing to a fixed size is an important prerequisite task before feeding them into the pre trained CNN models for classification, similarly neural networks receive input images of same dimensions. Since the images are of different dimensions, first we have find out the maximum, minimum, average and median dimensions of all the images, after that we have resized all the training, validation and testing sets using the OpenCV library of python into specific dimensions for inputs into the models.

3.2.3 Segmentation using Watershed Algorithm

The watershed transform, initially proposed by Digabel and Lantuejoul and later refined by Beucher and Lantuejoul [1], is the method of choice for image segmentation in gray scale mathematical morphology. Generally speaking, image segmentation is the act of separating visual objects from their backdrop, i.e., dividing the image into separate areas with each region being homogenous in terms of a certain feature, such as texture or grey value [3]. A way to segment data based on regions is the watershed transform. This method takes its cue from geographical concepts, specifically that of a submerged landscape or topographic relief, with watersheds delineating areas that draw various kinds of precipitation [2]. An alternate strategy is to picture the terrain submerged in a lake with holes cut out of the nearby minima. Beginning at these local minima, basins (also known as "catchment basins") will fill with water, and dams are constructed where water from several basins would otherwise meet. The procedure comes to an end when the water level reaches the highest point in the surrounding area. As a result, the landscape is divided into basins or areas known as watersheds that are separated by dams. Modeling this image segmentation process can be done in two ways: either by locating basins first and then watersheds by choosing to take a set complement, or by quantifying a complete separation of the image into basins, and then revealing the watersheds via boundary detection . Both of these methods are described further below. In our research, we adopt a strategy for dividing up watersheds that is based on the placement of markers. The workflow for the segmentation of watersheds by employing markers is depicted in figure 3.3.

It is up to us to decide which valley point pairs should be combined and which should not in the marker-based watershed algorithm. The "marker-based" method entails

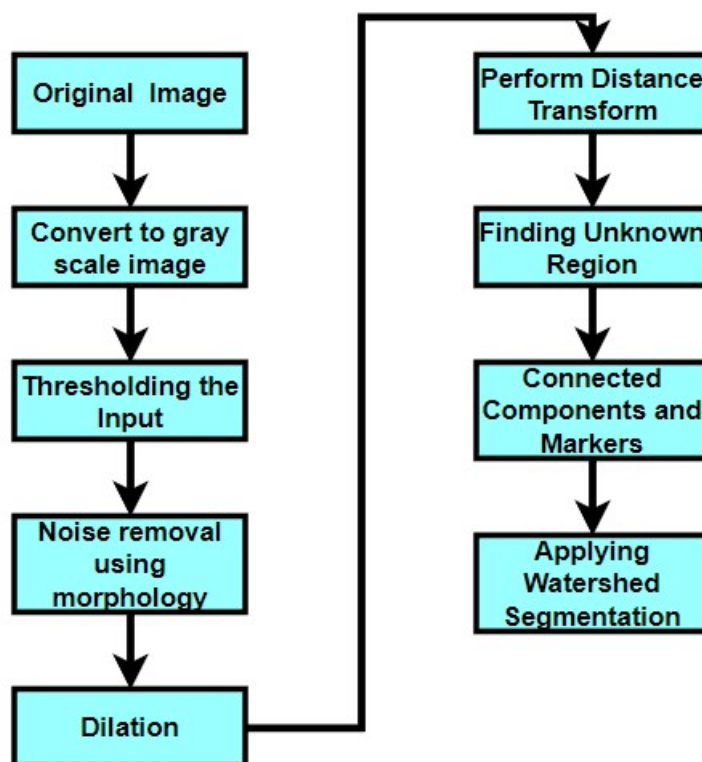


Figure 3.3: The flowchart of data segmentation using marker based Watershed segmentation.

assigning distinct names to the familiar objects we know to be in the foreground and background, respectively. Labeling the region that we know for a fact is either the foreground or the background with a different hue (or intensity) is a common visual trick. Then we put a 0 in the box encompassing the area about which we have no solid information. That will serve as a benchmark for us. First, we need to convert the image into a grayscale image. Secondly, we are thresholding the input. Performing Otsu's Binarization implies that if the pixel's value is above the threshold, it will be interpreted as a 1. Else, 0. Thirdly, we are removing the noise after specifying the background and foreground. The effect of the method is assumed to be bright on a dark background. The next step is dilatation. Images are convolved with a kernel of any size or shape, however squares and circles are the most common. Next, we were performing a distance transform. Distance transform modifies the gray level intensities of points in the foreground based on their respective distances from the nearest 0 value. Afterwards, we find the unknown region. Then, we assign a different hue to the region that we are certain is either the foreground or the background. Then we can just put a 0 in the space where there is doubt. Our point is here. The watershed algorithm is then used. At that point, our marker will be modified to reflect the newly assigned labels, and the borders of entities will be represented by the value of -1. Some of the pictures of watershed segmentation are shown in figure 3.4.

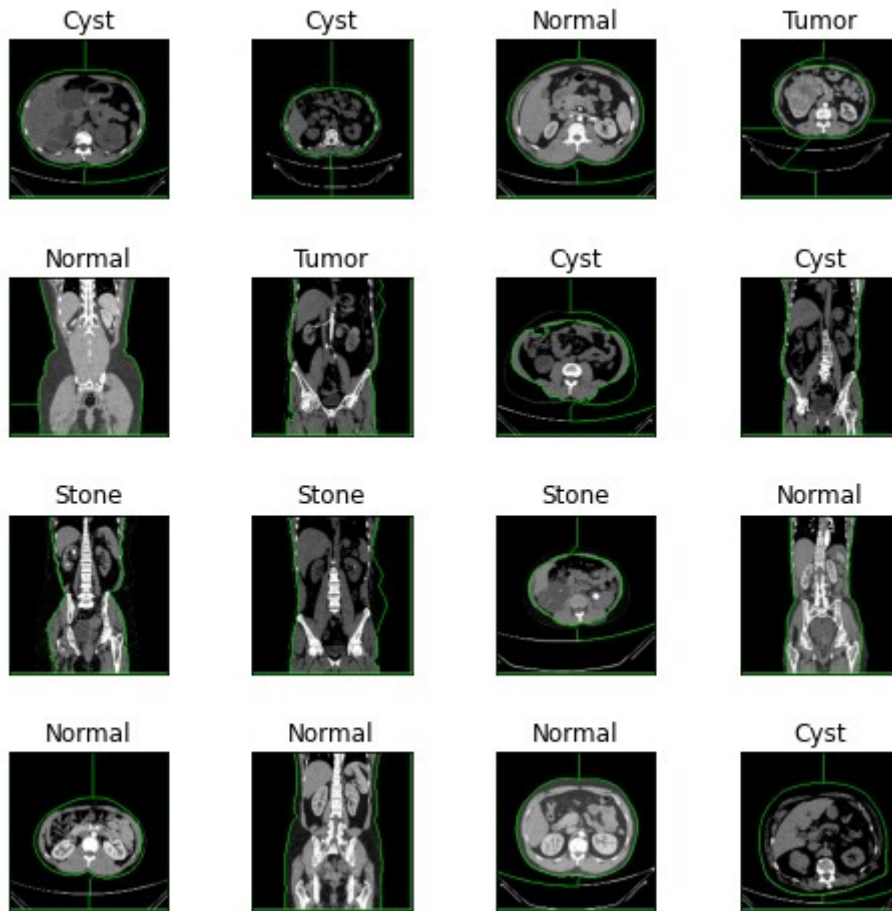


Figure 3.4: CT images after applying Watershed segmentation.

3.2.4 Data Normalization

Normalization of a database refers to the process of organizing data in a relational database according to a set of rules, or "normal forms". The database should have a higher level of data integrity after completing this method, and the number of redundant data should be reduced as much as possible. By doing this, we categorize the same kind of data in the same column. That will help to avoid data clustering. We normalized our training, validation and testing set by dividing the value of images pixels by 255. The grayscale value of an image is set to 255. After applying the normalization all the pixel values of every images are between (0,1).

3.2.5 Data Augmentation

Data augmentation is a process of artificially creating new data from existing data. After image resizing and segmentation we have augmented our training and validation set of our dataset using the Keras ImageDataGenerator Function. The parameters that we set when executing the augmentation process for training and validation sets, we set the zoom_range parameter by a scale factor of 0.05, we set the rotation_range parameter to 25, as a result the images will be rotated randomly by 25 degrees. The images were translated vertically or horizontally randomly by a scale factor of 0.05 of the image's height or width using the height_shift_range and

the `width_shift_range` parameters. Shear-based transformation is applied by setting the `shear_range` parameter to 0.05. The `horizontal_flip` parameter is kept true which causes half of the images to be randomly flipped horizontally. The `fill_mode` parameter is set 'nearest'. After applying augmentation, the figure 3.5 shows the augmented images of an image of Cyst, Normal, Stone and Tumor class.

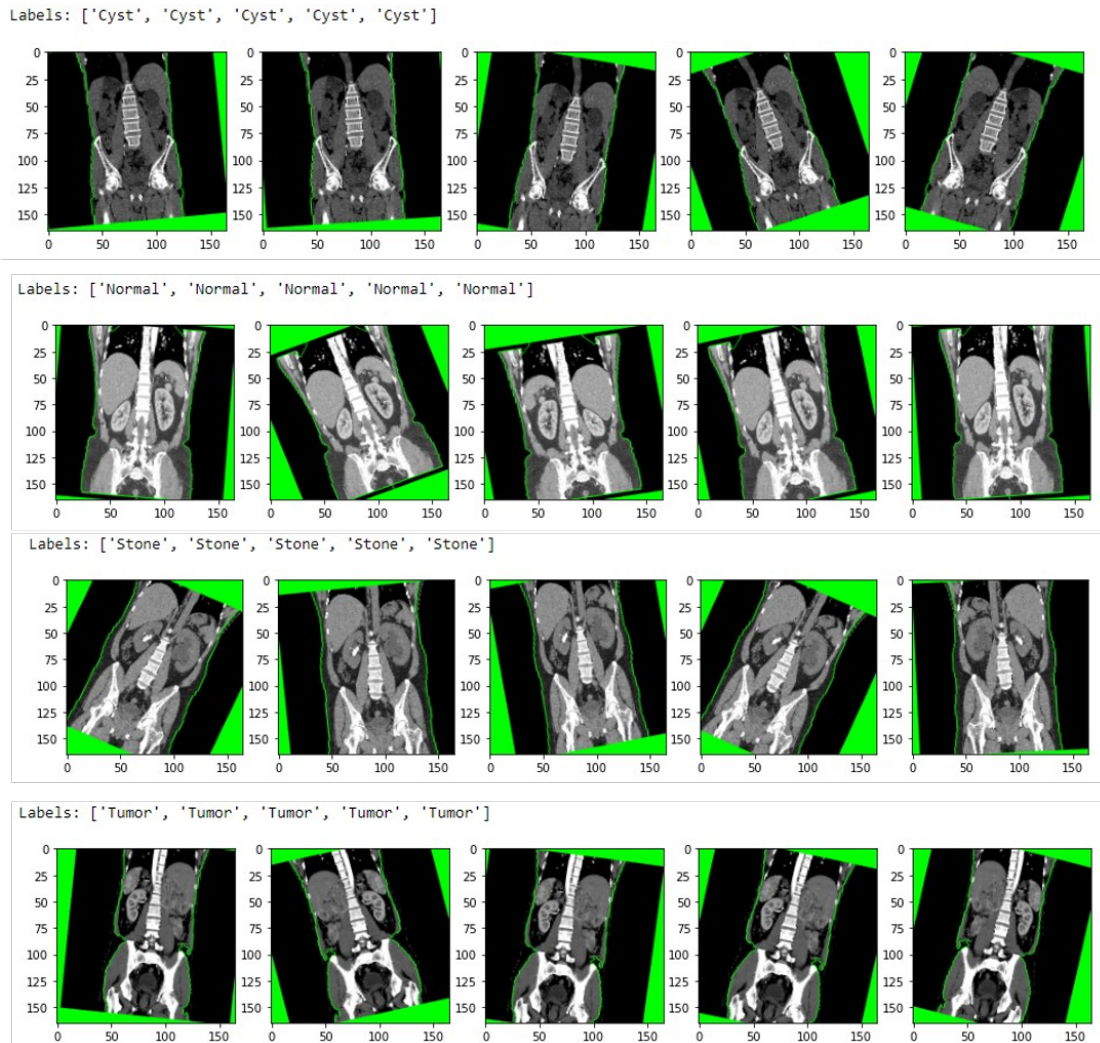


Figure 3.5: Augmented images of an image of each class.

3.3 Classification

After performing the data preprocessing, we have done our classification task and the models used for our classification are: ResNet50, VGG19, InceptionV3, SqueezeNet and EAnet.

3.4 Transfer Learning

Transfer Learning is a machine learning process in which a model is trained on a very large dataset like ImageNet which consists of around 14 million of images of 1000 groups, reused and trained on a relatively small dataset to execute the given task.

However, large datasets like ImageNet are very hard to find and those datasets are strongly required in training a model. To get around this issue, we employ transfer learning by reusing models that have already been trained on a big dataset (ImageNet) to complete the task at hand. The main idea behind transfer learning is that it is possible to use the general features learnt during training on a large dataset to deliver similar results when using seemingly distinct datasets [17]. One special feature of deep learning that makes it applicable in a variety of domain tasks with minimal datasets is the mobility of learnt generic features [17]. Few examples of such models are Inception, ResNet, VGG etc

In our paper, we have used the fixed feature extraction process for transfer learning. After a network has been pretrained on a large dataset, the final fully connected layer(s) are removed to create a fixed feature extractor, whereas the rest of the network, composed of pooling layers and convolutional layers (together referred to as the convolution base), remains unchanged and serves as a fixed feature extractor. The process of fixed feature extraction of transfer learning is shown in figure 3.6.

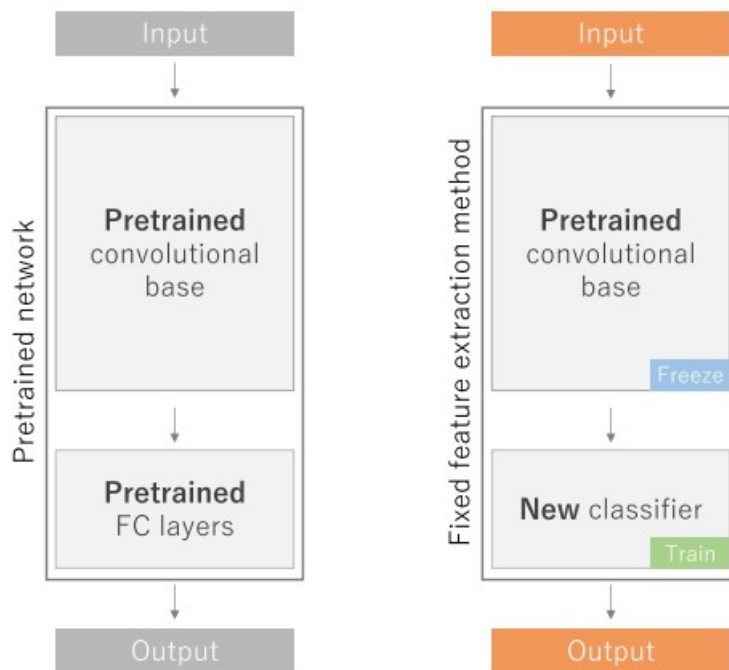


Figure 3.6: Fixed Feature Extraction of Transfer Learning.

The transfer learning models that we used in our classification task are already trained on ImageNet dataset which consists of around 14 million of images of 1000 groups. Fixed feature extraction was used to implement transfer learning models, which involved keeping all the layers frozen to preserve the learned weights, but discarding the final fully connected layer(s) and replacing them with a set of fully connected layers that were fine-tuned for a given classification problem. The transfer models we have used are ResNet50, VGG19, InceptionV3 and SqueezeNet.

3.4.1 VGG19

VGG abbreviates for Visual Geometry Group and it is a kind of deep convolutional neural network that was first introduced in 2014 [9]. VGG19 is a convolutional neural network which has been trained on more than 14 million pictures from the ImageNet dataset. The basic structure of VGG19 consists of 5 convolutional blocks and each convolution block has several convolutional layers with a max pooling layer [26]. The VGG19 model strictly uses 3×3 filters with pad of 1 and stride, along with 2×2 max pooling layers with stride 2 [10]. The VGG19 model consists of 19 trainable layers [26]. In our experiment, we have used all the layers of original VGG19 except the last 3 layers, we have used 16 convolution layers and 5 max pooling layers, the last 3 layers were replaced by a flattening layer, a dense layer, a dropout layer, a dense layer, a dropout layer and finally a dense layer. For the transfer learning of VGG19, we have frozen the first convolutional block of one max pooling and two convolutional layers and kept all the remaining layers as trainable and flipped the last 3 layers of original VGG19. In the modified VGG19, the activation function used in the last dense layer is softmax. The optimizer used in this model is Root Mean Squared Propagation ('RMSProp'), the learning rate was 0.00001 and categorical crossentropy was used for loss function. The total number of non-trainable and trainable parameters used in our custom VGG19 model are 2,325,568 and 24,517,636 respectively. The figure 3.7 shows the model architecture of VGG19. The layers of original VGG19:

- Convolution layer 1 with 64 filters
- Convolution layer 2 with 64 filters + Max pooling
- Convolution layer 3 with 128 filters
- Convolution layer 4 with 128 filters + Max pooling
- Convolution layer 5 with 256 filters
- Convolution layer 6 with 256 filters
- Convolution layer 7 with 256 filters
- Convolution layer 8 with 256 filters + Max pooling
- Convolution layer 9 with 512 filters
- Convolution layer 10 with 512 filters
- Convolution layer 11 with 512 filters
- Convolution layer 12 with 512 filters + Max pooling
- Convolution layer 13 with 512 filters
- Convolution layer 14 with 512 filters
- Convolution layer 15 with 512 filters
- Convolution layer 16 with 512 filters + Max pooling

- Fully connected layer with 4096 nodes(activation = “relu”)
- Fully connected layer with 4096 nodes(activation = “relu”)
- Output layer with Softmax activation with 1000 nodes

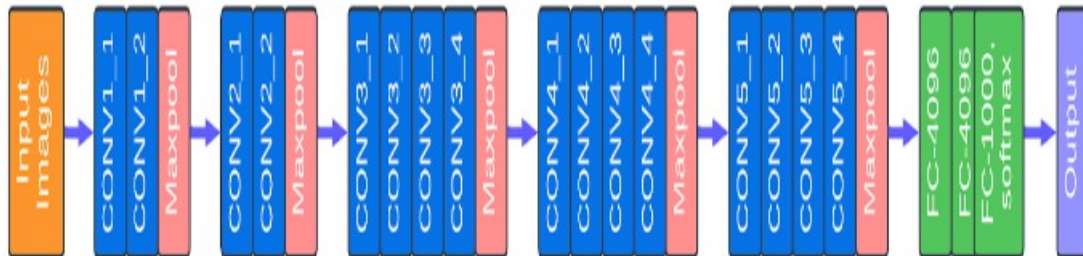


Figure 3.7: The Model Architecture of VGG19.

3.4.2 ResNet50

ResNet stands for Residual Network, which is a certain type of convolutional neural network that was first introduced in 2015 [10]. ResNet50 is a convolutional neural network which has been trained on more than 14 million pictures from the ImageNet dataset. This trained network can classify 1000 different object categories in photos, including keyboards, mice, pencils, and other animals [10]. The ResNet50 model is a 50 layers deep model. The size of the original ResNet50 model is around 25.6M parameters. In a residual network, the network learns from residuals, which are the characteristics that have been subtracted from the inputs of the layer from which the features are learnt [27]. Figure 3.7 shows the model architecture of ResNet50. The ResNet50 model architecture consists of:

- A convolution with a kernel size of 7x7 and 54 kernels with a stride of 2.
- A max pooling layer with a stride of 2.
- Convolution layers with 64 1x1 kernels, 64 3x3 kernels and 256 1x1 kernels. Then these layers are repeated 3 times.
- Convolution layers with 128 1x1 kernels, 128 3x3 kernels and 512 1x1 kernels. Then these layers are repeated 4 times.
- Convolution layers with 256 1x1 kernels, 256 3x3 kernels and 1024 1x1 kernels. Then these layers are repeated 6 times.
- Convolution layers with 512 1x1 kernels, 512 3x3 kernels and 2048 1x1 kernels. Then these layers are repeated 3 times.
- An average pooling layer and a flatten layer connected to a fully connected layer consisting of 1000 nodes, finishing with a softmax activation function.

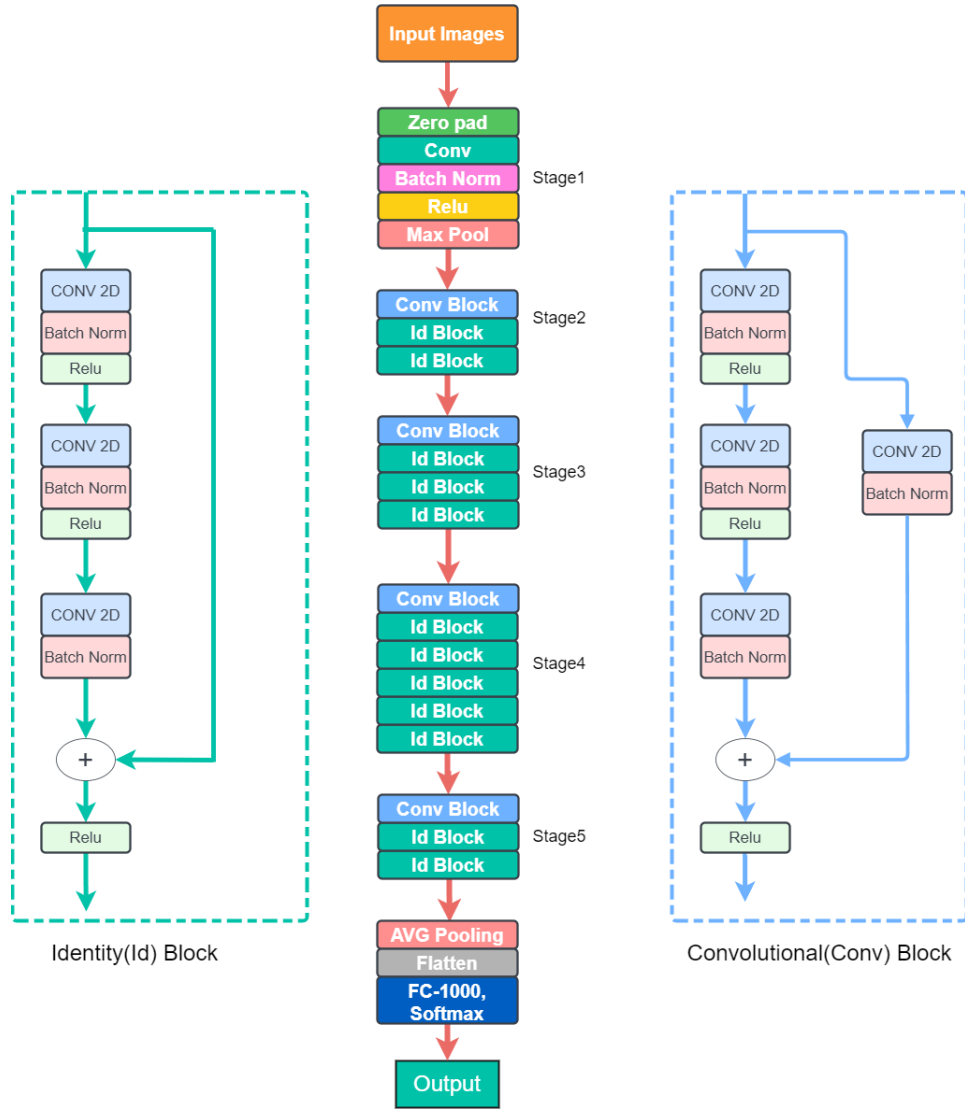


Figure 3.8: The Model Architecture of Resnet50.

In our experiment, first we load the pretrained ResNet50 model and utilize all the 50 layers except the last 3 layers. We removed the last 3 layers from the original ResNet50 model and added a flattening layer, a dense layer, a dropout layer, a dense layer, a dropout layer and finally a dense layer. Flatten layer is used to convert the multidimensional input vector into a single dimensional vector. The dropout rate is 30% in one and 20% in another dropout layer. ReLU activation function has been used in the first two dense layers with 512 and 256 nodes. Softmax has been used as an activation function in the last dense layer, because softmax is good for multiclass classification tasks. The optimizer used in this model is Root Mean Squared Propagation('RMSProp'), the learning rate was 0.001 and categorical crossentropy was used for loss function in our classification task. For the transfer learning part, since the last layer of pretrained ResNet50 model will act as a classifier for the previously trained on ImageNet dataset, that's why we remove the last 3 layers and add 6 layers which contains our customized fully connected layers(dense) as a result the final layer will be used as a classifier of Kidney CT images. Specifically, our customized ResNet50 model contains 8,611,712 non-trainable parameters and 52,857,604 train-

able parameters. Finally, skip connections are used in our custom ResNet50 model to evade the gradient vanishing problem and the performance abasement of Deep Neural Networks.

3.4.3 InceptionV3

This Convolutional neural network model employed in [12] for object detection and image analysis is a member of the Inception family of networks. For instance, InceptionV3 contains a batch normalization for layers at the sidehead, which implements label Smoothing, factorized 7x7 convolutions, and an auxiliary classifier to transfer label information along the network. It has a reduced error rate than its previous models, and has been optimized using a variety of strategies. InceptionV3 is more efficient than its previous version. The InceptionV3 model features a deeper convolutional neural network and is faster than the InceptionV1 and InceptionV2 models. It has less computational complexity and uses auxiliary Classifiers as regularizations. More than 14 million images from the ImageNet dataset were used to train InceptionV3. The figure 3.12 shows the model architecture of the original InceptionV3.

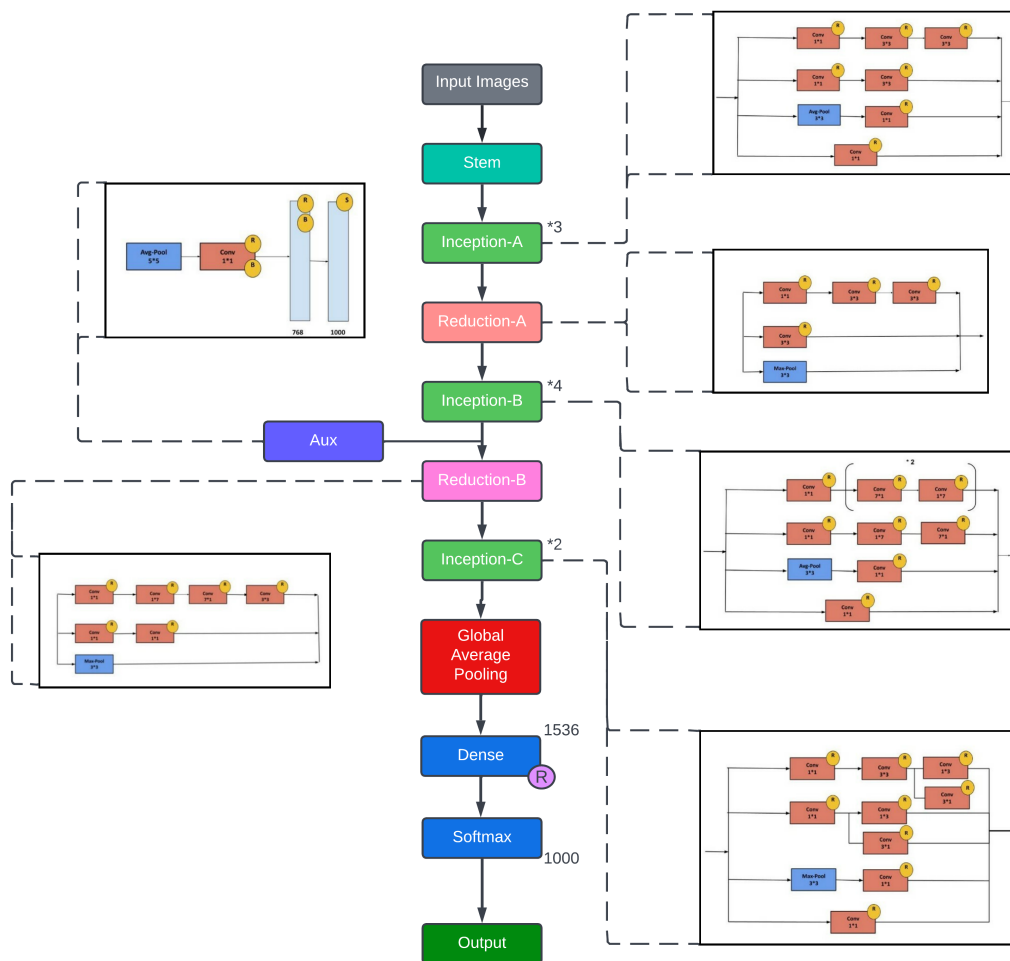


Figure 3.9: The Model Architecture of InceptionV3.

Similar to ResNet50 and VGG19, we have implemented transfer learning for the

InceptionV3 model by using all the layers of the original InceptionV3 except for the last 3 layers, we have flipped the last 3 layers with a flattening layer, a dense layer, a dropout layer, a dense layer, a dropout layer and finally a dense layer. In the modified InceptionV3, the dropout rate is 30% in one and 20% in another dropout layer. For the first two dense layers, we implement a 256-node and 128-node relu activation function, respectively. Softmax has been used as an activation function in the last fully connected layer(Dense). The optimizer used in this model is Stochastic Gradient Descent (SGD). In our classification challenge, we utilized a learning rate of 0.001 and a loss function of categorical crossentropy. The total number of trainable and non-trainable parameters used in our custom InceptionV3 model are 9,907,844 and 16,647,200 respectively.

3.4.4 Squeezenet

For gaining accuracy to be competitive, we can decrease the amount of parameters in CNN architecture. For instance, decreasing the filter size 3x3 to 1x1 will have 9x few parameters [11]. It is also important to decrease the number of input channels and also convolution layers have large activation maps. The SqueezeNet has 8 fire modules. One of the elements that goes into making up a Fire module is a convolution layer which is squeezed . (which only consists of 1x1 filters) feeding a convolution layer that has a combination of 1x1 and 3x3 filters. Pooling in late placements has been used. Squeeze and extend layer activations are subjected to ReLU (Nair Hinton, 2010). After fire9 module dropout has been used.Learning rate throughout the training has been decreased. A convolution layer that comprises several filter resolutions (such as 1x1 and 3x3) is not natively supported by the Caffe framework (Jia et al., 2014). A layer with 1x1 filters and a layer with 3x3 filters may be used in the implementation process to get past this problem. This can be accomplished by using an extended layer to combine two distinct convolution layers. The outputs of these layers are then concatenated together in the channel dimension to form the final product. This is the same as creating a single layer that has filters of both size 1x1 and 3x3 in it, according to the numerical equivalent. The spatial dimensions of the input are compressed into the channel dimension by the use of a flattening layer. Regularization strategies are a word used to refer to methods that either prevent models from becoming overly specific or enhance their ability to generalize. Dropout is one of the regularization methods that may be used in a variety of contexts (Srivastava et al., 2014). In its most basic form, dropout refers to the process by which, throughout each training cycle, a variable proportion of the dense layer's nodes are disabled. The dropout rate, which can range from 0 to 1, is used to explain this phenomenon and determines the proportion of nodes that go silent at any given moment. Dropout layers will only quiet nodes in a random fashion while you are training! During the stage that involves predictions, all of the nodes continue to be active (dropout is off). In order to ensure that all nodes get the opportunity to witness sufficient training data and acquire an understanding of its weights, the sample of nodes that are muted during training varies from one instance of training to the next. The convolutional layers of convolutional neural networks are frequently linked with the pooling layers of these networks. The pooling layer, in contrast to the convolutional layer, actually modifies the size of the picture and decreases it by a scaling factor. Pictures will effectively have a lower resolution as

a result of this. The idea behind this is that higher layers of the network should concentrate on higher-level characteristics of the picture. This was the logic behind it. The succeeding convolutional layer receives a more comprehensive 'view' of the original picture as a result of the addition of a pooling layer. Dense layers are referred to as completely linked layers due to the fact that all input neurons are considered by each output neuron in these levels. Therefore, the number of parameters that the network needs to learn is on the same order of magnitude as the number of input neurons multiplied by the number of hidden neurons. On the other hand, there are a great many distinct kinds of layers, each of which executes a unique set of calculations and receives a unique set of inputs. Convolutional layers and dropout layers are valuable not just in the context of picture data, but also in a wide variety of other sorts of (organized) data. In this episode, we will have a look at both of these types of layers. The figure 3.10 shows the model architecture of SqueezeNet.

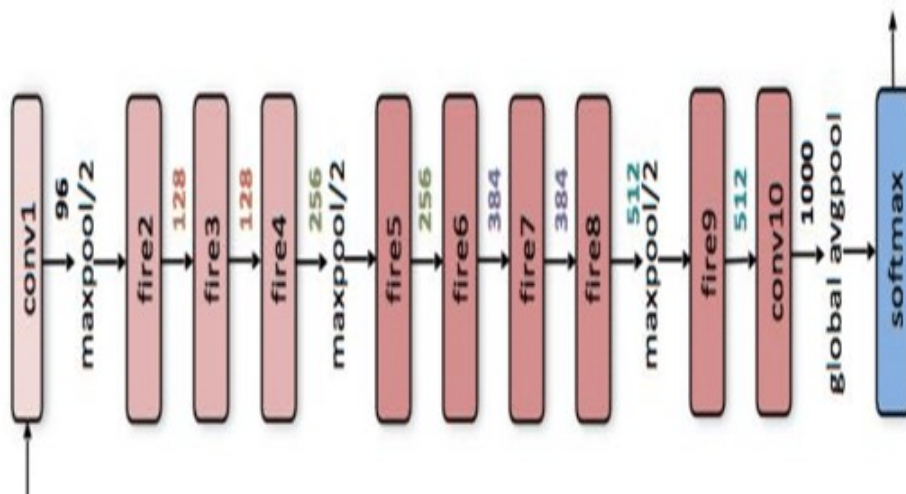


Figure 3.10: The Model Architecture of Squeezenet.

We have implemented transfer learning for the SqueezeNet model, just like we did for the other models, by using all of the layers from the original SqueezeNet model, with the exception of the last three layers, which we have flipped and replaced with a flattening layer, a dense layer, a dropout layer, a dense layer, a dropout layer, and finally a dense layer. This is similar to how we implemented transfer learning for the other models. In the modified SqueezeNet, the dropout rate is 30% in both dropout layers. The activation function known as relu with 512 nodes and 256 nodes is utilized in the first two dense layers of the network. The activation function known as softmax was utilized in the very last dense layer. Adam was chosen to serve as the optimizer for this particular model. The learning rate was set at 0.001, and categorical crossentropy was selected to serve as the loss function for the classification job. The total number of trainable and non-trainable parameters used in our custom SqueezeNet model are 21,752,708 and 336,320 respectively.

3.5 EAnet

By recognizing wide-range correlations, the self-attention procedure contributes to an improvement in performance across a wide variety of natural language processing

tasks[8][14][16][20]. As a result, several variations try to simulate self-attention with less computational overhead. Self-attention allows one to steer clear of the prospect of correlations with other samples in favor of concentrating on the self-affinities that exist between different locations within a single sample. It is clear that adding correlations between several samples can assist to improve the feature representation. In the semantic segmentation job, for example, characteristics from the same category that are spread over multiple samples should be handled consistently. A similar finding also holds true for picture classification and a number of other visual tasks. Focus from the outside world functions differently. They start with the attention maps, which is constructed by computing the affinities between the self-query vectors and an external learnable key memory. After that, they go on to the next step. In the following step, the researchers multiply this attention map by an additional external learnable value memory in order to construct a more refined feature map [32]. Since the two memories are often built with linear layers in reality, they lend themselves well to end-to-end optimization via back-propagation. They are not reliant on the specific samples and are shared over the whole dataset. This adds to a substantial regularization function and improves the attention mechanism’s capacity to generalize, since both of these benefits are a result of the fact that they are shared. Since the processing cost is proportional to the amount of input elements, paying attention to the outside world is an undemanding tactic to implement. This is as a result of the fact that the amount of elements included inside the memory is a great deal less than the number of components contained within the input feature. The objective of the external memory is to learn the properties of the complete dataset that are the most helpful and discriminative while simultaneously removing any information that is not relevant. One instance of the Vision Transformer based EAnet is shown in figure 3.11. The succession of picture patches is utilized as the

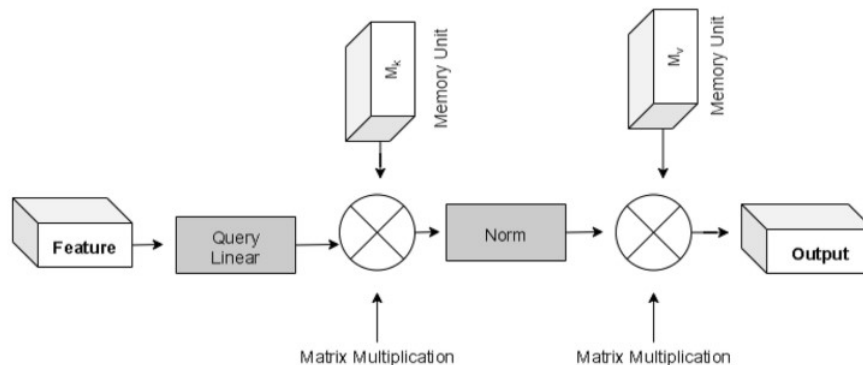


Figure 3.11: EAnet’s External Attention.

input for the multiple transformer block, which makes use of the multihead attention layer as the method for its own self-attention. The result of the transformer blocks is a tensor that contains the batch size, the number of patches, and the projection dim value. After that, you may send this tensor on to the classifier head, which uses softmax to determine the probabilities associated with each class. Dropping patches that include redundant and pointless information is the goal of EAnet, which will ultimately lead to improvements in both performance and the efficiency with which computations are carried out. The implementation of external attention utilizes two layers of normalization and two layers of cascading linear processing. The figure

3.11 shows the external attention of EAnet. EAnet20 makes use of external attention, which is built on two external memories, M_k and M_v , that are relatively tiny, learnable, and shareable [32]. The EAnet algorithm calculates attention between the input pixels and the external memory unit through:

$$A = \text{Norm}(FM_k^T) \quad (3.1)$$

Finally, input features are updated from M_v by the similarities in Attention A.

$$F_{out} = AM_v \quad (3.2)$$

Chapter 4

Result and Analysis

4.1 Experimental Evaluation

We used several different standard performance measures, such as classification accuracy, precision, recall, f-score, and receiver operating characteristic (ROC) curve, where it is summarized by area under the curve (AUC), log-loss, and confusion matrix, in order to validate the performance of each model. These performance measures include:

4.1.1 Accuracy

Accuracy is a way to measure how well a classification model works, and it is usually given as a percentage. The system functions by comparing the number of samples that were correctly categorized to the total number of samples that were put through the test. The following are the four possible outcomes:

- True Positive (TP): When predictions and results match.
- True Negative (TN): When both predictions and results are negative.
- False Positive (FP): when the predicted is positive but expected results are negative.
- False Negative (FN): when the predicted is negative but expected results are positive.

As such the formula used to find Accuracy is:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \times 100\% \quad (4.1)$$

4.1.2 Loss

After each cycle of optimization, the loss function is a calculation that determines how ineffectively or how effectively behaved a model is. To calculate this, we don't use percentages but rather add up all of these kinds of mistakes over all of the samples in our training and validation sets. In order to be more accurate, classifiers must minimize the amount of log-loss error they produce, ideally to zero.

4.1.3 Sensitivity, Precision and F1 Score

Sensitivity or Recall: Recall, which is also referred to as sensitivity, is a true positive rate that is calculated as the ratio of true positives (TP) to the total number of true positives (TP) and false negatives (FN):

$$Recall = \frac{TP}{TP + FN} \quad (4.2)$$

Precision: Precision, also known as positive predictive value, is an performance measure that was utilized in this work The following formula can be used to compute precision, which is defined as the ratio of actual positive cases to the total number of anticipated positive cases:

$$Precision = \frac{TP}{TP + FP} \quad (4.3)$$

F1-Score: F1 score is the harmonic mean of precision and recall, and it is crucial for asymmetrical class distributions. Since F1 uses the harmonic mean of precision and recall, it can be used to cancel out any inherent bias in the data. Correctly identifying real threats and reducing false alarms, as indicated by a low F1 score, is a sign of a good system. It can be calculated by this formula:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4.4)$$

4.1.4 ROC Curve and AUC

ROC curve is a probability curve used to evaluate a model's classification performance at different thresholds. The y-axis represents the sensitivity or true positive rate (TPR), while the x-axis represents the 1-specificity or false positive rate (FPR). The area under the ROC curve (AUC) is a measure of the probability or ROC used to assess the classifier's ability to distinguish between them. The AUC is a simple measure of how well a model predicts or separates true positives and true negatives; a high value indicates success in doing so, while a low value indicates that the model is incorrectly labeling most of the true positives as negatives and most of the true negatives as positives.

4.2 Implementation

As a deep neural network-based work that needs a high configuration PC to implement, we have used Nvidia RTX 2060 GPU, 24 GB ram, Ryzen 5 3600x with six core and 3.8 GHZ clock speed. Python version 3.9.7, Tensorflow version 2.9.1 and Keras version 2.7.0.

In order to save time, we decreased the images to 165x165 from their original resolution. After resizing the images we have segmented our training, validation and testing images using watershed algorithm. Then, we passed the segmented kidney data to the classification models (EANet, ResNet50, SqueezeNet, VGG19 and InceptionV3) for classification. Using five different CNN architectures, we have been able to get some results and analyze them further.

4.3 Analysis of Results

4.3.1 ResNet50

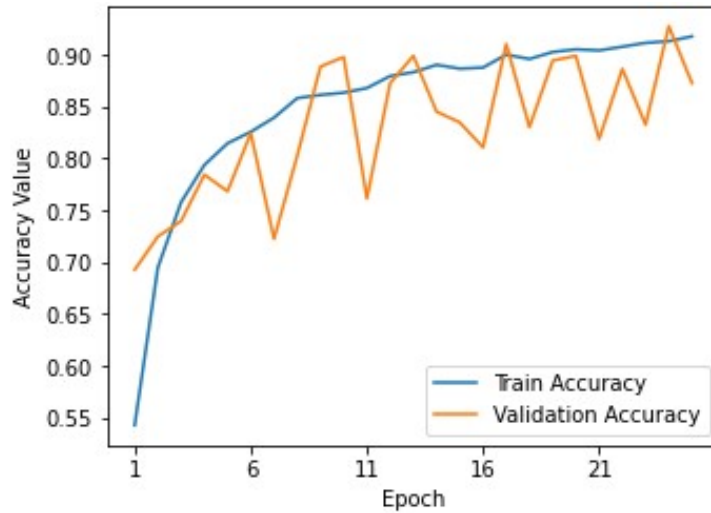


Figure 4.1: Train and Validation Accuracy graph of ResNet50.

From the figure we can observe that the train and validation accuracy was 55% and 70%. Then gradually both train accuracy and validation accuracy increased. However around 7 epochs validation accuracy began to fluctuate and decrease but train accuracy was increasing gradually. Until the end the validation accuracy fluctuated aggressively and ended by achieving validation accuracy at 87.27% and train accuracy at 88.34%.

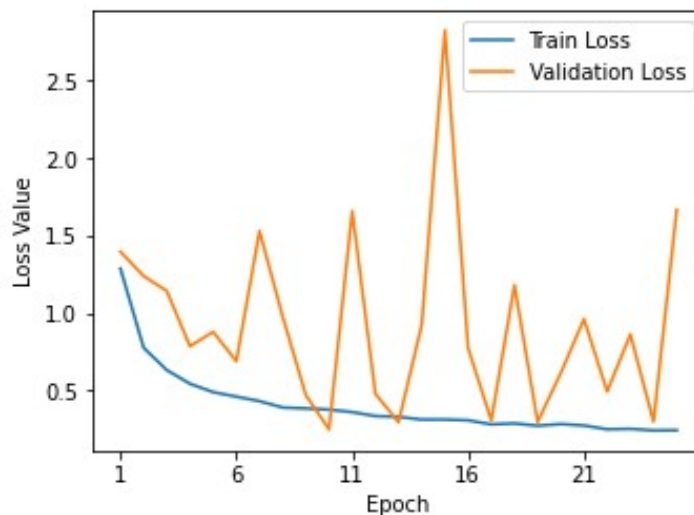


Figure 4.2: Train and Validation Loss of ResNet50.

We can also see from the train and validation loss graph that at the beginning of the graph train loss was around 1.37 and validation loss was 1.41. We can also see major fluctuation in validation loss while the train loss was stable and decreasing linearly. We can see some major validation loss around 11 epoch to 15 epoch.

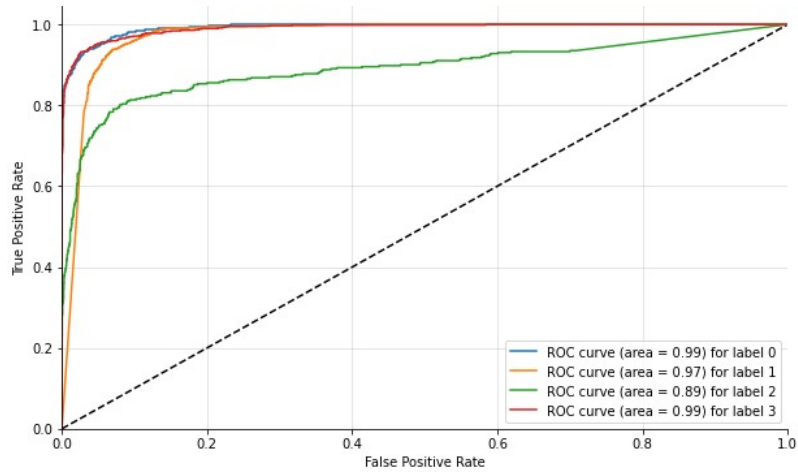


Figure 4.3: ROC Curve of ResNet50.

In the ROC curve graph we can see the performance of the ResNet50 model's sensitivity (or TPR) and specificity (1 - FPR). We can see that label 0 (cyst), label 1(normal) and label 3 (tumor) had good roc scores within 0.2 FPR which indicates that ResNet performed well for cyst ,normal ,tumor. However, it didn't do as good as others in terms of label 2 (stone).It took almost 1.00 FPR to gain similar performance for stone.

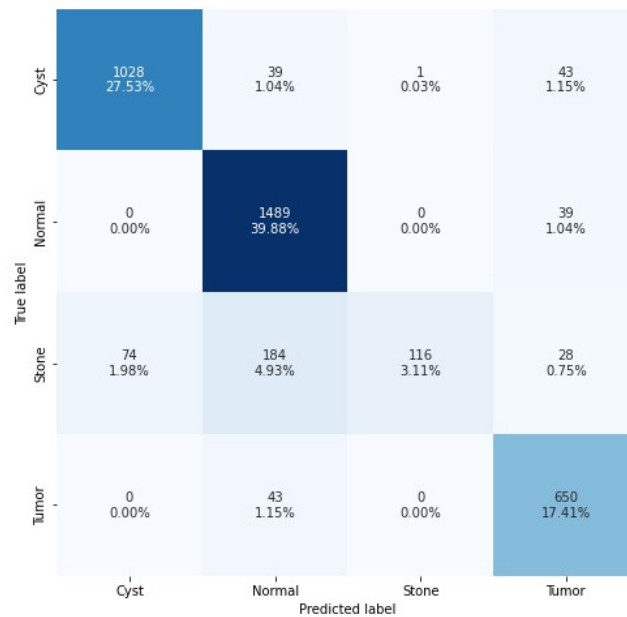


Figure 4.4: Confusion Matrix of ResNet50.

ResNet50 made a total of 3734 predictions and out of these 3734 predictions, 1028 (27.53%) right prediction are made from cyst kidney image class 1489 (39.88%) right prediction from the class which has normal category kidney images, 116 (3.11%) right prediction are made from stone kidney image class and lastly 650 (17.41%) right prediction are made from tumor kidney image class. The overall accuracy of this model was 87.9

4.3.2 VGG19

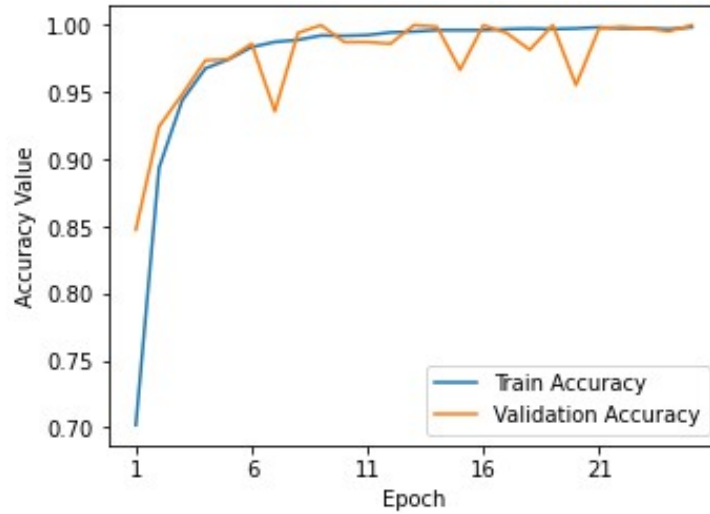


Figure 4.5: Train and Validation Accuracy graph of VGG19.

The accuracy graph for VGG19 shows that at first, the train accuracy was much lower, around 70%, while the validation accuracy was much higher, at 85%. When compared to other models, VGG 19's validation accuracy is substantially more stable. Initially, about 6 epoch, the train's accuracy grows aggressively, before stabilizing at a value of approximately 99%. But the accuracy of validation does drop a bit around 7 epoch, 15 epoch, and 20 epoch. But in the last part, the accuracy of the train hit 99.97% and the accuracy of the validation hit 100%.

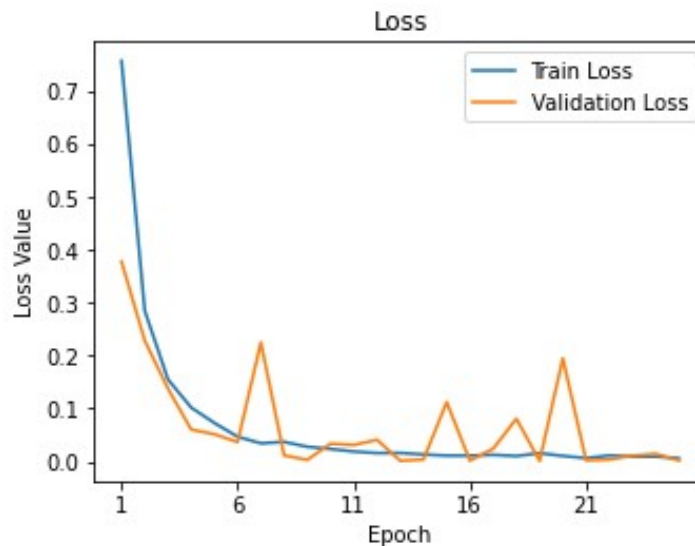


Figure 4.6: Train and Validation Loss of VGG19.

From Figure 4.6 loss graph we can observe that in the beginning, the train loss was greater than 0.7, while the validation loss was somewhere around 0.4. Nevertheless, in less than six epoch, both values will drop to below 0.1. In later epoch we can see some fluctuation in validation loss but after 21 epoch both validation loss and train loss becomes almost 0 and keep this value until the end.

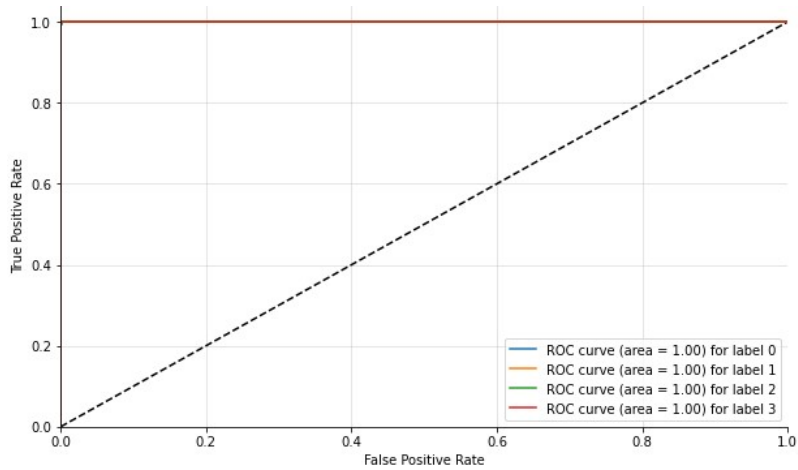


Figure 4.7: ROC Curve of VGG19.

The ROC curve of VGG19 reveals that this model performed exceptionally well on all four different kinds of kidneys, as was expected. It is clear that the area under the curve for each of the four labels is equal to 1.0.

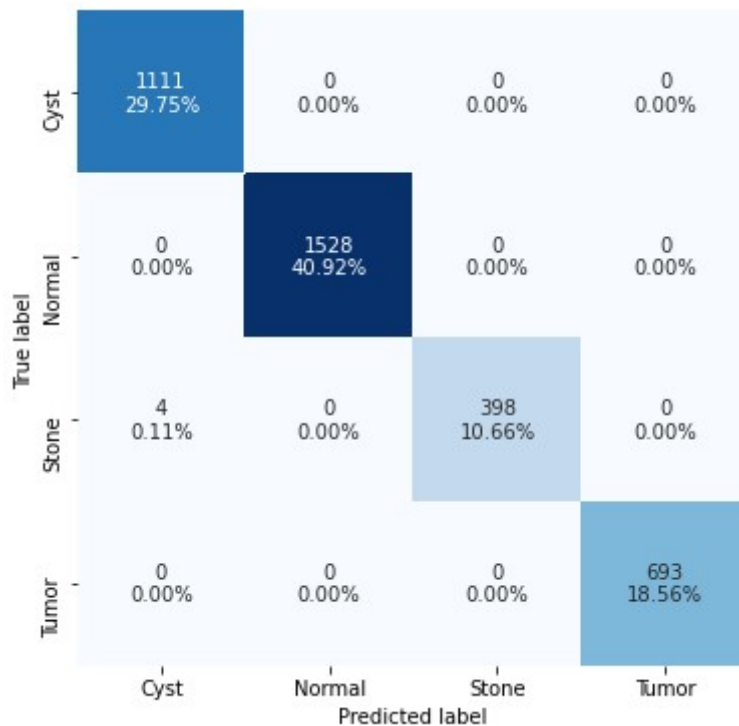


Figure 4.8: Confusion Martrix of VGG19.

VGG19 made a total of 3734 predictions and out of these 3734 predictions, 1111(29.75%) right prediction are made from cyst kidney image class, 1528 (40.92%) right prediction from the class which has normal category kidney images, 398 (10.66%) right prediction are made from stone kidney image class and lastly 693 (18.56%) right prediction are made from tumor kidney image class. The overall accuracy of this model was 99.89%.

4.3.3 InceptionV3

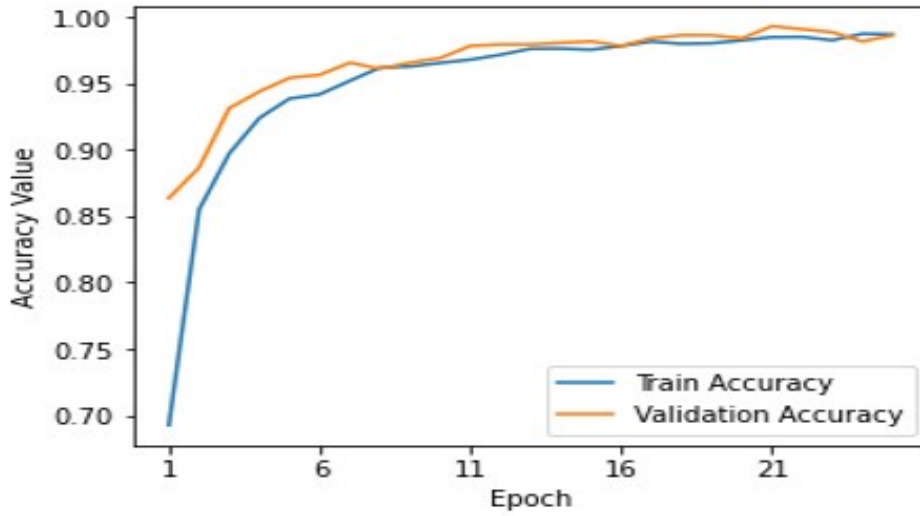


Figure 4.9: Train and Validation Accuracy graph of InceptionV3.

This figure 4.9 that the start train and validation accuracy of InceptionV3 was 70% and 86%. Then gradually both train accuracy and validation accuracy increased. However, in this model validation accuracy did not fluctuate too much and train accuracy and validation both increased gradually. In the end the both train and validation accuracy was stable by validation accuracy at 98.62% and train accuracy at 99.32%.

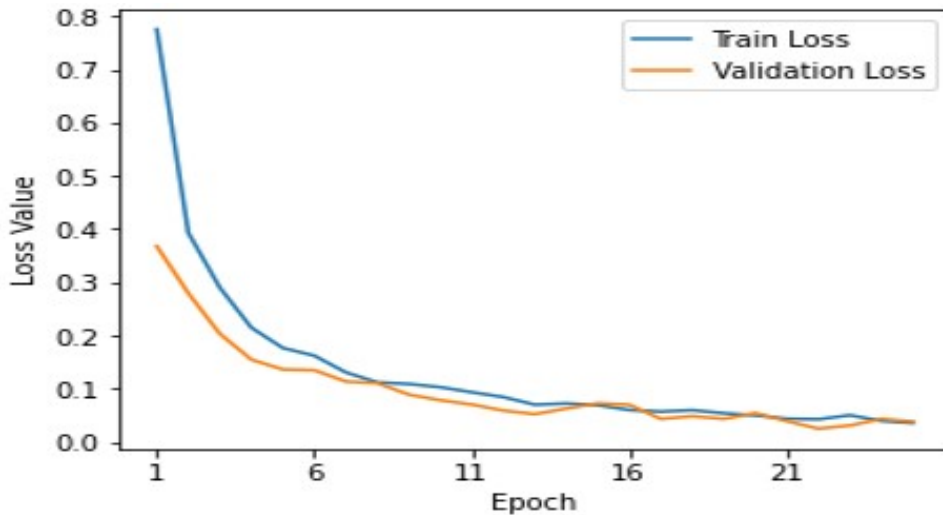


Figure 4.10: Train and Validation Loss of InceptionV3.

We can also see from the train and validation loss graph of InceptionV3 that at the beginning of the graph train loss was around 0.78 and validation loss was 0.38. We can also see validation loss and the train loss was stable and decreasing linearly. There was no major fluctuation in validation loss. At the end both train and validation loss was under 0.1.

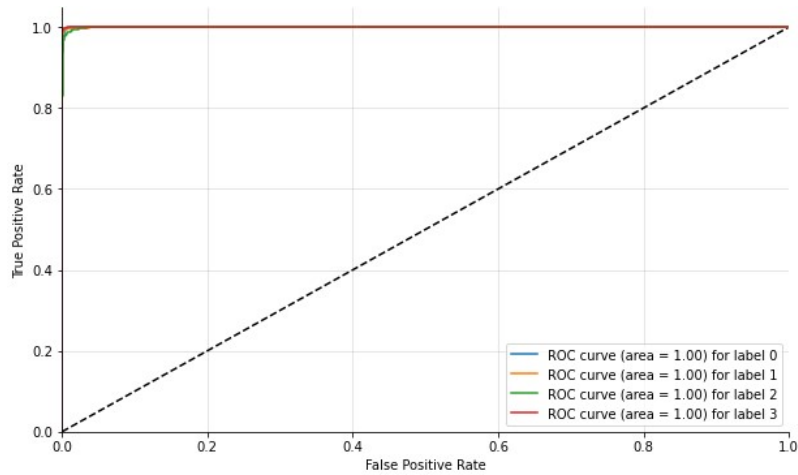


Figure 4.11: ROC Curve of InceptionV3.

From the ROC curve graph we can see that the InceptionV3 model performed very well in terms of sensitivity (or TPR) and specificity (1 - FPR). We can see that the area for label 1(cyst), label 2(normal) ,label 3(stone) , label 4(tumor) were 1.00 for TPR and FPR are almost zero. The one vs one and one vs rest roc auc score were 0.99.

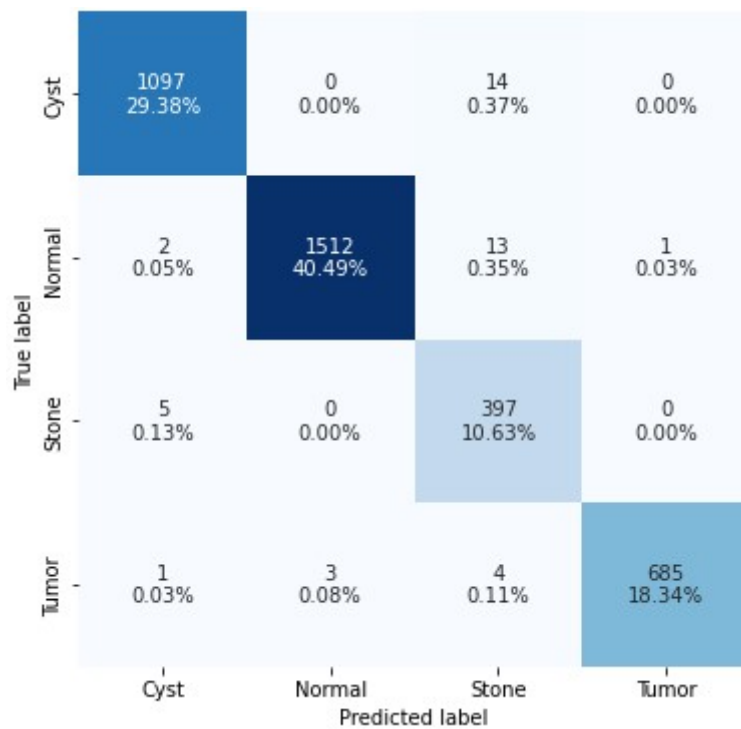


Figure 4.12: Confusion Martrix of InceptionV3.

InceptionV3 made a total of 3734 predictions and out of these 3734 predictions, 1097 (29.38%) right prediction are made from cyst kidney image class, 1512 (40.49%) right prediction from normal category kidney images, 397(10.63%) right prediction are made from stone kidney image class and lastly 685 (18.34%) right prediction are made from tumor kidney image class. The overall accuracy of this model was 98.8%.

4.3.4 EAnet

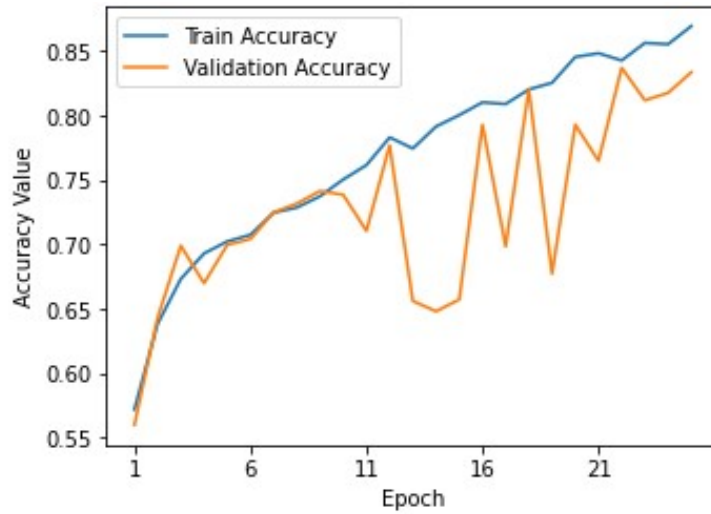


Figure 4.13: Train and Validation Accuracy graph of EAnet.

Figure 4.13 demonstrates that EAnet had an accuracy of roughly 56% both training and validation at the beginning. The train accuracy and validation accuracy were then gradually increased. Around 11 epochs, however, validation accuracy started to fluctuate and decline, whereas train accuracy was constantly rising. The lowest level of validation, roughly 65%, occurred during epoch 14. After that point, there was a major fluctuation in validation accuracy while training accuracy remained constant. However, in the end, validation accuracy remained consistent at around 83%, while train accuracy remained above 87%.

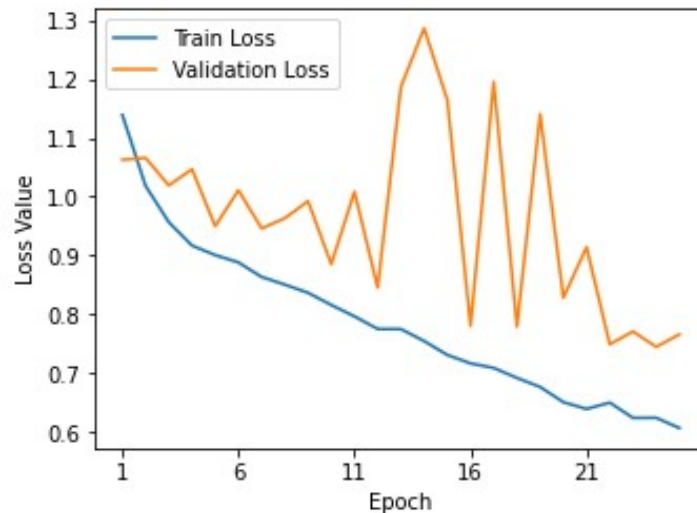


Figure 4.14: Train and Validation Loss of EAnet.

At the beginning of the train and validation loss graphs (figure 4.2), the train loss was approximately 1.15 and the validation loss was 1.08. Significant shifts in validation loss were visible, in contrast to the steady decline in train loss. Around 11 epoch to 21 epoch, there were significant ups and downs of validation loss that can be observed. however , in the end, the validation loss was around 0.8 and train loss at 0.6.

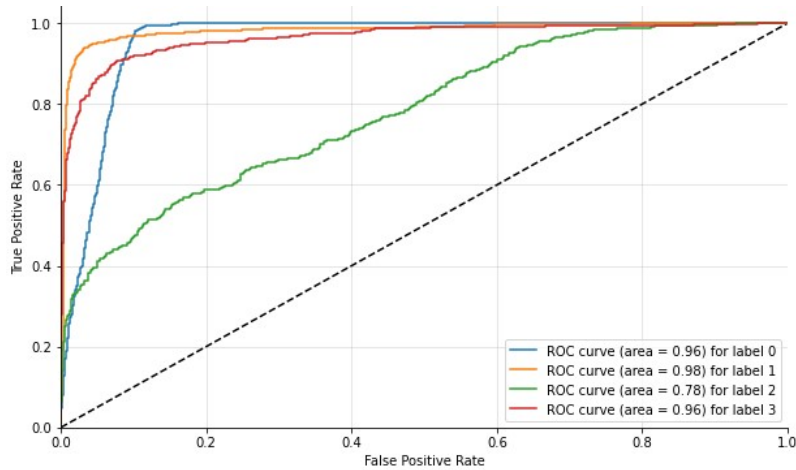


Figure 4.15: ROC Curve of EAnet.

In the ROC curve graph we can see the performance of the EAnet model's sensitivity (or TPR) and specificity ($1 - \text{FPR}$). It is clear that EAnet did a fantastic job with cyst, normal, and tumor labels because each one achieved a roc score of 0.8 TPR within 0.2 FPR. But in terms of label 2 (stone), it didn't do as well as others. Stone required 0.7 FPR in order to achieve the same level of performance as other labels.

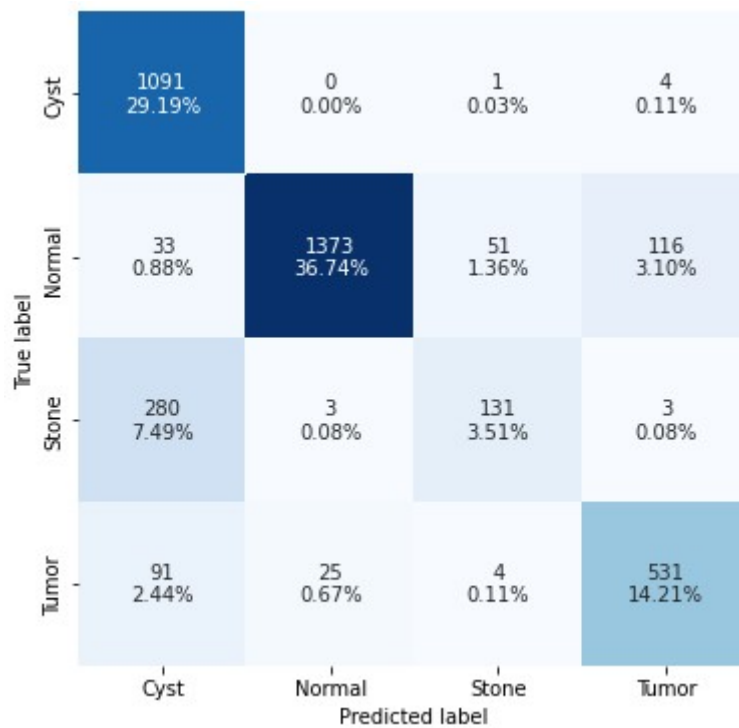


Figure 4.16: Confusion Matrix of EAnet.

The figure 4.16 shows the confusion matrix of EAnet, 1091 (29.19%) right prediction are made from cyst kidney image class, 1373 (36.74%) right prediction from the class which has normal category kidney images, 131 (3.51%) right prediction are made from stone kidney image class and lastly 531 (14.21%) right prediction are made from tumor kidney image class. The overall accuracy of this model was 83.6%.

4.3.5 SqueezeNet

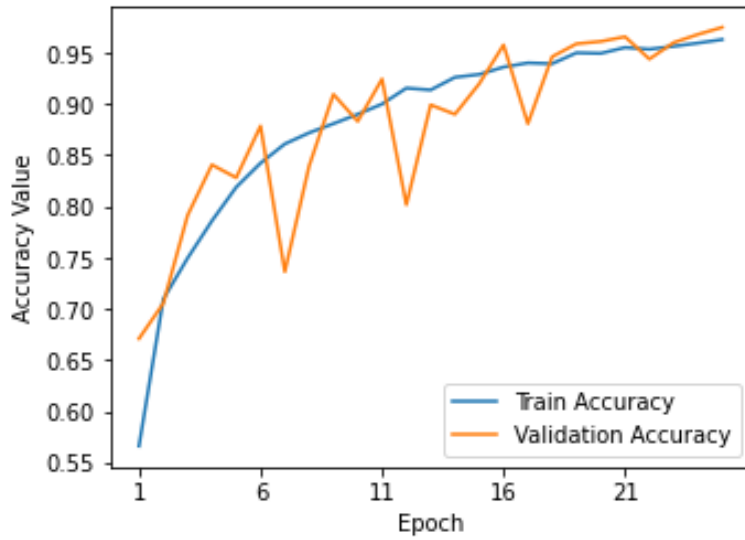


Figure 4.17: Train and Validation Accuracy graph of SqueezeNet.

The figure of the SqueezeNet model of the train and validation graph can be seen that at the beginning, the train accuracy was approximately 0.65, while the validation accuracy was approximately 0.55. Even though both of them gradually rose, the validation accuracy was quite inconsistent throughout the process. In the end both train and validation accuracy was around 0.95.

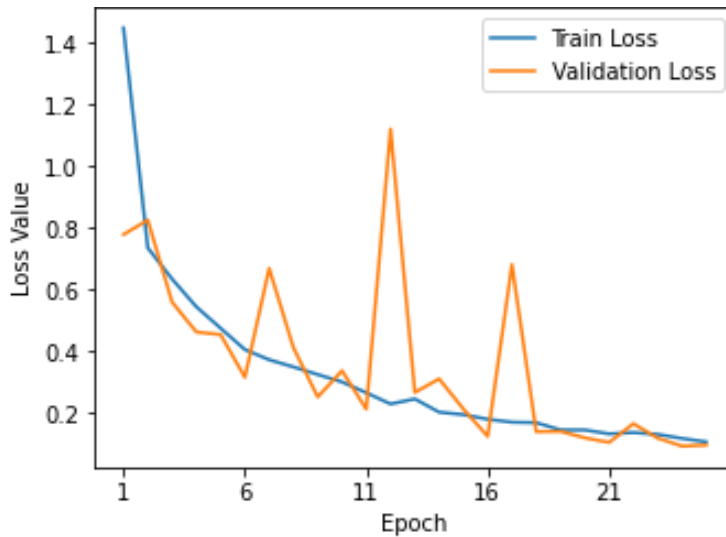


Figure 4.18: Train and Validation Loss of SqueezeNet.

We can observe from the graph of the train validation loss of SqueezeNet that at the starting point, the train loss was approximately 1.4, and the validation loss was approximately 0.8. Both of them went down to a lower level over time. Around epoch 12 and epoch 17, we see two significant fluctuations in the amount of validation loss. However, in the end they had finally reached a state of stability almost close to 0.1.

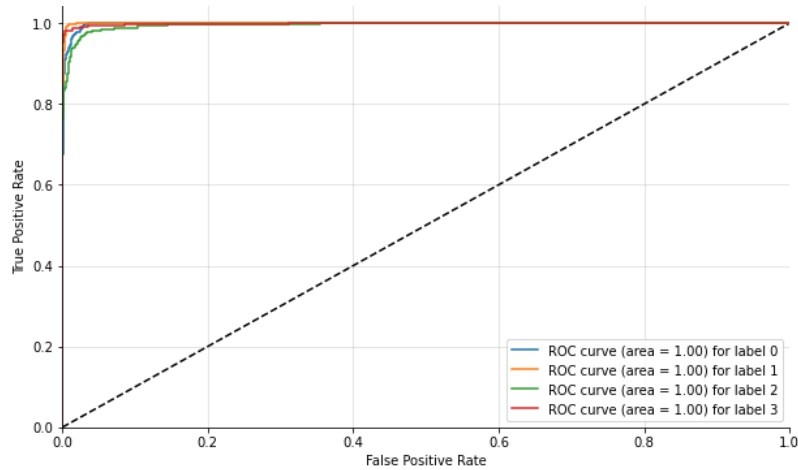


Figure 4.19: ROC Curve of SqueezeNet.

We can see from the ROC curve graph of SqueezeNet that the curves for all four labels of kidney are located in the top-left corner, which indicates that the model performed very well in all 4 types of kidney. However, if we examine the graph at a deeper level, we can see that the model performed the best in the label 1 (normal) category and the least well in the label 2 (stone) category.

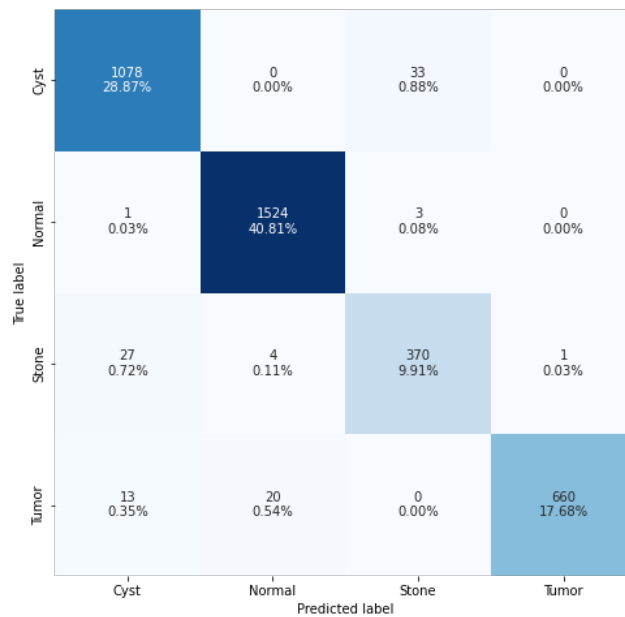


Figure 4.20: Confusion Martrix of SqueezeNet.

SqueezeNet made a total of 3734 predictions and out of these 3734 predictions, 1078 (28.87%) right prediction are made from cyst kidney image class, 1524 (40.81%) from the normal class, 370 (9.91%) right prediction are made from stone kidney image class and lastly 660 (17.68%) right prediction are made from tumor kidney image class. The overall accuracy of this model was 97.3%.

4.4 Discussion of models

Model	Accuracy	Class	Precision	Recall	F1Score	AUC
EAnet	83.65%	Cyst	0.7297	0.9954	0.8421	0.96
		Normal	0.9800	0.8728	0.9233	0.98
		Stone	0.7005	0.3141	0.4337	0.78
		Tumor	0.8119	0.8156	0.8137	0.96
ResNet50	87.92%	Cyst	0.9328	0.9252	0.9290	0.99
		Normal	0.8484	0.9744	0.9070	0.97
		Stone	0.9914	0.2885	0.4470	0.89
		Tumor	0.8552	0.9379	0.8947	0.99
InceptionV3	98.85%	Cyst	0.9927	0.9873	0.9900	1.0
		Normal	0.9980	0.9895	0.9937	1.0
		Stone	0.9275	0.9875	0.9566	1.0
		Tumor	0.9985	0.9984	0.9934	1.0
VGG19	99.89%	Cyst	0.9964	1	0.9982	1.0
		Normal	1.0	1.0	1.0	1.0
		Stone	1.0	0.9900	0.9950	1.0
		Tumor	1.0	1.0	1.0	1.0
SqueeNet	97.27%	Cyst	0.9633	0.9702	0.9668	1.0
		Normal	0.9844	0.9973	0.9908	1.0
		Stone	0.9113	0.9203	0.9158	1.0
		Tumor	0.9984	0.9523	0.9748	1.0

Table 4.1: Classification report of all models.

In our research paper we used 5 different types of deep learning models which are EAnet, ResNet50, InceptionV3, VGG19 and SqueeNet. In the above classification table we can observe VGG19 having highest accuracy, where the overall accuracy of the model is about 99.89%. However, EAnet got the lowest overall accuracy of 83.65%. For instance, the cyst class in VGG19 has precision scores of 99.64%, recall scores of 100%, F1 scores of 99.82%, and auc scores of 100%. The other two classes of stones and tumors for VGG19 are almost identical. On the other hand, cyst class of EAnet have precision score of 72.97%, recall 99.54%, F1 score 84.21% and auc 96%. as well as normal class had precision score of 98%, recall 87.28%, F1 score 92.32% and auc 98%. But we can see a low precision, recall, f1 score in stone and tumor class of EAnet. In ResNet50 we can see that the model accuracy is about to have 87.92%, where the cyst class have precision score of 93.28%, recall 92.52% , f1 Score have 92.90% and auc of 99%, as well as the stone class have a precision score of 99.14%, recall 28.85%, f1 score and auc followed by 44.70% and 89%. Similar goes for tumor class the precision score of 85.52%, recall 93.79%, f1 score and auc followed by 89.47% and 99%. Now in InceptionV3, the model accuracy is very much high compare to EAnet and resnet50 and it was 98.85% and also for the cyst class precision score is 99.27%, recall 98.73%, F1 score 99% and auc 100%, for the tumor class precision score is 99.85%, recall 99.84%, F1 score 99.34% and auc 100%, this score is almost similar goes for stone and normal class. SqueeNet had an accuracy of 97.27% and had a very high precision, recall, f1 and auc score in all four classes. In comparison to the other three models, Stone class recall and

f1 score are significantly lower for EAnet and resnet50. Almost all of the models have lower precision, recall, F1, and AUC scores in the stone class. This means that all 5 models had more difficulty identifying stones in the kidney than cysts, normal cells, or tumors. A high precision score (99.14%) is only seen in the stone class using resnet50, while the scores for cyst, normal, and tumor are only 93.28%, 84.84%, and 85.52%, respectively.

Chapter 5

Conclusion

Prediction and diagnosis of renal disease are important topics of research in the scientific community. A variety of methods, including machine learning, neural networks, deep learning, and others are being implemented in an effort to improve and speed up data processing. Because of this, it is really essential to sort through the requirement to try a variety of approaches before settling on the most productive one. Within the scope of this article, we analyzed the findings obtained from five neural networks used to detect and diagnose renal illness. Our major objective was to demonstrate which networks had the potential to achieve the very best results using the information at their disposal. When all five the models were compared, the one that worked best for all four types of kidney disease (cyst, normal, stone, and tumor) was found and Positive and encouraging outcomes were seen. It will need more research to implement the proposed methods on a larger data set covering a wider range of kidney problems and to create an AI system that can be utilized as an additional tool in automatic classification of CT kidney images.

Bibliography

- [1] S. Beucher, “Use of watersheds in contour detection,” in *Proceedings of the International Workshop on Image Processing*, CCETT, 1979.
- [2] J. Serra, “Image analysis and mathematical morphology,” (*No Title*), 1982.
- [3] R. M. Haralick and L. G. Shapiro, “Image segmentation techniques,” *Computer vision, graphics, and image processing*, vol. 29, no. 1, pp. 100–132, 1985.
- [4] J. B. Roerdink and A. Meijster, “The watershed transform: Definitions, algorithms and parallelization strategies,” *Fundamenta informaticae*, vol. 41, no. 1-2, pp. 187–228, 2000.
- [5] K. C. Saw, J. A. McAteer, A. G. Monga, G. T. Chua, J. E. Lingeman, and J. C. Williams Jr, “Helical ct of urinary calculi: Effect of stone composition, stone size, and scan collimation,” *American Journal of Roentgenology*, vol. 175, no. 2, pp. 329–332, 2000.
- [6] V. Grau, A. Mewes, M. Alcaniz, R. Kikinis, and S. K. Warfield, “Improved watershed transform for medical image segmentation using prior information,” *IEEE transactions on medical imaging*, vol. 23, no. 4, pp. 447–458, 2004.
- [7] 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.
- [8] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.
- [9] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [10] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [11] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, “Squeezenet: Alexnet-level accuracy with 50x fewer parameters and 0.5 mb model size,” *arXiv preprint arXiv:1602.07360*, 2016.
- [12] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826.
- [13] T. Alelign and B. Petros, “Kidney stone disease: An update on current concepts,” *Advances in urology*, vol. 2018, 2018.

- [14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [15] T. Les, T. Markiewicz, M. Dziekiewicz, and M. Lorent, “Automatic recognition of the kidney in ct images,” in *19th International Conference Computational Problems of Electrical Engineering*, IEEE, 2018, pp. 1–4.
- [16] X. Wang, R. Girshick, A. Gupta, and K. He, “Non-local neural networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.
- [17] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, “Convolutional neural networks: An overview and application in radiology,” *Insights into imaging*, vol. 9, no. 4, pp. 611–629, 2018.
- [18] A. J. Brownstein, S. U. B. Mahmood, A. Saeyeldin, *et al.*, “Simple renal cysts and bovine aortic arch: Markers for aortic disease,” *Open Heart*, vol. 6, no. 1, e000862, 2019.
- [19] C. Chen, L. Ma, Y. Jia, and P. Zuo, “Kidney and tumor segmentation using modified 3d mask rcnn,” 2019.
- [20] J. Fu, J. Liu, H. Tian, *et al.*, “Dual attention network for scene segmentation,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3146–3154.
- [21] G. Mu, Z. Lin, M. Han, G. Yao, and Y. Gao, “Segmentation of kidney tumor by multi-resolution vb-nets,” 2019.
- [22] A. Myronenko and A. Hatamizadeh, “Edge-aware network for kidneys and kidney tumor semantic segmentation,” *University of Minnesota Libraries Publishing: Mankato, MN, USA*, 2019.
- [23] Y. Ren, H. Fei, X. Liang, D. Ji, and M. Cheng, “A hybrid neural network model for predicting kidney disease in hypertension patients based on electronic health records,” *BMC medical informatics and decision making*, vol. 19, no. 2, pp. 131–138, 2019.
- [24] G. Chen, C. Ding, Y. Li, *et al.*, “Prediction of chronic kidney disease using adaptive hybridized deep convolutional neural network on the internet of medical things platform,” *IEEE Access*, vol. 8, pp. 100 497–100 508, 2020.
- [25] N. Hadjiyski, “Kidney cancer staging: Deep learning neural network based approach,” in *2020 International Conference on e-Health and Bioengineering (EHB)*, IEEE, 2020, pp. 1–4.
- [26] Z. Hameed, S. Zahia, B. Garcia-Zapirain, J. Javier Aguirre, and A. Maria Vanegas, “Breast cancer histopathology image classification using an ensemble of deep learning models,” *Sensors*, vol. 20, no. 16, p. 4373, 2020.
- [27] H. A. Khan, W. Jue, M. Mushtaq, and M. U. Mushtaq, “Brain tumor classification in mri image using convolutional neural network,” *Math. Biosci. Eng.*, vol. 17, no. 5, pp. 6203–6216, 2020.

- [28] M. Loey, G. Manogaran, and N. E. M. Khalifa, “A deep transfer learning model with classical data augmentation and cgan to detect covid-19 from chest ct radiography digital images,” *Neural Computing and Applications*, pp. 1–13, 2020.
- [29] S. Sudharson and P. Kokil, “An ensemble of deep neural networks for kidney ultrasound image classification,” *Computer Methods and Programs in Biomedicine*, vol. 197, p. 105 709, 2020.
- [30] I. Alnazer, P. Bourdon, T. Urruty, *et al.*, “Recent advances in medical image processing for the evaluation of chronic kidney disease,” *Medical Image Analysis*, vol. 69, p. 101 960, 2021.
- [31] A. V. Gregory, D. A. Anaam, A. J. Vercnocke, *et al.*, “Semantic instance segmentation of kidney cysts in mr images: A fully automated 3d approach developed through active learning,” *Journal of Digital Imaging*, vol. 34, no. 4, pp. 773–787, 2021.
- [32] M.-H. Guo, Z.-N. Liu, T.-J. Mu, and S.-M. Hu, “Beyond self-attention: External attention using two linear layers for visual tasks,” *arXiv preprint arXiv:2105.02358*, 2021.
- [33] M. N. Islam, *Ct kidney dataset: Normal-cyst-tumor and stone*, Nov. 2021. [Online]. Available: <https://www.kaggle.com/datasets/nazmul0087/ct-kidney-dataset-normal-cyst-tumor-and-stone>.
- [34] K. Somasundaram, P. Sivakumar, *et al.*, “An efficient detection of kidney stone based on hdvs deep learning approach,” 2021.
- [35] S. M. Sozio, K. A. Pivert, F. J. Caskey, and A. Levin, “The state of the global nephrology workforce: A joint asn–era–edta–isn investigation,” *Kidney international*, vol. 100, no. 5, pp. 995–1000, 2021.
- [36] K. Yildirim, P. G. Bozdog, M. Talo, O. Yildirim, M. Karabatak, and U. R. Acharya, “Deep learning model for automated kidney stone detection using coronal ct images,” *Computers in biology and medicine*, vol. 135, p. 104 569, 2021.
- [37] Y. Bi, X. Shi, J. Ren, M. Yi, and X. Han, “Transarterial chemoembolization of unresectable renal cell carcinoma with doxorubicin-loaded callispheres drug-eluting beads,” *Scientific Reports*, vol. 12, no. 1, pp. 1–8, 2022.
- [38] D. C. Elton, E. B. Turkbey, P. J. Pickhardt, and R. M. Summers, “A deep learning system for automated kidney stone detection and volumetric segmentation on noncontrast ct scans,” *Medical Physics*, vol. 49, no. 4, pp. 2545–2554, 2022.
- [39] M. Gharaibeh, D. Alzu’bi, M. Abdullah, *et al.*, “Radiology imaging scans for early diagnosis of kidney tumors: A review of data analytics-based machine learning and deep learning approaches,” *Big Data and Cognitive Computing*, vol. 6, no. 1, p. 29, 2022.
- [40] M. N. Islam, M. Hasan, M. Hossain, *et al.*, “Vision transformer and explainable transfer learning models for auto detection of kidney cyst, stone and tumor from ct-radiography,” *Scientific Reports*, vol. 12, no. 1, pp. 1–14, 2022.

- [41] S. S. Parvathi and H. Jonnadula, “An efficient and optimal deep learning architecture using custom u-net and mask r-cnn models for kidney tumor semantic segmentation,”