HAWASSA UNIVERSITY

INSTITUTE OF TECHNOLOGY

FACULTY OF INFORMATICS

MSC PROGRAM IN COMPUTER SCIENCE


# KIDNEY HYDRONEPHROSIS STAGE CLASSIFICATION USING DEEP LEARNING AND IMAGE PROCESSING

**MSC. THESIS**

**BY**


**ALAYU MELAK**


**MAY 2024**

**HAWASSA, ETHIOPIA**

**HAWASSA UNIVERSITY**

**INSTITUTE OF TECHNOLOGY**

**FACULTY OF INFORMATICS**

**MSC PROGRAM IN COMPUTER SCIENCE**

# KIDNEY HYDRONEPHROSIS STAGE CLASSIFICATION USING DEEP LEARNING AND IMAGE PROCESSING

A Thesis Submitted to Hawassa University, Institute of Technology, Faculty of Informatics, Department of Computer Science in Partial fulfillment of the Requirements for the Degree of Master of Science in Computer Science.

**BY: Alayu Melak**

**Advisor: Efrem Yohannes (Ph.D.)**

**MAY 2024**

**HAWASSA, ETHIOPIA**

**APPROVAL SHEET-I**

This is to certify that the thesis entitled,"**KIDNEY HYDRONEPHROSIS STAGE CLASSIFICATION USING DEEP LEARNING AND IMAGE PROCESSING**" submitted in partial fulfillment of the requirements for the degree of Master's with specialization in Computer Science, the Graduate Program of the Faculty of Informatics, and has been carried out by Alayu Melak (Student), and Efrem Yohannes (Ph.D.) (Advisor). Therefore we recommend that the student has fulfilled the requirements and hence hereby can submit the thesis to the department.

| | | |
|---|---|---|
| Efrem Yohannes (Ph.D.) | ______________ | July 03,2014 |
| Name of major advisor | Signature | Date |

**EXAMINERS' APPROVAL SHEET**
**SCHOOL OF GRADUATE STUDIES**
**HAWASSA UNIVERSITY EXAMINERS' APPROVAL SHEET**
**(Submission Sheet-2)**

We, the undersigned, members of the Board of Examiners of the final open defense by **Alayu Melak** have read and evaluated his thesis entitled "**KIDNEY HYDRONEPHROSIS STAGE CLASSIFICATION USING DEEP LEARNING AND IMAGE PROCESSING**", and examined the candidate. This is, therefore, to certify that the thesis has been accepted in partial fulfillment of the requirements for the degree **MCS in computer science.**

| | | |
|---|---|---|
| Efrem Yohannes(Ph.D.) | | July 03,2014 |
| Name of Major Advisor | Signature | Date |
| Dr. V. ANITHA | | July 03, 2024 |
| Name of Internal Examiner-I | Signature | Date |
| | | |
| Name of Chairperson | Signature | Date |
| Michael Melese (PhD) | | July 03,2024 |
| Name of External examiner | Signature | Date |
| | | |
| SGS Approval | Signature | Date |

Final approval and acceptance of the thesis is contingent upon the submission of the final copy of the thesis to the School of Graduate Studies (SGS) through the Department/School Graduate Committee (DGC/SGC) of the candidate's department.

**Stamp of SGS Date: ____________**

**Remark**

- Use this form to submit the thesis with ***MINOR CORRECTION*** suggested by the examining board

# DECLARATION

I, the undersigned, declare that this thesis entitled "**KIDNEY HYDRONEPHROSIS STAGE CLASSIFICATION USING DEEP LEARNING AND IMAGE PROCESSING**" was prepared by me with the help and guidance of my advisor. I have acknowledged and referenced all materials used in this work. I understand that non-adherence to the principles of academic honesty and integrity, and misrepresentation fabrication of any idea/data/fact/source will constitute sufficient grounds for disciplinary action by the University and can evoke penal action from the sources, which have not been properly cited or acknowledged.

Name of the student __Alayu Melak____________Signature _____________

Date of submission: _____________

Place: Hawassa

This thesis has been submitted for examination with my approval as a university advisor.

Advisor Name: Efrem Yohannes (Ph.D.)

Advisor's Signature: _____________

# ACKNOWLEDGMENTS

First and foremost, I would like to thank my **Almighty God**, and Ever-Virgin, Mother of our Lord for giving me the strength, knowledge, ability, and opportunity to undertake this thesis work.

Second I would like to express my greatest appreciation to my Advisor, Efrem Yohannes (Ph.D.). I am deeply thankful to him for giving his insights and constructive comments, criticism, qualified advice, and continuous guidance starting from title selection to the completion of this research. I am very proud of working with you.

I am very thankful to my friend, Dinkie Gashaye, for her continuous help, giving constructive comments and support throughout this study.

I am also extremely grateful to my parents for their unwavering support, love, encouragement, help, prayers, caring, and sacrifices for educating me starting from early school and preparing me for my future.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| AI | Artificial intelligence |
| ANN | Artificial Neural Network |
| API | Application Programming Interface |
| CNN | Convolutional Neural Network |
| CT | Computed Tomography |
| DL | Deep learning |
| ESWL | Extracorporeal Shock Wave Lithotripsy |
| FP | False Positive |
| FP | False Negative |
| JPEG | Joint Photographic Experts Group |
| KHSC | Kidney Hydronephrosis Stage Classification |
| LSVRC | Large-Scale Visual Recognition Challenge |
| MSE | Mean Squared Error |
| MRI | Magnetic Resonance Imaging |
| NLM | Non-local Means |
| PNG | Portable Network Graphics, |
| PSNR | Peak Signal-to-Noise Ratio |
| Relu | Rectified Linear Unit |
| RN | Residual Network |
| RNN | Recurrent Neural Networks |
| SAR | Synthetic Aperture Radar |
| TN | True Negative |
| TP | True Positive |
| VGG | Visual Geometry Group |

# ABSTRACT

Kidney hydronephrosis is a common condition characterized by the dilation of the renal pelvis and calyces due to obstruction in the urinary tract. In, today, where multiple scanned images are obtained for each patient, the current hydronephrosis stage classification is time-consuming, depends on the knowledge and the mood of the radiologist, and is susceptible to variability in classification between and among observers. Deep learning techniques can be utilized for kidney hydronephrosis stage classification. Utilizing deep learning in kidney hydronephrosis stage classification reduces the workload of radiologists and increases diagnosis accuracy. In this thesis, we explore the use of deep learning models, specifically VGG16, ResNet50, and InceptionV3, combined with image preprocessing techniques like noise reduction, image resize, augmentation, segmentation, feature extraction, and image normalization to improve model performance. Additionally, we conduct comparative analyses to assess the robustness and generalization capabilities of the fine-tuned models across kidney CT-scanned datasets. A total of 7400 CT scanned images are used to build the model with 80% of the dataset used for training, 10 % for validation, and 10% for testing. Features from kidney CT images are extracted using transfer learning with pre-trained VGG16, ResNet50, and InceptionV3 models and fed into fully connected layers for stage classification. The performance of each model is evaluated using model evaluation metrics of accuracy, f1 score, and recall. The accuracy score of ResNet50, VGG16, and InceptionV3 is 79%,91%, and 74% respectively, 87%,86% and,63% recall, 85%,86%, and 67% f1 scores. Comparing those values VGG16 has higher accuracy, recall, and f1 score, and is larger and is the best model for kidney hydronephrosis stage classification from CT scanned images. Misclassifications occurred due to the similar structure of kidney images at each level that caused hydronephrosis. Future work should address these challenges. This research contributes to the advancement of automated kidney hydronephrosis stage classification by using deep learning and image processing techniques. The findings highlight the potential of VGG16 as a powerful model for accurate and efficient classification of kidney hydronephrosis stages.

**Keywords: Features Extraction, Kidney hydronephrosis, InceptionV3, ResNet50, Transfer learning, VGG16**

# CHAPTER ONE

# INTRODUCTION

## 1.1. Background

Human disease is an abnormal condition affecting an individual's body, leading to dysfunction or discomfort. Diseases can affect any part of the body and range in severity from mild and temporary to severe and chronic. Nephrolithiasis, another name for kidney stones, is a common and excruciating urological illness that impacts millions of people worldwide [2]. It is typified by the accumulation of solid mineral deposits in the kidneys or urinary tract. Solid mineral and salt deposits that accumulate in the kidneys are known medically as renal calculi [3]. If treatment is not received, these tiny, hard mineral deposits that develop in the kidneys can cause excruciating pain, interfere with daily activities, and result in serious consequences. They can vary in size and shape, from tiny crystals to larger, more complex structures. These stones develop when there is an imbalance in the substances that make up urine. All kidney stones can grow to damage areas of normal kidney tissue if left untreated, which could lead to disability and possibly be fatal.

Kidney stones are becoming more common [4]. According to the Ethiopian Public Health Association hundreds of thousands of visits to dialysis services, and kidney implants [5]. Several factors contribute to kidney stone formation. The buildup of mineral and salt crystals that are expelled in urine and eventually turn into stones is the cause of kidney stone disease. A combination of environmental and genetic factors is also responsible for their cause [6]. Dehydration is a major risk factor because it concentrates minerals in the urine. In addition, smoking, drinking too little water, being overweight, eating a poor diet, getting too little sleep, and many other things can contribute to it. An excessively salty, protein-rich, and oxalate-rich diet can also raise the risk of kidney stones. Kidney stones can be more common in people with specific medical conditions, such as hyperparathyroidism, urinary tract infections, and a family history of the disease [7]. All racial, cultural, and geographic groups are affected by kidney stones.

Artificial Intelligence (AI) is revolutionizing various fields by enabling machines to perform tasks that typically require human intelligence[8]. AI comprises a variety of technologies, such as machine learning, deep learning, natural language processing, and computer vision. These technologies enable AI systems to analyze massive volumes of data, identify patterns, and make conclusions without any human participation.

Artificial intelligence, renal disease, and health are converging to change the way that medicine is provided[9]. AI-driven technologies are increasingly being used to enhance the early classification, detection, and management of kidney disease. By analyzing patient data, machine learning algorithms are able to forecast the course of diseases, and detect risk factors,

Hydronephrosis is the swelling of the kidney due to the backup of urine caused by an obstruction in the urinary tract [10]. This obstruction can occur at any point in the urinary tract, impeding the normal drainage of urine from the kidneys to the bladder. Common causes include kidney stones, tumors, or congenital abnormalities. The increased pressure within the kidneys can lead to kidney damage over time, potentially resulting in renal impairment if left untreated. Diagnosis typically involves imaging studies such as ultrasounds or CT scans, and management varies depending on the underlying cause, ranging from conservative approaches to surgical interventions aimed at relieving the obstruction and restoring normal urinary flow.

Since there is a rapid growth in digital image processing and computer vision it is important to develop an improved computer-based stage classification system that can reduce the current kidney stone hydronephrosis stage classification.

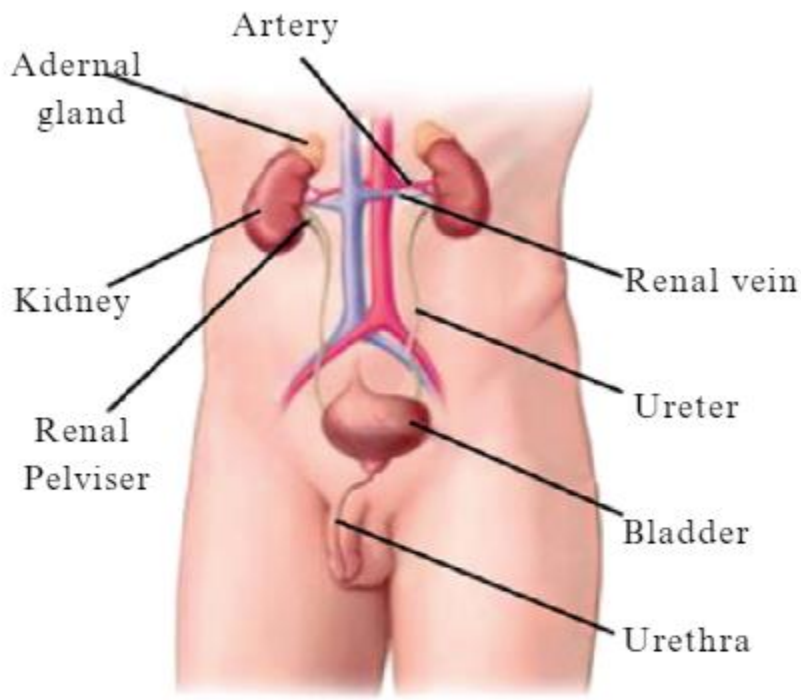



Figure 1. 1 Structure of Kidney[11]

### 1.2. Motivation

We live in a world dominated by technology on a significant scale, major network coverage, high-end smartphones, machine learning, Deep Learning, and discoveries and improvements in AI. Deep learning models have been increasingly applied in disease detection, classification, and image feature extraction. Technology development in recent years has led to the widespread adoption of machine learning and deep learning approaches for disease diagnosis[12]. These methods provide a reliable tool for making definitive diagnostic decisions that require long and complex processes because both the diagnostic accuracy and diagnosis time are shortened by them.

Kidney hydronephrosis is a prevalent urological condition that impacts millions of people globally, causing significant pain and potentially leading to severe complications. Early and accurate stage classification of kidney hydronephrosis is crucial for effective patient management and treatment planning.

Among many other killer diseases, kidney hydronephrosis is the most severe cause of death in the world, which affects the kidney part of the human body [13]. Medical professionals can currently classify the stage of kidney hydronephrosis using CT scans, ultrasound, and magnetic resonance imaging. The potential to improve patient outcomes, lower healthcare costs, and increase the overall effectiveness of healthcare systems are some of the driving forces behind research into kidney hydronephrosis stage classification and machine learning-based hydronephrosis stage classification. The key factors that initiate the conduct of studies on hydronephrosis stage classification are:

- Kidney hydronephrosis is a killer disease so timely and accurate classification of its severity helps to take appropriate medical treatment based on the stage of the disease.
- Early hydronephrosis prevention, and stage classification: timely prevention and classification of hydronephrosis based on its severity can aid in preventing irreversible kidney damage.
- Diagnosing problems leads healthcare professionals to provide incorrect medical treatment, which may aggravate the problem.

### 1.3. Statement of the Problem

Kidney hydronephrosis disease may pose a serious threat to life. Consequently, kidney hydronephrosis must be diagnosed as soon as possible. Sometimes kidney hydronephrosis is difficult to identify with the naked eye due to the low quality of the images, which might cause false alarms. Today, where multiple scanned images are obtained for each patient, the current hydronephrosis stage classification is time-consuming and susceptible to variability in classification between and among observers. The rate of errors in clinical radiology practices is 4%. Even though 4% might not seem like much, when we take into account all of the scans that are done globally, it adds up to a substantial number of errors [14]. It is a very time-consuming and tedious task. Hence, it is associated with many challenges, such as the depth of knowledge that the radiologist possesses, and the mood of the radiologist. Existing methods for kidney hydronephrosis stage classification often rely on manual interpretation of medical imaging, which can be time-consuming, subjective, and error-prone. Therefore, there is a pressing need for advanced computer-aided techniques to enhance the process of kidney hydronephrosis stage classification.

Diseases are diagnosed in medicine using the expertise of medical professionals [15]. This study proposed better and more reliable results for patients so that more patients can be diagnosed and cured. In line with this, early identification of kidney hydronephrosis is very useful in encouraging a good chance of patient recovery**.** The primary problem to be addressed in this study is the development of a robust and efficient model for kidney hydronephrosis stage classification.

By addressing these aspects, this study aims to contribute to the improvement of kidney hydronephrosis stage classification and management, ultimately enhancing patient care and reducing the burden of kidney hydronephrosis-related complications on healthcare systems and individuals.

## 1.4. Research Questions

- Which image preprocessing techniques are suitable for kidney CT hydronephrosis images?
- Which transfer learning model performs best for kidney hydronephrosis image classification?
- Which Deep learning technique model achieves the best result for kidney hydronephrosis stage classification?
- To what extent does the developed model classify the kidney hydronephrosis stage?

## 1.5. Objectives

### 1.5.1. General Objective

The general objective of this study is to develop a kidney hydronephrosis stage classification model using deep learning and image processing techniques.

### 1.5.2. Specific Objective

To achieve the general objective, the following specific objectives are identified:

- To select a suitable preprocessing technique for kidney CT scanned image.
- To determine appropriate deep learning pre-trained feature extraction techniques for hydronephrosis stage classification model development.

- ➢ To determine which deep learning pre-trained model achieves the best result for kidney hydronephrosis stage classification.
- ➢ To evaluate the performance of the developed models.

### 1.6. Scope and Limitations of the Study

The focus of this research is to design and create a model for kidney hydronephrosis stage classification using kidney-scanned images(only computed tomography (CT) scanned) as input, image preprocessing, and Deep learning. However, it does not include samples taken from laboratory blood and urine tests and does not classify hydronephrosis caused by women's pregnancy, kidney tumors, or other conditions.

### 1.7. Significance of the Study

In today's where multiple scanned images are obtained for each patient, the manual classification of kidney hydronephrosis in laborious is susceptible to variability between and among observers. The rate of errors in clinical radiology practices is 4%. Even though 4% might not seem like much, when we take into account all of the scans that are done globally, it adds up to a substantial number of errors [14]. It is a very time-consuming and tedious task, and hence it is associated with many challenges, such as the depth of knowledge that the radiologist possesses, the mood of the radiologist, and the accuracy of the machine.

In most fields of medicine that require specialized knowledge, radiology is among the top therefore there is a need for a kidney hydronephrosis model that can assist the physicians in reading the images. In addition, even though radiologists' manual methods of detecting, classifying, and analyzing scanned images are reliable, with no doubt it is tedious, laborious, time-consuming, and highly subjective. Automation of the kidney hydronephrosis stage classification process is therefore highly desired. The proposed study has the following significance in the medical area.

1. Using deep learning techniques lowers risk and improves classification accuracy while lightening the workload for doctors.
2. Improve patient outcomes: Ultimately, the use of deep learning in kidney hydronephrosis stage classification can lead to better patient outcomes by enabling earlier intervention and accurate diagnosis.

3. The use of deep learning in medical imaging, such as CT scans improves diagnostic accuracy and reduces the possibility of human errors across a wide range of medical areas.

### 1.8. Thesis Organizations

The thesis is organized into five chapters in each chapter there is an organized sub-topic to describe the chapter topic roles in detail explanations. The first chapter describes the background, study area, problem to be solved, significance, scope, aim of the study, and limitation of the study are included. Chapter two discusses literature reviews on mechanisms of data collection, processing techniques, feature extraction techniques, and how models are developed to classify accurately the stage of kidney hydronephrosis diseases. Chapter three represents the selected techniques starting from data collection to model building including research flow, and system model architecture of kidney hydronephrosis stage classification. The experiment results and discussions as well as model performance are presented in chapter four of the study. The thesis, conclusion, contributions, and recommendations for future work are discussed in the last chapter which is in chapter five.

# CHAPTER TWO

# LITERATURE REVIEW

## 2.1. Introduction

In this Chapter, an overview of kidney hydronephrosis, medical imaging techniques, and several previous research works that have been done related to automated kidney hydronephrosis stage classification systems using different approaches have been reviewed. Then different techniques of digital image processing (image acquisition, image preprocessing, feature extraction, segmentation, and classification), and deep learning are discussed in detail. Many proposed techniques have been introduced by different researchers for kidney disease with Computer-Aided Systems. The field of focus and the gap, results of these thesis works are also presented and discussed.

## 2.2. Human Health

Health is a broad notion that includes physical, mental, and social well-being [1]. It is not just the absence of disease or disability, but also an overall approach to sustaining a healthy lifestyle. Proper nutrition, regular physical activity, adequate sleep, and mental health care are essential components of a healthy lifestyle. Several diseases destroy human normality or health status. One of them is kidney disease which causes the loss of proper functionality on the kidney organ. One of the diseases is kidney stone which leads to hydronephrosis.

## 2.3. Kidney Stone and Hydronephrosis

The urinary tract's anatomy consists of a bladder, two ureters, and two kidneys [16]. The kidney is an organ with the shape of a double bean that can process 200 quarts of blood and produce two-quarters of urine, which is made up of water and waste particles. Urine travels via the ureter tubes from the kidney to the bladder [17]. When a person urinates, the bladder fills with urine and releases it. Stones are formed in the urinary bladder, ureter, or kidney due to the collection of chemical substances[18].

Hydronephrosis is a medical condition marked by the swelling or enlargement of one or both kidneys due to the accumulation of urine[7]. This occurs when urine cannot flow properly from the kidney to the bladder, resulting in an obstructive backup. The primary cause of hydronephrosis is obstruction within the urinary tract, often attributed to conditions such as kidney stones, tumors, blood clots, or congenital abnormalities. Additionally, structural issues like vesicoureteral reflux and ureterocele can contribute to the condition. Even pregnancy may induce hydronephrosis as the growing uterus exerts pressure on the ureters, impeding the normal flow of urine.

The symptoms of hydronephrosis are typically noticeable and include flank pain, urinary frequency, urgency, discomfort during urination, and in severe cases, nausea, vomiting, and fever[20].

The diagnosis involves various imaging studies such as ultrasound, CT scans, or MRI to visualize the kidneys and urinary tract, urodynamic studies, and blood/urine tests. Early identification and classification are crucial to prevent complications, such as kidney damage or recurring infections.

Treatment strategies involve addressing the underlying cause, which may include surgery to remove obstructions or repair abnormalities, pain management, antibiotics for associated infections, and sometimes the placement of a ureteral stent to facilitate urine flow.

The prognosis for hydronephrosis varies based on the underlying cause, the promptness of diagnosis, and the extent of kidney damage. Timely intervention is essential to prevent irreversible harm to the kidneys and ensure a more favorable outcome[21]. Overall, hydronephrosis underscores the importance of a comprehensive approach in identifying and managing both the symptoms and the root causes of urinary obstruction to preserve kidney function and promote overall well-being. If an individual suspects hydronephrosis or experiences related symptoms, consulting with a healthcare professional is crucial for an accurate diagnosis and appropriate treatment [22].

In recent years, the occurrence of kidney hydronephrosis has been on the rise [18]. Unfortunately, many of these are classified too late, after symptoms appear. It is much easier and safer to remove a small stone than a large one. More percent of hydronephrosis starts as a lower-grade stage. However, mild hydronephrosis becomes severe hydronephrosis. Once symptoms appear, it is generally too late to treat the kidney stone.

Based on the severity of the condition hydronephrosis can be categorized into mild, moderate, or severe based on imaging studies.

### 2.3.1. Mild Hydronephrosis

In the mild stage of hydronephrosis, there is typically slight dilation of the renal pelvis, which is the part of the kidney where urine collects before flowing into the ureter and eventually the bladder [23]. This dilation indicates some degree of urinary obstruction, causing a backup of urine within the kidney. Despite this mild dilation, kidney function is generally preserved, and individuals may not experience significant symptoms. In some cases, mild hydronephrosis may be asymptomatic and only discovered incidentally during imaging studies conducted for unrelated reasons.

### 2.3.2. Moderate Hydronephrosis

In the moderate stage of hydronephrosis, there is a more significant dilation of the renal pelvis compared to the mild stage[23]. This increased dilation suggests a more pronounced obstruction in the urinary tract, leading to a greater accumulation of urine within the kidney. As a result, individuals with moderate hydronephrosis may experience more noticeable symptoms, including flank pain, urinary urgency, and discomfort during urination. While kidney function is often still preserved at this stage, there is an increased concern for potential complications and a need for more active management.

### 2.3.3. Sever Hydronephrosis

In the severe stage of hydronephrosis, there is a substantial and often dramatic dilation of the renal pelvis, signifying a significant obstruction in the urinary tract[23]. At this advanced stage, the swelling of the kidney is profound, and individuals may experience severe symptoms, including intense flank pain, frequent urination, and potential complications such as nausea, vomiting, and fever[20]. There is an increased risk of damage to kidney function, and urgent medical attention is essential to address the underlying cause and prevent further deterioration.

The location, composition, and size of kidney hydronephrosis stones all affect how they are treated. With more fluid intake and pain relief, small stones may pass naturally. Larger kidney stones might need to be treated with ureteroscopy, shock wave lithotripsy, or percutaneous nephrolithotomy [2].

Preventive measures include eating a balanced diet, staying hydrated, and taking care of any underlying medical conditions.

Depending on the kind of kidney stone and its cause, there are different approaches to treating hydronephrosis.

1. Hydronephrosis with minimal symptoms.

The majority of hydronephrosis with minimal symptoms can be treated non-invasively. A tiny stone might be able to go through:
Drinking water: up to two to three quarts (1.8 to 3.6 liters) of fluid per day will keep urine diluted and may help avoid the formation of stones.

Pain relievers: tiny stones can be uncomfortable to pass. A doctor might suggest taking naproxen sodium (Aleve) or ibuprofen (Advil, Motrin IB, and other brands) to treat mild pain.

Medical therapy: a medication to help pass kidney stones may be prescribed by doctors. By relaxing the ureter's muscles, this kind of medication known as an alpha-blocker allows to pass kidney stones more quickly and with less discomfort. Tamsulosin (Flomax) and the medication combination dutasteride and tamsulosin (Jalyn) are examples of alpha-blockers.

2. Hydronephrosis with more symptoms.

More intensive treatment may be necessary for kidney stones that are too big to pass on their own or that result in bleeding, kidney damage, or persistent urinary tract infections. Procedures could consist of:

Using sound waves to break up stones: Extracorporeal shock wave lithotripsy (ESWL) is a procedure that doctors may recommend for specific kidney stones, depending on their size and location. By producing powerful vibrations known as shock waves, ESWL breaks up stones into small fragments that can be excreted in urine [24]. Patients may be given light anesthesia or be sedated during the 45–60-minute procedure, which has the potential to cause moderate pain.

Surgery to remove very large stones in the kidney: - A kidney stone can be surgically removed during a procedure known as a percutaneous nephrolithotomy. Small telescopes and instruments are inserted through a tiny incision in the patient's back.

## 2.4. Medical Imaging Diagnosis Techniques

Visual representations of internal organs are produced for clinical studies through medical imaging[20]. The conventional diagnosing method for hydronephrosis varies from one medical organization to another. It depends on the medical center. According to the American Urological Association, the current gold standard for confirming kidney stones is a non-contrast CT of the abdomen and pelvis [25]. However, by incorporating machine learning and Digital Image Processing, it is possible to make a classification of hydronephrosis. This can be implemented using a computer-aided system. Computer-aided systems and technology show that Computer-aided systems can help to improve the hydronephrosis stage classification accuracy of radiologists, lighten the burden of increasing workload, reduce errors missed due to fatigue, overlooked or data overload, and improve inter and intrareader variability [26].

Medical images nowadays play an essential role in the detection and diagnosis of numerous diseases [26]. There are various types of medical imaging acquisition techniques and the most common are X-ray, Ultrasound, CT scan, and MRI. These imaging modalities have been used to train deep learning (DL) models that are accurate and robust enough to aid doctors in diagnosing conditions like cardiac abnormalities, COVID-19, prostate cancer, kidney stones, brain tumors, breast cancers, and skin cancers.

A. X-rays, or Radiography: is a form of electromagnetic radiation, just like light waves and radio waves [27]. They are a widely used medical imaging technique that plays a crucial role in the diagnosis of various diseases and conditions. It involves the use of a small amount of ionizing radiation to create images of the inside of the body. X-ray images, also known as radiographs, help healthcare professionals visualize the structure and condition of bones, organs, and tissues. X-rays are a simple method and cost-effective.
B. CT scan, or a Computed Tomography Scan, is a medical imaging technique that uses X-rays and computer processing to create detailed cross-sectional images of the body. It is one of the most common imaging models used to identify kidney stones by clinical experts [20]. It provides more detailed information than traditional X-rays and is commonly used in disease diagnosis and treatment planning. During a CT scan, a patient rests on a table that slides into a large, tunnel-shaped scanner [24]. Some exams require that a contrast dye be injected into a vein before the procedure.

C. Ultrasound scan, also known as sonography, is a non-invasive medical imaging technique that utilizes high-frequency sound waves to create real-time images of internal body structures [28]. It is commonly employed to visualize organs, tissues, and blood flow patterns, aiding in diagnosing various medical conditions, including pregnancy monitoring, abdominal disorders, and cardiovascular diseases. Unlike X-rays, ultrasound does not involve ionizing radiation, making it safe for use during pregnancy and repeated examinations. Additionally, its portability and versatility make it a valuable tool in diverse medical settings, offering quick and detailed insights into the body's anatomy and function.
D. MRI is a type of medical imaging that creates finely detailed images of the body's internal structures by using radio waves and a strong magnetic field[24]. An MRI does not use ionizing radiation, in contrast to CT or X-ray scans. Rather, it depends on specific atoms' magnetic characteristics that are found in the body, usually hydrogen atoms found in water molecules.

The gold standard method for kidney stone hydronephrosis detection, size measurement, and localization is a non-contrast CT (NCCT) scan of the kidneys, ureters, and bladder [29]. In the past years, medical imaging has seen significant advancements for diagnosis disease. Currently, diagnosis of disease are effective on the use of existing technology which is artificial intelligence (AI) . AI has major roles in the health sector for different approaches such as disease epidemic prediction, and classification using various ways of data representation. These advancements have improved the accuracy, efficiency, and overall outcomes of diagnostic procedures. The field of AI which have advantageous on the health field are Machine learning and Deep Learning. Using of deep learning in the large dataset have outperformed result than traditional machine learning.

## 2.5. Deep Learning

Deep learning(DL) is a subset of Machine Learning that follows a multi-layered strategy that aids in performing better classification functions like identifying anomalies and irregularities in medical images, patient clustering based on similar characteristics and levels of severity, or finding the interrelation between outcomes and disease-related symptoms[25].

In DL, artificial neural networks with multiple layers between the input and output layers are used to model high-level abstractions in data. These networks are capable of learning from large amounts of labeled or unlabeled data, automatically discovering intricate patterns and features that are difficult to capture with traditional machine-learning techniques.

Deep learning has been successfully applied to various tasks such as image recognition, speech recognition, natural language processing, and more. Some popular deep learning architectures include convolutional neural networks (CNNs) for image-related tasks, recurrent neural networks (RNNs) for sequence data and transformer models for natural language processing tasks like language translation and text generation.

### 2.5.1. Transfer Learning

Transfer learning is the process by which the information/knowledge obtained while resolving a particular problem can be applied to solve another problem on similar lines[47]. Transfer Learning is an approach to adapt and apply the knowledge acquired on another related task to the task at hand. A model is developed to solve tasks and transfer the knowledge of the model to related tasks to solve the same problems in the image recognition task. However, it is difficult to train the convolutional neural network from scratch due to the required large image data.

### 2.5.2. Convolutional Neural Network

Convolutional Neural Networks (CNNs) represent a class of deep learning models specifically designed for processing structured grid data, such as images [32]. CNNs have computer vision tasks by effectively capturing hierarchical features in an image through a series of convolutional and pooling layers. The most important and unique component of a convolutional neural network over other neural networks is the convolution layer. The convolutional layers apply filters to detect patterns and features in local regions of the input, while pooling layers reduce spatial dimensions, preserving essential information. This architecture enables CNNs to automatically learn and extract meaningful representations, making them highly effective for tasks like image classification, object detection, and facial recognition.

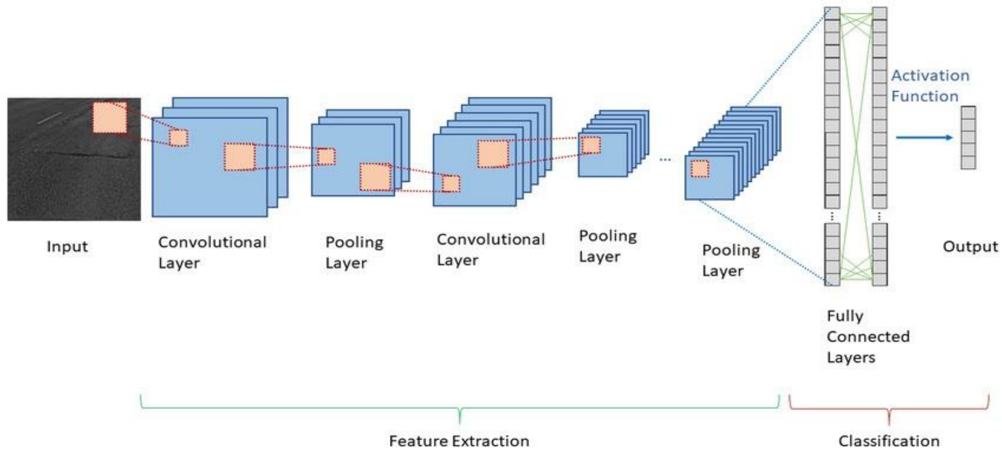



Figure 2. 1 Convolutional neural network architecture [48].

Convolutional Neural Networks (CNNs) consist of several key components that enable them to process and analyze images effectively.

Convolutional layers: - The convolutional layer is the first layer in Convolutional Neural Networks. The convolution layer performs mathematical operations of convolutions on the input images for the classification task using a set of filters (kernels) to extract feature maps [49]. The kernels have dimensions of N * N * D, where N * N is the size of the kernel and D is the number of channels that correspond to the channels in the images convolved. The filters slide over the input, capturing local patterns and features. Convolutional layers help the network automatically learn hierarchical representations. In this layer, a set of filters is applied to the image to identify and extract specific features, such as edges, corners, and textures.

| | | |
|---|---|---|
| a | b | c |
| d | e | f |
| g | h | i |

Input

| | |
|---|---|
| x | y |
| z | w |

Kernal

| | |
|---|---|
| ax+by+dz+ew | bx+cy+ez+fw |
| dx+ey+gz+hw | ex+fy+hz+iw |

output

Figure 2. 2 Convolutional operation

Activation functions: the activation function of a convolutional neural network determines which output neuron, out of the inputs from a particular set of nodes, should be activated [50]. Introduce non-linearity to the model, allowing it to learn complex relationships in the data. Common activation functions used in CNNs include:

1. ReLU (Rectified Linear Unit): - ReLU is one of the most widely used activation functions in CNNs. It introduces non-linearity by setting all negative values to zero and leaving positive values unchanged. Mathematically, it is given by this simple expression in the equation below.

$$R(x) = \begin{cases} 0, x < 0 \\ x, x \geq 0 \end{cases} \quad \text{or} \quad R(x) = max\,(0, x) \qquad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(2.1)$$

Where (x) is the input to the function,

R(x) is the output.

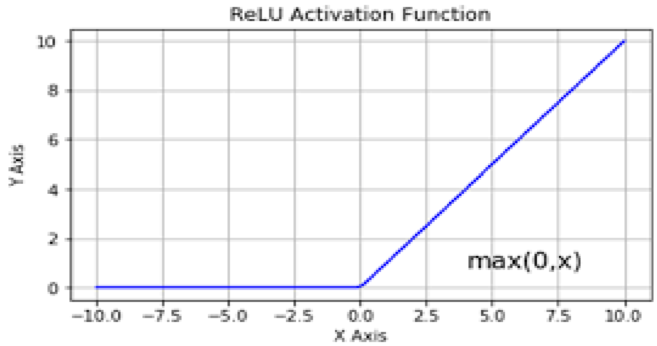



Figure 2. 3 Relu Activation Function.[51]

2. A Sigmoid function is a mathematical function that has a characteristic S-shaped curve[52]. Sigmoid functions have the property that they map the entire number line into a small range such as between 0 and 1, or -1 and 1, so one use of a sigmoid function is to convert a real value into one that can be interpreted as a probability. Mathematically, it is described by the formula,

$$\alpha(x)= \frac{1}{1+e-x} \qquad \text{……………………………..………………………..……………….(2.2)}$$

Where (x) represents the input to the function,

$\alpha(x)$ is the output and

$e$ represents Eulers's number which is approximately equal to 2.71828.

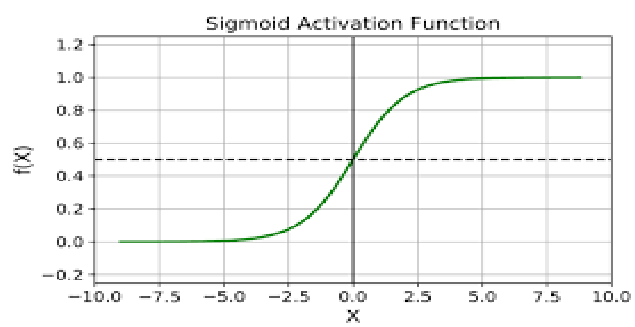



Figure 2. 4 Sigmoid activation function [53]

3. Softmax:- In the output layer of CNNs, softmax is frequently utilized for multi-class classification tasks. It assures that the total of the probabilities equals one by

normalizing the output into a probability distribution across several classes. Mathematically, it is described by using the formula:-

$$\frac{e^{z}i}{\sum_{i=0}^{K} e^{zJ}} \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots (2.3)$$

Where (z_i) is the input to the softmax function for class(i).

(K )is the total number of classes.

$e$ represents Eulers's number which is approximately equal to 2.71828.

(z_j) is the input to the softmax function for class(j).

Pooling layers: To lower the dimension of the feature map while preserving the information required to minimize the number of network parameters and the computational cost of the network, a pooling layer is inserted between two convolutional layers[49]. The pooling layer reduces the size of the input image by combining the features of neighboring pixels into one feature value. Downsample the spatial dimensions of the input data, reducing its resolution. Max pooling, for example, selects the maximum value from a group of neighboring pixels, helping to retain essential information while reducing computational complexity [48].

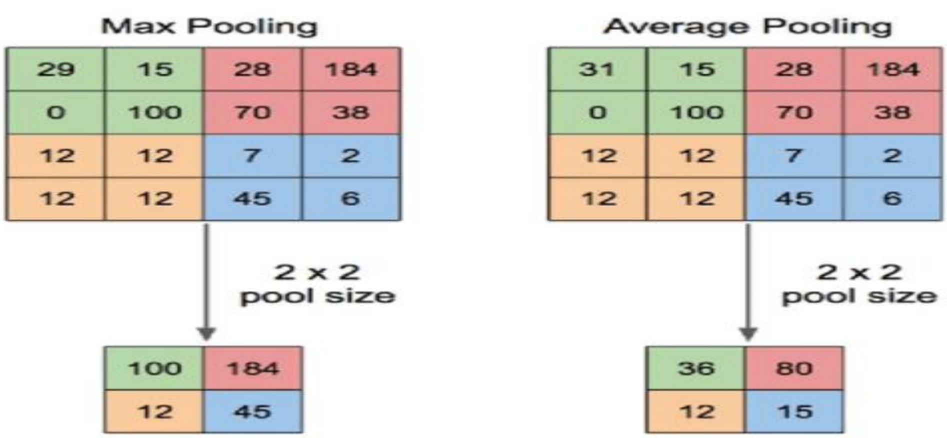



Figure 2. 5 Max and average poling [54]

Fully connected layers: The fully connected layer is the last in CNN architecture [49]. These layers connect every neuron from one layer to every neuron in the next layer. The result from the last pooling or convolutional layer is the input to the dense or fully connected layer, in which it is flattened and provided to the fully connected layer. Fully connected layers are typically used in the later stages of the network to combine high-level features and make final predictions.

Flattening: Before connecting to fully connected layers, the output from the convolutional and pooling layers needs to be flattened into a one-dimensional vector. Creates a vector that is needed as input to the artificial neural network. It is needed because ANN requires a long vector.
This process retains the spatial relationships learned by the convolutional layers. Creates along vector that is needed as input to the neural network.

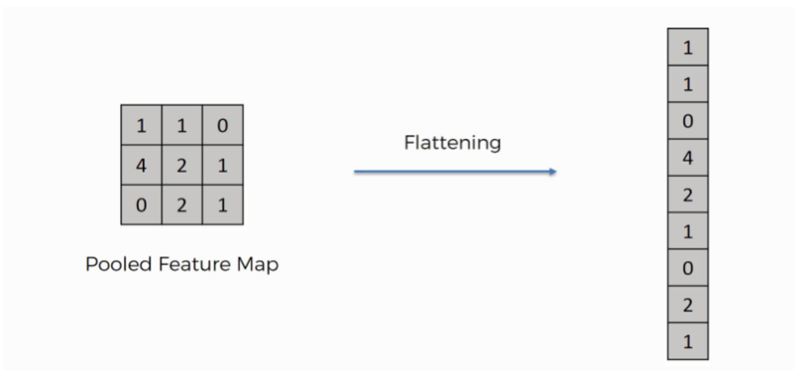



Figure 2. 6 Flattening [55]

## 2.6. Feature Extraction

Features are said to be properties that describe the whole image. Feature selection in image processing refers to the process of choosing a subset of relevant features (attributes or characteristics) from the original set of features in an image [56].

Feature extraction aims to reduce the dimensionality of the data by retaining only the most discriminative and significant features, thereby speeding up processing algorithms and enhancing the overall performance of image processing systems.

Deep learning offers powerful feature extraction techniques that have revolutionized various fields, from computer vision to natural language processing[57]. One of the key strengths of deep learning lies in its ability to automatically learn intricate and hierarchical representations directly from raw data. Convolutional Neural Networks (CNNs), a type of deep learning architecture widely used in image analysis, excel at extracting meaningful features from images. These networks consist of multiple layers that learn progressively complex features, starting from simple edges and textures to high-level object representations. CNNs have been instrumental in tasks like image classification, object detection, and image segmentation due to their feature extraction capabilities.

Another noteworthy deep-learning technique for feature extraction is Transfer Learning. Transfer learning leverages pre-trained models, such as VGG, ResNet, or Inception, that have been trained on large-scale datasets like ImageNet. By utilizing these pre-trained models, researchers can benefit from the learned features without needing to train the entire network from scratch. Transfer learning is particularly useful when working with limited datasets or when computational resources are constrained. It allows practitioners to fine-tune pre-trained models on specific tasks, facilitating rapid development and deployment of deep learning solutions.

The selected features should ideally capture the essential information needed for a specific task, such as object recognition, classification, or segmentation while minimizing redundancy and irrelevant details [58]. Techniques for feature selection include statistical methods, information-theoretic measures, machine learning algorithms, and conventional neural network pre-trained models that evaluate the contribution of each feature to the overall task. By focusing on the most relevant features, feature selection helps improve the accuracy, reduce computational complexity, and enhance the interpretability of image processing systems.

## 2.7. CNN Model Architectures

A. VGG16

VGG is popular because of its simplicity and uniform architecture, making it easy to understand and implement. The VGG16 architecture consists of 16 convolutional and fully connected layers. The convolutional layers are stacked one after the other, and they use small 3x3 filters. Max-pooling layers with 2x2 filters are used to downsample the spatial dimensions.

The VGG16 architecture strictly uses 3 by 3 convolution kernels with intermediate max-pooling layers for feature extraction and a set of three fully connected layers toward the end for classification. Each convolution layer is followed by a ReLU layer in the VGGnet architecture. The design choice of using smaller kernels leads to a relatively reduced number of parameters, and therefore efficient training and testing [48].

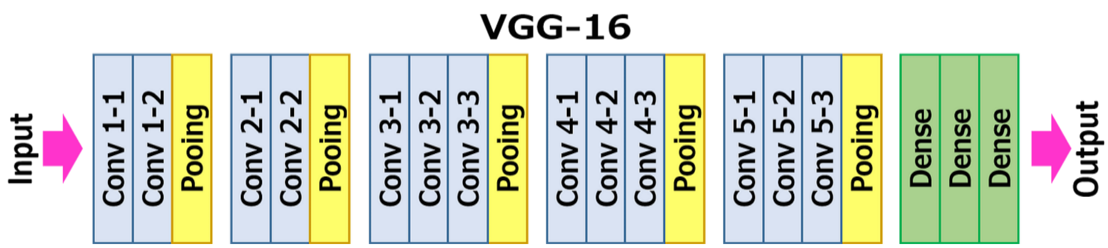



Figure 2. 7 VGG16 -architecture[48]

B. GoogLeNet

The Inception Module is another name for the GoogleNet architecture. The 2014 ILSVRC winner is The Inception Module [48]. Less computation complexity and fewer parameters characterize the Inception block. The GoogLeNet architecture has 22 layers and fully connected layers. It has only 5 million parameters.

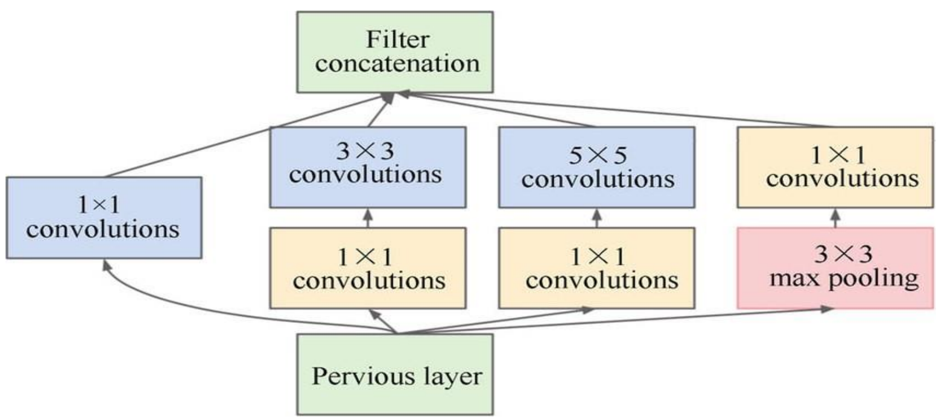



Figure 2. 8 GoogLeNet architecture [48]

C. ResNet50

ResNet50 is a widely-used deep learning architecture, part of the ResNet (Residual Network) family, developed to address the challenges of training very deep neural networks. ResNet consists of layers, primarily composed of residual blocks that allow for the training of exceptionally deep networks without encountering vanishing gradient problems. The key innovation is the inclusion of skip connections, or shortcuts, that enable the gradient to flow more directly through the network during backpropagation [59]. This design facilitates the training of deeper networks, leading to improved accuracy in image classification and other computer vision tasks. ResNet50 has become a popular choice for various applications, especially when a robust and deep feature extractor is required

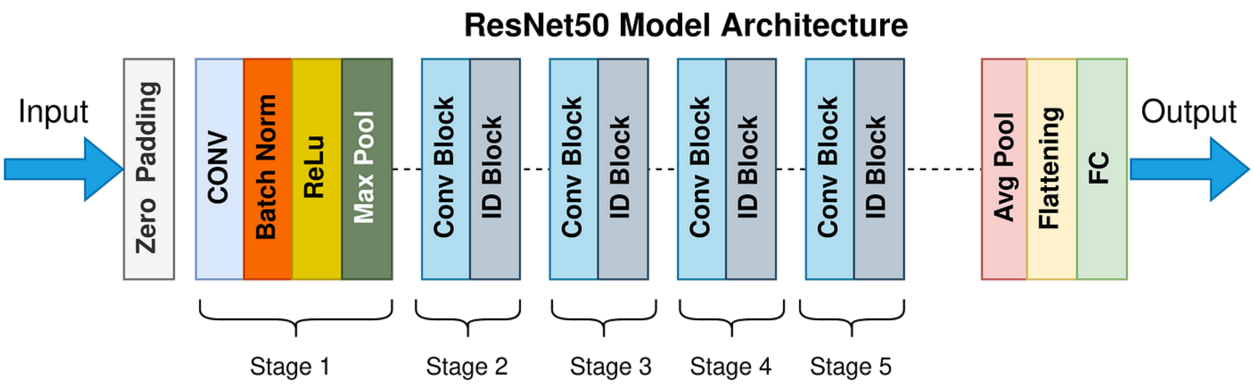



Figure 2. 9 ResNet50 architecture[60]

D. InceptionV3

InceptionV3 is a type of convolutional neural network architecture intended for object detection and classification. It is a member of the Google-developed Inception model family, which has become famous for making effective use of computational resources. InceptionV3 introduces the concept of "Inception modules," which are blocks containing multiple parallel convolutional layers with different filter sizes[59].This allows the network to capture features at various scales. The architecture also incorporates factorized convolutions and global average pooling to reduce the number of parameters and computations. InceptionV3 achieved state-of-the-art performance on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) and is often used for transfer learning in computer vision applications due to its versatility and efficiency in extracting meaningful features from images.

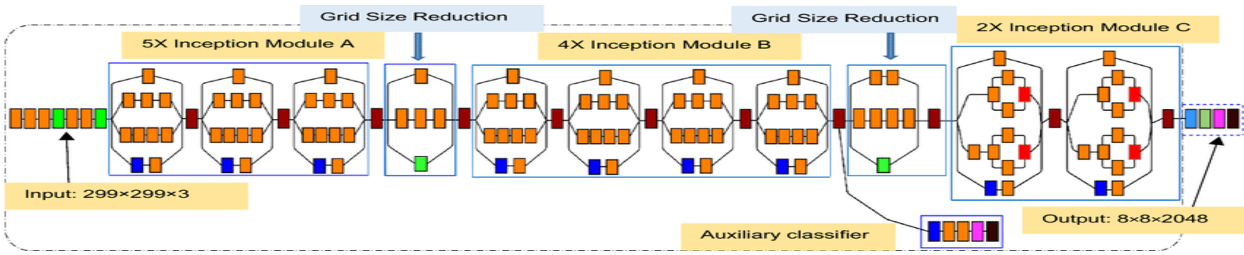



Figure 2. 10 Architecture of InceptionV3

## 2.8. Digital Image Processing

An image can be defined as a two-dimensional function f (x, y), where x and y are spatial (plane) coordinates [26]. We refer to an image as a digital image when it contains finite, discrete values for x, y, and the amplitude value of (f) at any pair of coordinates (x, y) is called the intensity of the grey level of the image at that point.

Mathematically a digital image is a two-dimensional matrix representation made up of a finite number of point cell elements, also known as pixels (picture elements, or pixels). Numerical values are used to represent each pixel in an image. In grayscale images, this is usually limited to a single value that indicates the pixel's intensity, which falls between 0 and 255. In color images, however, three values are stored, each of which represents the amount of red (R), green (G), and blue (B). An image is referred to as binary if it contains just two intensities.

Digital image processing is the process of using a digital computer to process digital images[26]. It is a multidisciplinary field that involves the manipulation of digital images using computer algorithms. Images are captured through devices such as cameras or generated through various means, and digital processing techniques are then applied to enhance, analyze, or extract information from these images. The overarching goal is to improve the visual quality of images, making them more suitable for specific applications across diverse industries. Digital image processing is important in a wide area of mathematical and computational methods to address challenges such as noise reduction, image sharpening, and the extraction of meaningful features, contributing to advancements in fields like medicine, computer vision, and remote sensing.

Most Digital image processing application for image classification follows similar steps such as image acquisition, image preprocessing, segmentation, feature extraction, and classification [26].

### 2.8.1. Image Acquisition

Image acquisition refers to the process of capturing visual information or data from the real world to create a digital image[30]. This process involves the use of imaging devices, such as cameras or sensors, to convert optical information into a digital format that can be stored and processed by computers. The quality and characteristics of the acquired image depend on the properties of the imaging device, the environmental conditions, and the nature of the scene being captured.

Image acquisition is the initial step in the digital image processing pipeline, involving the capture or generation of digital images through devices like cameras and collecting medical data (MRI, CT scanner, ultrasound). Image acquisition sets the foundation for subsequent processing tasks, such as enhancement, restoration, and analysis, and is a critical component in various fields, including medicine, surveillance, astronomy, and computer vision. The quality and characteristics of the acquired images significantly impact the effectiveness of the overall image-processing workflow. before any video or image processing can commence an image must be captured by the camera and converted into a manageable entity [31].

### 2.8.2. Image Preprocessing

Image preprocessing is a crucial step in computer vision and image analysis, involving the application of various techniques to enhance the quality and relevance of digital images before further analysis [32]. This preparatory stage aims to address challenges such as noise reduction, contrast enhancement, and normalization to ensure that subsequent algorithms and models can effectively extract meaningful features. Common preprocessing methods include resizing, cropping, and color normalization, along with more advanced techniques like histogram equalization and edge detection. By optimizing image data through preprocessing, the accuracy, and efficiency of subsequent image-based tasks, such as object recognition or segmentation, are significantly improved [3]

### 2.4.3. Noise Reduction

In image processing and computer vision, noise represents unwanted information which deteriorates image quality [33]. They are unwanted random variations in brightness or color that can degrade the quality of an image. It is often considered an undesirable artifact because it can obscure the true details of the image and reduce its overall clarity. Medical image analysis requires the pre-processing of the image, because, the noise may be added to the scanned images due to imaging device malfunction, signal interference, and patient maladjustment [26].

#### 2.4.3.1.Types of Noise

Knowing the type of noise present in an image is important to remove the noise. The following are types of noise present in an image.

A. Poisson Noise

Poisson noise arises from the random nature of photon arrival in image acquisition processes such as X-ray imaging and astronomical imaging. It is characterized by its variance being equal to its mean intensity. Can be present in a wide range of images, especially those captured under low-light conditions or involving the detection of low-intensity signals. In such scenarios, the random nature of photon arrival or the counting of weak signals can lead to variations in pixel intensity, resulting in the presence of Poisson noise. Studies [34] claimed that Poisson noise is common in Astronomical Images, Biomedical Images, and Microscopy Images

B. Gaussian Noise

It is a type of additive noise that follows a Gaussian distribution. Gaussian noise is commonly encountered in electronic sensors and communication channels[35]. Gaussian noise is often introduced unintentionally during image acquisition or transmission, leading to a degradation in image quality.

C. Salt and Papper Noise

This type of noise appears as randomly occurring white and black pixels, resembling grains of salt and pepper. It often occurs in images captured in low-light conditions or due to transmission errors[36].

D. Speckle Noise

Speckle noise is multiplicity noise that affects homogeneous regions in an image, creating a grainy appearance. It is common in ultrasound and synthetic aperture radar(SAR) images[37].

### 2.8.3. Noise Reduction Technique

In image processing, various techniques are used to reduce noise and enhance image quality.

A. Median Filter

The median filter operates by replacing each pixel in an image with the median value of its local neighborhood [36]. First, for each pixel in the image, a local neighborhood is defined. This neighborhood is typically a square or rectangular region around the pixel. Then pixel intensity values within the local neighborhood are sorted in ascending order. This sorting process arranges the pixel values numerically. After that, the median value is determined by selecting the middle value in the sorted list of pixel intensities. If the neighborhood size is even, the median is usually calculated as the average of the two middle values. Finally, the original pixel value at the center of the neighborhood is then replaced with the calculated median value.

The median filter operates by iteratively going pixel by pixel across the image and substituting the median value of adjacent pixels for each value. The "window" refers to the pattern of neighbors that moves pixel by pixel across the entire image. To calculate the median, first arrange all of the window's pixel values numerically, and then swap out the pixel under consideration for the middle (median) pixel value [26].

B. Gaussian smoothing

Gaussian smoothing is a low-pass filtering technique that reduces Gaussian noise by convolving the image with a Gaussian kernel [35].

C. Non-local Means (NLM) filters

NLM filtering averages similar patches from different parts of the image, effectively reducing various types of noise while preserving image details[38].

### 2.8.4. Scaling (Normalization)

The process of scaling /normalization involves the conversion of such pixel values into a standard format in the range of 0-1. The major advantage of such an approach is that it reduces the time taken to train the deep-learning models[25]. The normalization is achieved using decimal scaling by dividing each pixel value using the value 255 and this is performed for all channels. The ImageDataGenerator function in Keras API has been used to achieve the normalization of the pixel values.

### 2.8.5. Image Resizing

Image resizing is the process of changing the dimensions (width and height) of an image. Resizing an image reduces the computational load during both the training and testing. Smaller images often require less memory and computational resources, enabling more efficient processing.

In the Convolutional Neural Networks (CNNs) model we resize the image, because CNN is a common architecture for image-related tasks, and requires a fixed input size[39]. Resizing ensures that the input to the network is consistent, facilitating the design and training of the neural network.

### 2.8.6. Image Augmentation

Deep Learning needs thousands of datasets for the model performance and handles the problem of overfitting, and class imbalance in training the model [40]. Image augmentation is a crucial technique in deep learning for improving model performance and generalization by artificially expanding the training dataset through transformations applied to the original images. Transformations like rotation, flip, scaling, translation, shearing, and zooming are applied to the existing images [41]. The goal is to introduce diversity and variability into the dataset, allowing machine learning models to generalize better to unseen data.

### 2.8.7. Image Segmentation

In computer vision, image segmentation is the process of dividing a digital image into several meaningful segments or regions according to specific attributes like texture, color, or intensity. Segmentation is the process of separating foreground from background or clustering regions of pixels based on similarities in color or shape [42]. The primary objective of image segmentation is to simplify the representation of an image or make it more meaningful for analysis, enabling computers to understand and interpret visual information effectively [43]. This technique plays a crucial role in various applications, including medical image analysis, object recognition, and scene understanding, as it allows for the isolation and identification of specific objects or structures within an image, facilitating more accurate and detailed analysis in fields where precise delineation of regions of interest is essential.

1. U-Net

U-Net is a convolutional neural network (CNN) architecture designed for image segmentation tasks, particularly in biomedical image segmentation[44]. U-Net is characterized by a U-shaped architecture, featuring a contracting path for capturing context and a symmetric expansive path for precise localization. This network is particularly effective for segmentation tasks where the detailed delineation of objects or structures is crucial.

U-Net architecture works by combining both contracting and expanding paths in a U-shaped configuration, allowing it to capture context at various scales while maintaining fine spatial information.

On the left there is a contracting path(encoder) in which the input image is processed through a series of convolutional layers, downsampling the spatial resolution and increasing the number of feature channels. There is also a max pooling layer that is used to reduce the spatial dimensions while retaining the most salient features. Also, on the right side, there is an expanding path(decoder). The decoder consists of upsampling layers, which increases the spatial resolution of features. Skip connections directly link layers from the contracting path to the expanding path, allowing the network to retain detailed information during the upsampling process. In this architecture the convolutional layers in the expanding path refine the segmented output, capturing both local and global context. The final layer called the output layer employs a softmax activation function to produce a probability map, where each pixel in the input image is assigned a probability of belonging to a particular class or segment.

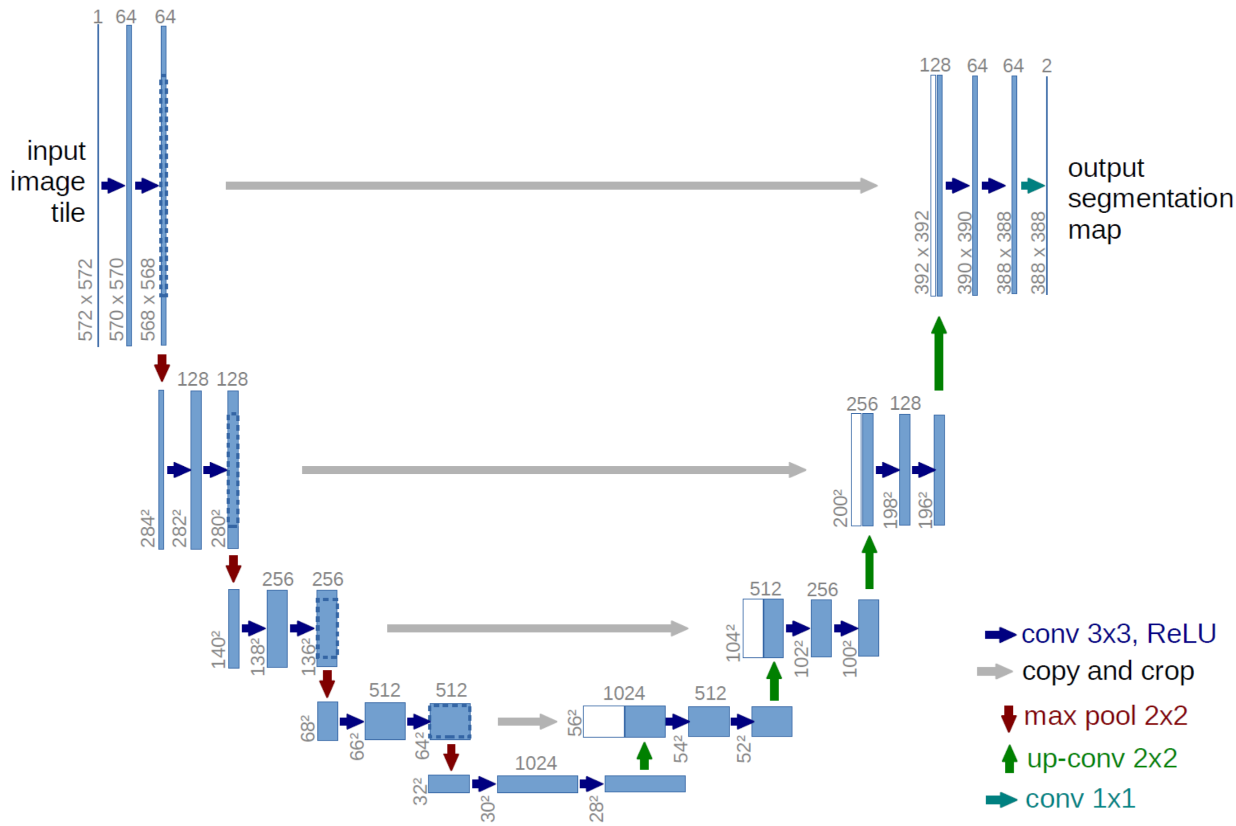



Figure 2. 11 Structure of U-net [45]

2. Watershed image segmentation

Watershed image segmentation is a technique employed in computer vision to partition an image into distinct regions based on the topographical analogy of a watershed [19]. In this method, pixel intensities are treated as elevation values, and flooding is simulated to identify boundaries between different objects or regions. The process begins by identifying markers that represent the initial points of flooding, often derived from intensity gradients or user-defined inputs. As the flooding progresses, watershed lines emerge, delineating the segmented regions. While watershed segmentation is effective in capturing intricate details and separating objects with well-defined boundaries, it may be sensitive to noise and can lead to over-segmentation. Careful marker selection and preprocessing are crucial for achieving accurate and meaningful segmentation results in diverse applications, including medical imaging, remote sensing, and biological image analysis.

3. Threshold-based image segmentation

Threshold-based image segmentation is a fundamental technique in image processing and computer vision, where the goal is to partition an image into regions or objects based on intensity values[46]. The process involves setting a threshold value and classifying each pixel in the image as belonging to either the foreground or background based on whether its intensity is above or below the threshold. First, the suitable threshold value is selected, and each pixel in the image is compared to this threshold. Pixels with intensities above the threshold are classified as foreground, while those below are classified as background.

## 2.9. Related Works

According to the author in [5] the researcher developed kidney stone detection and classification using endoscopy images. The researcher used five 3D-CNN models for kidney stone identification and classification. Less computation complexity and fewer parameters characterize the Inception block. The first three models are used to detect kidney stones; each model has an eight-layer convolutional neural network (CNN-8), while the final two models use a six-layer convolutional neural network (CNN-6) to classify kidney stones. A novel dataset of 1000 images has been collected from various hospitals in Ethiopia. A training set of 0.8 and a testing set of 0.2 were formed from the dataset. The accuracy scores for the 3D-CNN models were 0.985. The limitation of this research work is since the researcher used a deep learning algorithm it requires more amount of data but, small numbers of data images, and is used only for training and testing cases.

In [47] The presence or absence of kidney stones in CT scan images has been determined using four distinct deep-learning techniques. The four algorithms that are employed are InceptionNetV3, ResNet50V2, MobileNetV2, and VGG16. These kidney stone detection models were constructed using a dataset of 1799 kidney CT scan images. A combination of accuracy, precision, and recall metrics was used to evaluate each of the four models' classification performance. For classification, the InceptionNet neural network yielded the best results in terms of recall, accuracy, and precision. It yielded a recall of 0.8331, precision of 0.866, and accuracy of 0.862. These measurements are, respectively, 11%, 10.5%, and 2.9% higher than the corresponding values for the other three models.

An efficient deep learning technique for kidney stone detection is the convolutional neural network.[61]. According to the author, the CNN algorithm has been applied especially to the detection of kidney stones. Unwanted obstructions called ureteric stones develop in the ureters, which are elongated, muscular tubes that carry urine from the kidneys to the bladder. They are about ten meters in length. 465 male and female CT scan images were used to create the proposed model. Using a pixel/voxel-based machine learning approach, the method learns directly from raw data, bypassing the need for extracted features. The connected components approach was used for pre-processing, and the sensitivity was found to be 100%. The connected components approach consists of two steps namely the binarization and then the grouping of all pixels. The primary drawback of this approach is that the model was trained using a dataset that contained no images free of ureteral stones. Due to the absence of significant real-time data subsets in the source data, the performance metric appears to be skewed. The main finding of this study is that it focuses only on ureteral stones, which are more difficult to detect than kidney stones because their target area is smaller. This makes the research particularly noteworthy.

In [62] kidney stone localization and detection from CT images were accomplished using the XResNet-50 deep neural network. Kidney stone detection is accomplished by a cross-residual network, which is comprised of the XResNet-50 algorithm. Furthermore, augmentation methods like rotation and zooming are used to get around the overfitting issue that arises when a model learns to memorize the input data. The parameters of the XRestNet-50 model were also adjusted using the Adam Optimizer and the Cross-Entropy Loss Function. 1800 photos of more than 500 patients in two distinct classes kidney stone and normal with 790 and 1009 photos respectively in each category make up the dataset that was used to train the model. Sensitivity and Specificity, which were determined to be 95% and 97%, respectively, were two metrics used to assess the model's performance. Four normal images were incorrectly classified as having kidney stones by the model. The kind of scan images that were used to model the data is the primary cause of the misclassification. Because the images were of the coronal CT type, the scan also included images of the abdomen, thorax, and pelvis, among other parts. By segmenting the source image data to concentrate more on the kidney region in the source CT scan, the performance of this classification method could have been further improved.

In [63] after speckle noise was removed using the median filter, the Artificial Neural Network algorithm was used to identify kidney stones. Furthermore, to reduce complexity and increase classification accuracy, a specialized optimization algorithm known as the crow search optimization algorithm has been used. One hundred ultrasound pictures make up the dataset that was used to create the model. The model achieved 100% sensitivity, 90% specificity, and 93.45% accuracy in classification. The lack of a large number of images to train the model is the main disadvantage of this approach. Given the overfitting factor and the use of only 100 images, the classification accuracy is questionable.

[2] describes a method for classifying kidney stone composition using a point-of-care image recognition system that combines smartphone microscopy and deep convolutional neural networks (CNNs). Surgically extracted kidney stones were imaged using a smartphone microscope, and a novel CNN architecture was built to classify the stones into four types: calcium oxalate (CaOx), cystine, uric acid (UA), and struvite. The model's performance was evaluated using accuracy, positive predictive value, sensitivity, and F1 scores, achieving high accuracy and F1 scores for each stone type.

Table 2. 1 Summary of related work

| No | Authors | Title | Dataset | Techniques | Gap |
|---|---|---|---|---|---|
| 1 | Yildirim | The deep learning model for automated kidney stone detection | 1800 | XResNet-50 | • images contain another part such as the abdomen, or thorax. |
| 2 | Cui et al. | Automatic Detection and Scoring of Kidney Stones | 625 | Deep Learning and Thresholding Methods | • kidney images with significant anomalies were excluded |
| 3 | Nithya et al. | Kidney disease detection and segmentation | 100 | ANN and multi-kernel k-means clustering | • few datasets |

| | | | | | |
|---|---|---|---|---|---|
| 4 | Kristian M. Black et al. | Deep learning computer vision algorithm for detecting kidney stone | 63 human kidney stones | deep (CNN), ResNet | • only 63 human kidney stones are taken. |
| 5 | Alper Caglayan et .al | Deep learning model-assisted detection of kidney stones on CT | 455 CT images | xResNet50 CNN architecture | • The model trained on a small number of data |
| 6 | Längkvist et .al | Computer-aided detection of ureteral stones in thin slice CT volumes using CNN | 465 male and female CT scan | CNN | • The model was trained using a dataset that contained no images free of ureteral stones. |

In the above table (2.1) there are some gaps considered which are addressed in this study. This study focused on preprocessing techniques, collecting more CT images than the previous study for the robest classification model. While none of the conducted research addresses kidney stone hydronephrosis stage classification.

# CHAPTER THREE

# DESIGN AND METHODOLOGIES

## 3.1. Overview

The goal of this chapter is to give an overall description of this research methodology proposed for the kidney hydronephrosis stage classification model. The study passes through different steps; such as image acquisition, preprocessing, feature extraction, segmentation, and classification of kidney hydronephrosis stage to identify the given CT scanned image based on the disease level type. In this chapter, the research workflow and the general architecture of the system for the kidney hydronephrosis stage classification model are also described. The research process is defined as a framework of strategies and methods chosen by a research worker to mix various components of research in a reasonably logical to solve research problems. For the research to be completed, choosing an appropriate research design methodology is crucial. At this point in the analysis, the choice of research methods is crucial to the quality of the conclusions that will be made from the data. Thus, an experimental research design is chosen. Since the overall approach was numerical and the performance result of the expected outcome from this study is expressed in percentage.

## 3.2. Research Flow

To perform this study, we start by identifying the problem and reviewing articles and previous research that were conducted using deep learning and traditional machine learning related to the study to gain a deep and clear understanding of the research domain. The kidney hydronephrosis stage classification model is proposed using the Deep learning pre-trained model and image processing techniques. Image preprocessing is applied to the image dataset and the proposed deep learning model is applied to the training, testing, and validation datasets.

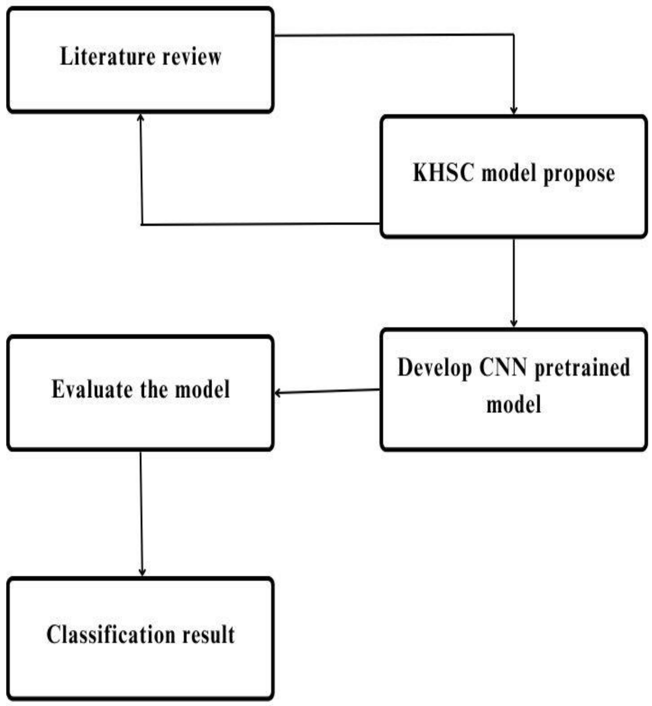



Figure 3. 1 Research flow to build model.

### 3.3. System Architecture

The system architecture model for kidney hydronephrosis stage classification(KHSC) is described below in Figure 3.2. The model shows the hydronephrosis stage classification of kidney stone images using a deep learning pre-trained model.

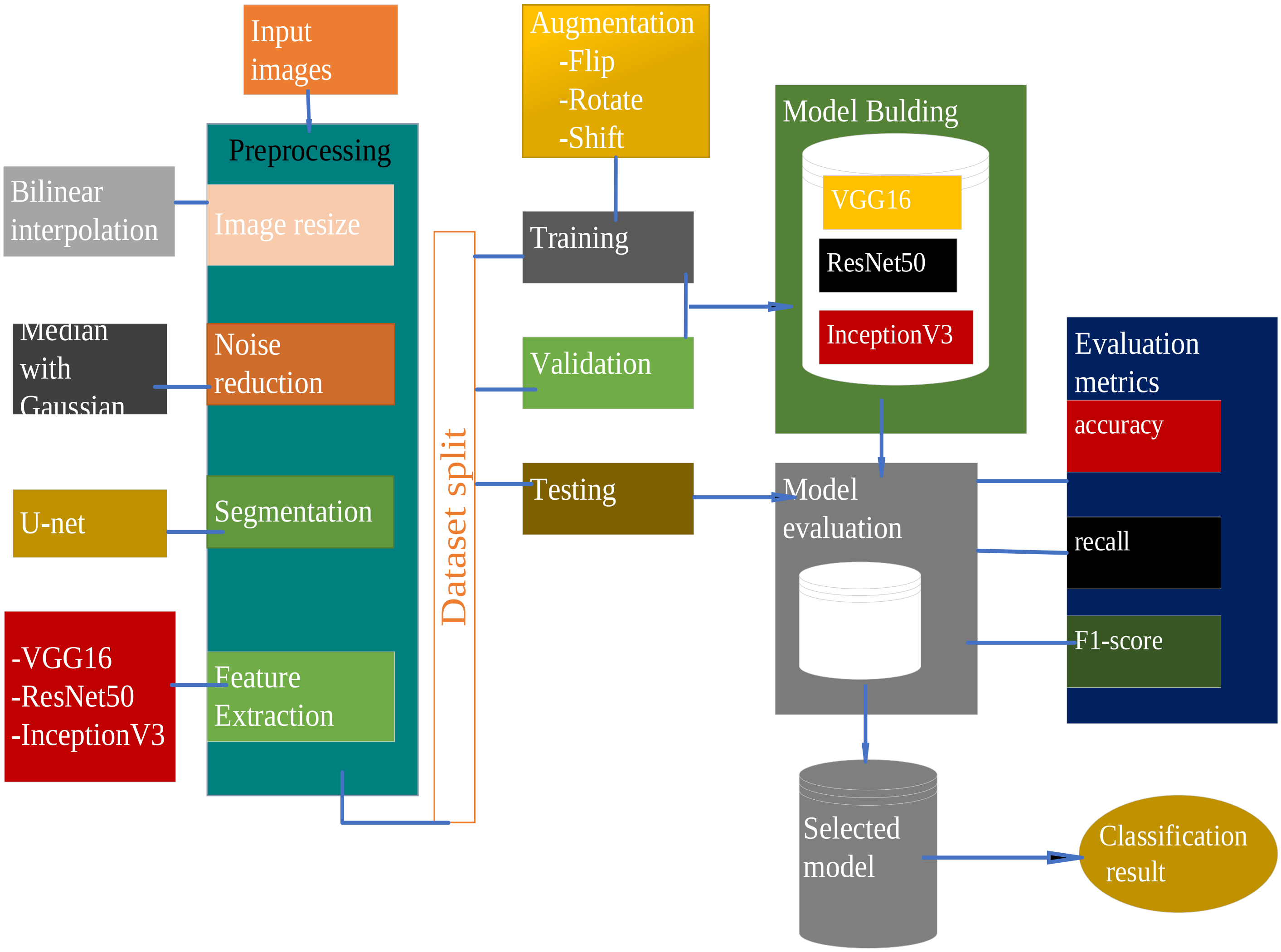



Figure 3. 2 System Architecture

The design typically provides a preview of the entire procedure used to assign a particular input image to one of the four classes. First input images are imported for preprocessing. Preliminary image processing techniques like resizing by bilinear interpolation, noise reduction by Gaussian with median filter, segmentation by u-net, augmentation, and feature extraction are applied to the CT image to increase the accuracy of the model. As in the figure, the system consists of three distinct phases. The training phase, validation, and test phase. Pre-trained model is built and trained on the training and validation dataset, and then the model is evaluated based on model evaluation metrics. After all the best classification model is selected and saved for unseen kidney CT scanned hydronephrosis image dataset.

## 3.4. Domain Expert

In this research, a domain expert guide to collect and label data having more than five years of experience in radiology identified CT scanned images of kidney stone patients from Dilla University Specialized Hospital and Selam General Hospital. The domain experts determine the kidney hydronephrosis stage based on imaging results obtained from patients and are specialized in the radiologist department. Kidney CT scanned images are analyzed by the specialized radiologist to provide the result of the patient which is (mild, moderate, severe, or stone with hydronephrosis).

## 3.5. Image Preprocessing

The collected image has some noise and different sizes therefore needs preprocessing techniques. Selecting the best techniques depends on the types of image application. In this study, different preprocessing techniques such as image resizing, normalization, noise reduction, segmentation, feature extraction, and image augmentation are used as an image preprocessing technique. The techniques used as preprocessing in this study are described as follows.

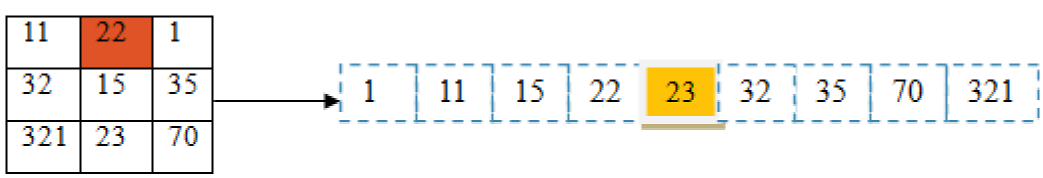
### 3.5.1. Image Acquisition

In this study, different organizations of diagnosis centers located in Dilla Ethiopia for kidney stone hydronephrosis stage classification are asked to get medical images using a letter written from Hawassa University. We presented the objective of images taken from patients and then asked for oral assent and consent before the pictures were taken. We collected kidney hydronephrosis (Mild, Moderate, and Severe) and kidney stones with no hydronephrosis from Dilla University specialized hospital, Selam General Hospital in Dilla City, and the online image data repositories(kaggel.com). CT-scanned images are taken from each class. Besides these image aquation for quantity and quality of classification model image augmentation is applied to the original images.

During image acquisition, the Specialized doctors of the hospitals were guided very well in identifying and labeling the human kidney stone hydronephrosis stage. So, we have collected a total of 4,815 original images from four classes and to increase the number of image datasets we used data augmentation techniques.

### 3.5.2. Noise Reduction: Median Filter with Gaussian

In image processing pipelines, noise reduction plays a vital role by enhancing image quality, improving feature visibility, and ensuring the accuracy and reliability of image analysis results. Different noise filtering techniques are applied based on the noise type on the image. The image dataset in this study is CT medical images, so for such images, there are different noise reduction methods are used, and select the suitable one using image quality evaluation metrics value from experimental results. CT medical images suffer from salt and paper noise leads to very high or low intensity caused by transmission errors or faults in image sensors. Such noise can be handled using a median filter. The median filter is the most common de-noising method used to reduce salt and paper noise and improve the quality of an image while preserving the edges and textures of the images [64]. The median filter is a powerful technique in digital image processing that reduces noise without compromising the sharpness of essential image features. Unlike traditional filters, such as the mean filter, which uses the average of neighboring pixels, it calculates the middle value within a local neighborhood. By doing so, it effectively removes outliers while preserving crucial details in the image. The median filter works by sliding window or kernel moves across the image pixel by pixel. At each position, the pixel intensities within the window are sorted, and the median value is selected as the new intensity for the central pixel. The median value represents the middle value in a sorted list of pixel intensities and is resistant to extreme values, making it an excellent option for noise removal.

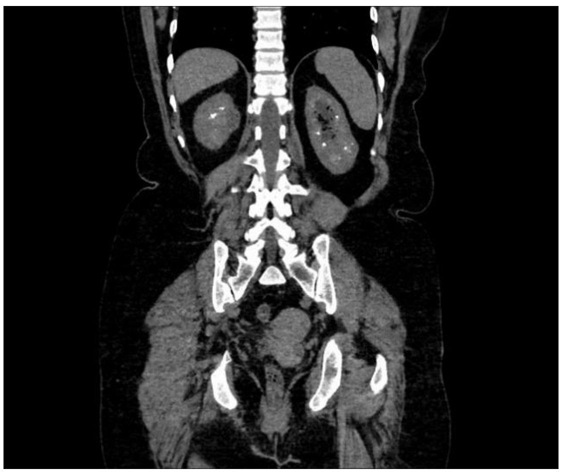



When confronted with irregular extreme noise values like salt-and-pepper noise, the median filter excludes these outliers when determining the median value. As a result, it can effectively address noise that would otherwise interfere with the performance of alternative filtering methods.

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| 1 | 11 | 15 | 22 | 23 | 32 | 35 | 70 | 321 |

### 3.5.3. Image Resize

In this study when we collect datasets from hospital and online image repositories images have different sizes ranging from length of 391 to 512 pixels and width of 320 to 512 pixels and their was size incompatablity. The images are resized to 224 by 224 image width and height of matrix pixels. When we resize an image from larger size to smaller, and from smaller to larger information loss may happen. To remove information loss, we use a technique called interpolation.



Original image

Resized 224 by 224 image

Figure 3. 3 Images after and before resized

### 3.5.4. Image Rescaling/ Normalization

Collected images have different ranges of pixel values. To standardize the input data, making it easier for the model to learn and converge faster during training. We rescale the input images to pixel values in the range of 0 to 1. The normalization is achieved using decimal scaling by dividing each pixel value using the value 255. Normalizing an image is important for reducing the training time.

### 3.5.5. Train-Test Dataset Split

The preprocessed image datasets are arranged into four folders which are mild, moderate, severe, and stone with no hydronephrosis dataset images to train the proposed CNN pre-trained model algorithms. The total number of datasets is 7400 CT images. From this 80% of images are used for training,10% for validation, and 10% for testing, to evaluate the proposed model during training time.

Figure 3. 4 Mild hydronephrosis images

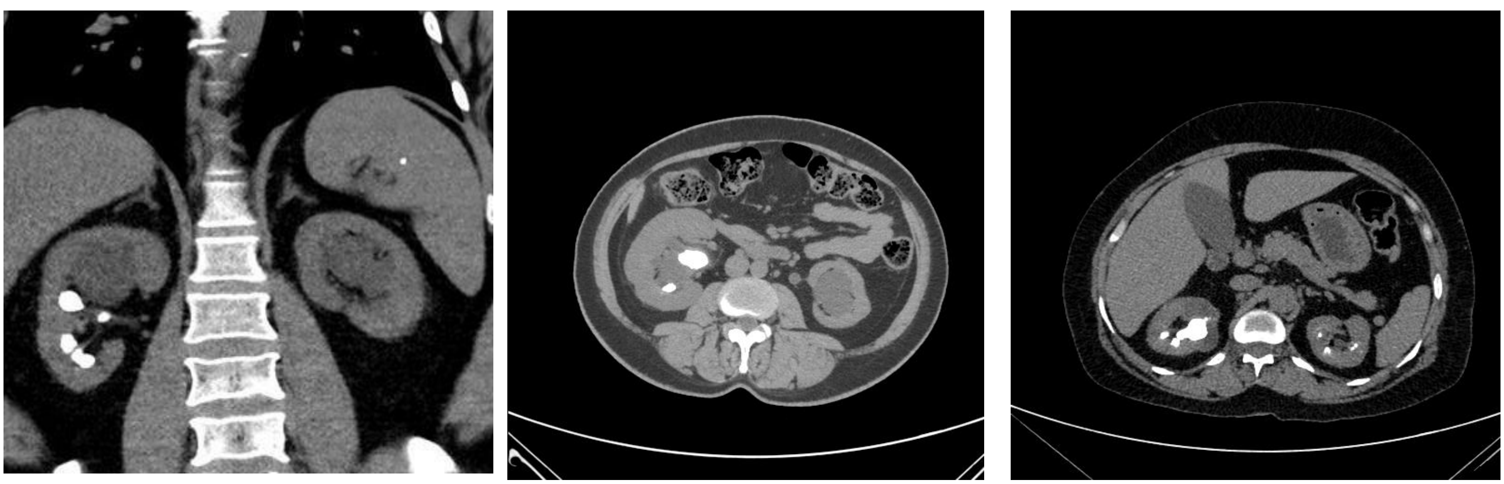


Figure 3.5 Moderate hydronephrosis images

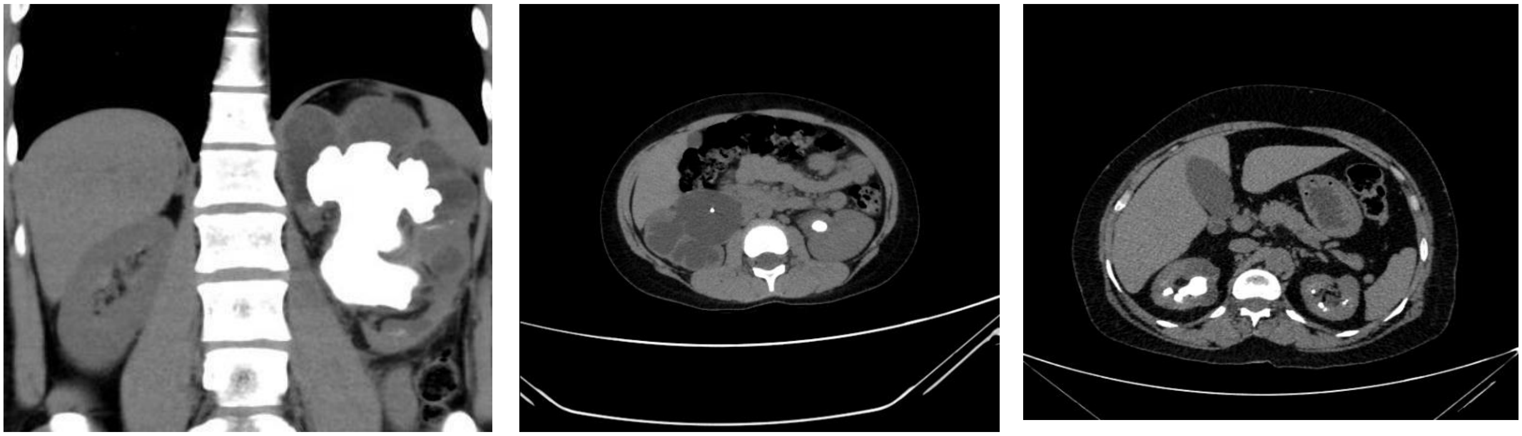


Figure 3. 6 Severe hydronephrosis images

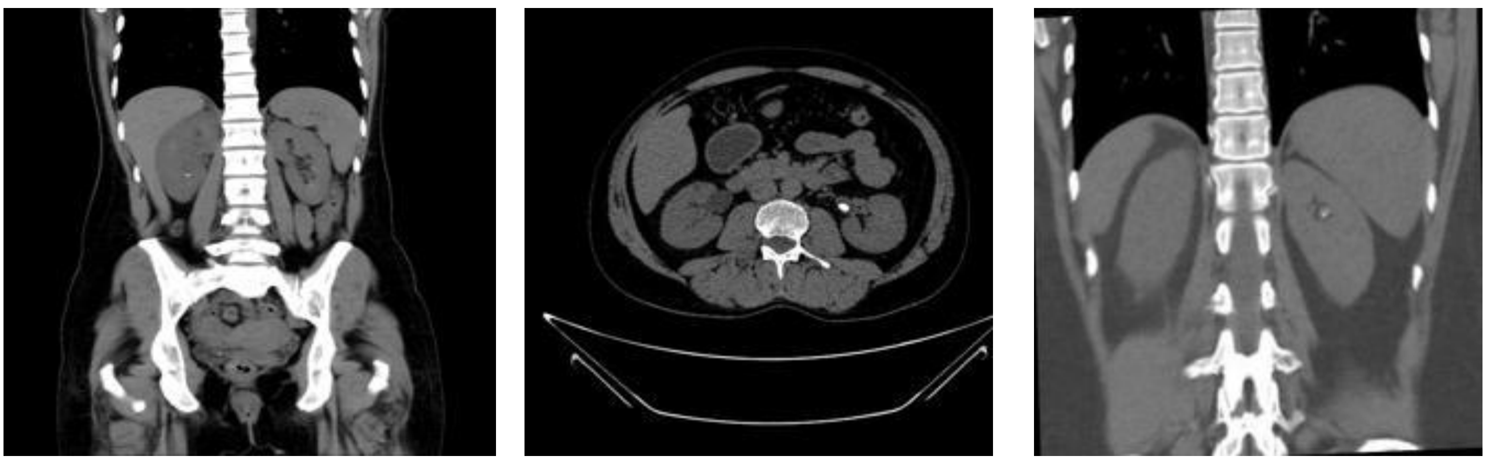


Figure 3. 5 Kideny with stone but no hydronephrosis images.

### 3.5.6. Image Augmentation

In this study to increase the number of image training datasets artificially and to avoid class imbalance augmentation is used. We used a Python library called Keras and ImageDataGenerator, which helps with the transformation of images in different formats. ImageDataGenerator class is used, and transformation like Rotate_range: which is used to randomly rotate the original image from a degree range 0 to 180, Width_shift_range which: shows the infraction of total width and is used for random horizontal shift, Height_shift_range: the same as the Width_shift_range, but it works with height, Zoom_range: which used for random zooming of an image, Horizontal_flip:

used for random flipping of image horizontally, vertical_flip: used for random flipping of image vertically are used are applied to the given images.

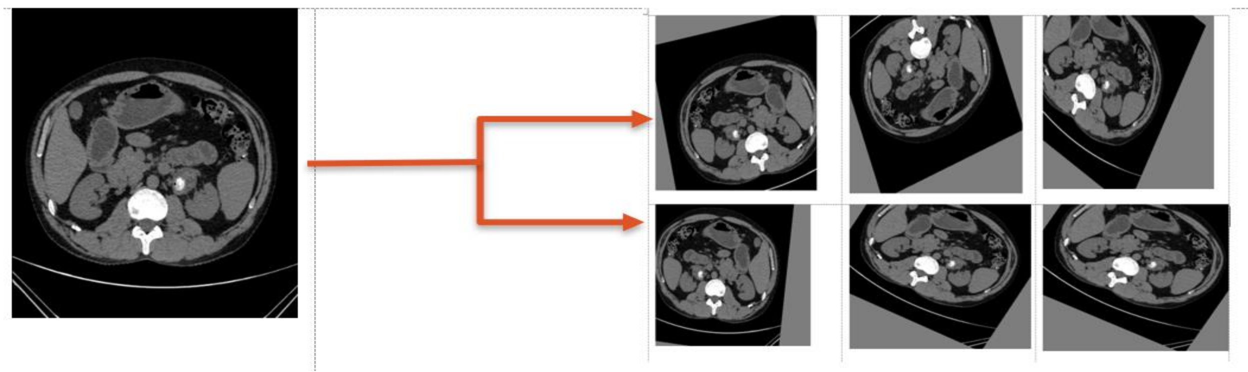


Figure 3. 6 Image Augmentation

### 3.5.7. Image Segmentation

U-Net image segmentation is favored for biomedical datasets due to its hierarchical feature extraction capabilities, preserving spatial information through skip connections, and its adaptability to limited annotated data common in biomedical imaging. Biomedical datasets often encompass diverse modalities, and U-Net's ability to be fine-tuned for different modalities while maintaining state-of-the-art performance makes it a versatile and effective tool for tasks like cell segmentation, organ detection, and various biomedical image analysis tasks. Its track record of achieving high accuracy, even with limited data, and its ability to capture details at different scales contribute to its widespread adoption among researchers and practitioners in the biomedical field.

Original image

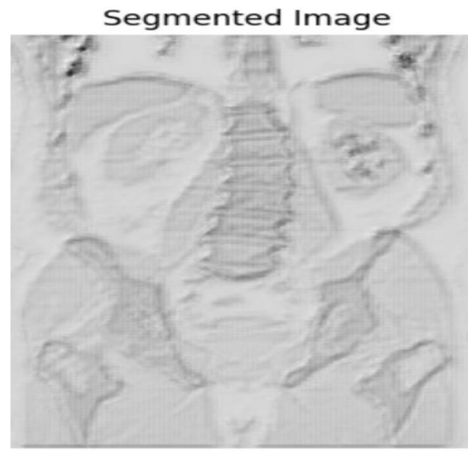



Image segmented by U-net

Figure 3. 7 U-net Image Segmentation

### 3.5.8. Feature Extraction

Convolutional Neural Networks (CNNs) extract features from image datasets by applying learnable filters, also known as kernels or feature detectors, to the input image[65]. Through this convolution operation, the filters identify various patterns and features within the image, including edges, textures, corners, shapes, and object parts. Pretrained CNN models like VGG16, InceptionV3, and ResNet extract features from image data like general CNNs [66]. These pre-trained models follow the standard CNN architecture, which involves convolutional layers, non-linear activation functions, pooling layers, and fully connected layers. However, their key advantage lies in their prior training on extensive datasets, such as ImageNet, which enables them to learn highly meaningful and discriminative features from images.

Pretrained models receive an input image, usually of a predefined size, as their input. The input image undergoes a sequence of convolutional layers, where each layer employs a collection of pre-learned filters [67]. These filters have been trained to recognize diverse patterns and features in the input image, enabling the model to capture a wide range of visual concepts. On the convolutional layer nonlinear activation function is applied to introduce nonlinearity into the network to enhance complex relationships between the features. From the extracted feature less important information is discarded by the pooling layer [68]. After these, the feature maps are flattened into a one-dimensional vector and connected to fully connected layers. These layers, with a large number of neurons, learn to classify the extracted features into different classes or categories. The output layer produces the final predictions or probabilities for the different classes present in the dataset.

**Feature extracted by VGG16:** consists of several convolutional layers stacked on top of each other, with small 3x3 filters and a fixed stride of 1 [69]. This design choice allows VGG16 to learn a large number of filters, capturing both low-level and high-level features in the input image. The depth of VGG16 is what sets it apart, with a total of 16 weight layers, including 13 convolutional layers and 3 fully connected layers.

**Feature extracted by inceptionV3:** the main difference between InceptionV3 and traditional CNNs lies in the design of the inception modules [70]. Inception modules are comprised of multiple parallel convolutional operations with different filter sizes (e.g., 1x1, 3x3, 5x5) and pooling operations. These parallel operations are then concatenated to form a unified set of features. By using a combination of different filter sizes in parallel, InceptionV3 can capture features at various spatial resolutions. The 1x1 filters help reduce the dimensionality of the input, allowing the network to process information more efficiently. The larger filters (3x3 and 5x5) capture more complex and high-level features that span larger regions of the input. Additionally, InceptionV3 incorporates the concept of "bottleneck layers" that further reduce computational complexity [65]. Bottleneck layers use 1x1 convolutions to reduce the number of input channels before applying more computationally expensive operations. This allows InceptionV3 to strike a balance between capturing intricate features and maintaining computational efficiency. Overall, InceptionV3 leverages the power of inception modules with parallel operations, different filter sizes, and bottleneck layers to extract features from images. This architecture enables the network to capture features at multiple scales and levels of abstraction, leading to improved performance in tasks such as image classification and object recognition.

**Feature extracted with ResNet50**: its key distinction from traditional CNN architectures lies in the inclusion of residual connections, which mitigate the problem of vanishing gradients during training [65]. In conventional CNNs, deep networks can encounter challenges with vanishing gradients, meaning the gradients become extremely small as they propagate through the layers. This issue hinders effective learning. ResNet50 tackles this problem by introducing residual connections, also known as skip connections. These connections allow the network to bypass one or more layers and establish direct connections from earlier layers to later ones. By doing so, ResNet50 ensures a smoother flow of gradient signals during training, thereby helping the network learn deep representations more successfully[67].

In this study, we build a trained end-to-end model for kidney hydronephrosis stage classification. Pre-trained models perform feature extraction and classification. CT-scanned image features are extracted by the pre-trained model and fed to the top layer of the model for classification.

## 3.6. Transfer learning

Transfer learning is a machine learning technique where a model developed for a specific task is reused as the starting point for a model on a second, related task[71]. This method employs the knowledge from a large, pre-trained model on a new but related problem. For example, image classification with models like VGG16, ResNet, or InceptionV3 trained on ImageNet. The importance of transfer learning lies in its ability to significantly reduce the time and computational resources required for training, enhance model performance, and achieve better generalization, especially when dealing with limited labeled data. In many medical imaging tasks, such as kidney hydronephrosis stage classification, collecting a large annotated dataset can be challenging and time-consuming.

## 3.7. Software Tools

1. TensorFlow

TensorFlow is the most famous library used in production for deep learning models. TensorFlow allows users to define, train, and deploy deep learning models efficiently, supporting a range of applications such as image and speech recognition, natural language processing, and more. TensorFlow has played a pivotal role in advancing the field of deep learning, providing researchers, developers, and data scientists with a powerful and accessible tool for creating and deploying sophisticated AI models. It implements standard mathematical operations on tensors, as well as many operations specialized for machine learning, and supports Multidimensional-array-based numeric computation, GPU and distributed processing, Automatic differentiation, Model construction, training, and export. It has a lot of classes Modules and functions [72].

2. Keras

Keras is a high-level API built on TensorFlow (also on Theano). It is more user-friendly than TensorFlow and allows researchers to quickly build and test networks with minimal effort. Keras is an open-source deep-learning library written in Python that serves as a high-level neural networks application programming interface (API). Originally developed as a user-friendly interface on top of other deep learning frameworks, Keras has become an integral part of TensorFlow since TensorFlow version 2.0. One of Keras' key strengths lies in its simplicity and user-friendly design, allowing both beginners and experienced researchers to quickly prototype and build neural network models. It provides a high-level abstraction for defining and training complex neural networks, enabling users to focus on model architecture and experimentation rather than low-level implementation details [72].

3. Google Colab

Google Colab, short for Google Colaboratory, is a cloud-based platform that provides a free and accessible environment for running and developing machine learning and data science projects. Hosted on Google Drive, Colab allows users to create and share Colab Notebooks which are interactive documents combining code, text, and visualizations. One significant advantage of Google Colab is its provision of free resources, enabling faster computations and more efficient training of machine learning models. This feature is particularly beneficial for researchers who have access to high-performance hardware locally.

Colab’s Notebook looks like the famous Jupyter Notebook with an additional bonus feature:

- Access to GPU and
- Stored data in Google Drive
- Has a huge library for machine learning, and deep learning.

4. Mendeley

Mendeley is a comprehensive reference management tool that facilitates efficient organization, collaboration, and citation of academic research materials. Mendeley allows users to create a centralized digital library where they can store, organize, and annotate scholarly articles, papers, and other reference materials. One of Mendeley's key features is its ability to automatically extract metadata from imported documents, simplifying the process of adding and categorizing references. Furthermore, Mendeley offers powerful citation management capabilities. It integrates with popular word processing software, such as Microsoft Word and LaTeX, allowing users to easily insert citations and generate bibliographies in various citation styles. The platform also provides a plugin for web browsers, making it convenient to save and organize references directly from online sources. Mendeley's user-friendly interface, extensive database, and collaborative features make it a valuable tool for researchers, academics, and students alike, streamlining the often-complex process of managing and citing references in scholarly work.

### 3.8. Model Building

After all the proposed image preprocessing is applied in the kidney CT medical dataset classification model is developed using different convolutional neural network deep learning architectures.

1. VGG16 Model

Pretrained models are typically trained on large datasets like ImageNet and have learned rich feature representations. In this case, we select VGG16 and we load the pre-trained model without the top layers (i.e., the fully connected layers used for classification). Then we Freeze the layers of the pre-trained model to prevent them from being trained again during feature extraction. Based on our interest we added dropout, Batch normalization, and new layers on top of the pre-trained model to perform feature extraction or classification based on your specific task.

2. ResNet50 Model

Pretrained models are typically trained on large datasets like ImageNet and have learned rich feature representations. In this case, we select ResNet50 and we load the pre-trained model without the top layers (i.e., the fully connected layers used for classification). Then we Freeze the layers of the pre-trained model to prevent them from being trained again during feature extraction. Based on our interest we added dropout, Batch normalization, and new layers on top of the pre-trained model to perform feature extraction or classification based on your specific task.

3. InceptionV3 Model

In this case, we select InceptionV3 and we load the pre-trained model without the top layers (i.e., the fully connected layers used for classification). Then we Freeze the layers of the pre-trained model to prevent them from being trained again during feature extraction. Based on our interest we added dropout, Batch normalization, and new layers on top of the pre-trained model to perform feature extraction or classification based on your specific task.

### 3.9. Image Quality Metrics

1. Mean Square Error (MSE)

The MSE (Mean Square Error) subtracts the input image from the output images to determine the image quality. It calculates the square errors of the input images and output images. Mathematically it is described by the equation

$$\text{MSE}=\frac{1}{mn}\sum_{=0}^{m-1}\sum_{j=0}^{n-1}[A(i,j)-B[I,J)]\,2 \qquad \text{……………………..…………………(3. 1)}$$

Where:
mn is the dimension of the image
A (i, j) and B(i,j) represent the pixel values at location (i,j)in the original (reference) image and the distorted image, respectively.

2. Peak to signal noise ratio (PSNR)

Peak to-signal noise ratio evaluates the quality of the images between the input and the reconstruction images. Mathematically it is described by the equation.

$$\text{PSNR} = \quad 10\log\left(\frac{MAX2}{MSE}\right) \qquad \text{……..…………….…………..……………………(3. 2)}$$

Where MAX represents the maximum possible pixel value of the image

### 3.10. Performance Measurement Parameters

The comparison of classification is done based on classification performance parameters. The performance measurement parameters are computed from the confusion matrix. This study model performance is measured using the following performance measurement parameters.

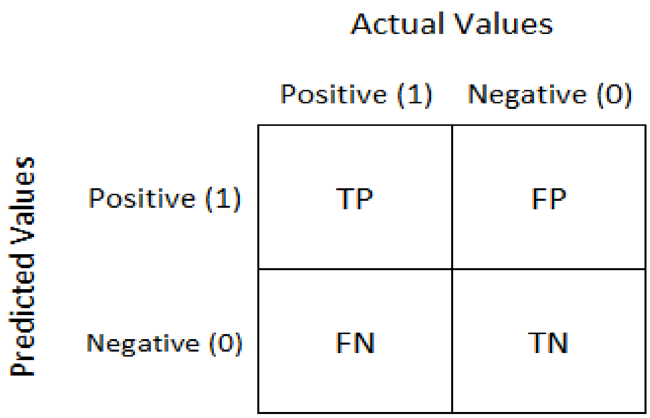



Figure 3. 8 Confusion matrix[73]

True positive(TP): refers to a case where the model correctly predicts a positive class for an image, and the actual ground truth label for that image is also positive.

False positive(FP): in image classification refers to a situation where the model incorrectly predicts a positive outcome when the actual outcome is negative.

False negative(FN): in image classification occurs when the model incorrectly predicts a negative outcome when the actual outcome is positive

True negative (TN): refers to a correctly predicted negative instance.

Precision: measures the proportion of true positive predictions among all positive predictions made by the model. It focuses on the correctness of positive predictions.

Mathematically Precision= $\frac{TP}{TP+FP}$ ……………………………………………………………(3.3)

Recall: measures the proportion of true positive predictions among all actual positives in the dataset. It is all about correct results. It focuses on the model's ability to correctly identify positive instances.

Mathematically Recall=$\frac{TP}{TP+FN}$ ……………………………………………………….(3.4)

Accuracy: measures the overall correctness of the model's predictions. It is calculated as the ratio of correctly classified samples to the total number of samples.

Mathematically Accuracy=$\frac{TP+TN}{TP+FP+TN+FN}$ ……………………..………………..………….(3.5)

# CHAPTER FOUR

# EXPERIMENT AND RESULT DISCUSSION

## 4.1. Overview

This chapter discusses the detailed implementation procedure, dataset preparation, and experimental results on kidney hydronephrosis stage classification at the training, validation, and testing dataset. This section provides an explanation and presentation of the developed model's results using the study's pre-trained model. So, different experiment results are explored for the proposed CNN pre-trained models. The best performance model is selected from the CNN pre-trained models (InceptionNet, ResNet50, and VGG 16) models.

## 4.2.Dataset Preparation

In this study, we collect datasets (mild, moderate, severe, and kidney stone with no hydronephrosis) for training, validation, and testing datasets from the Dilla University referral hospital, Selam General Hospital at Dilla City, and the Kaggle online free dataset repository. Images are labeled by domain experts(radiologists). The kidney hydronephrosis stage classification model is developed by conducting different experiments. Experiments are conducted starting from the preprocessing stage to the model development and evaluation step. Finally, the best result is selected based on model selection evaluation criteria.

### 4.2.1. Image Preprocessing

**Noise reduction**

CT-scanned images are subjected to salt and paper noise as well as Gaussian noise. In our study we conduct Gaussian smoothing, non-local means denoising, median filter, and adaptive median with Gaussian filtering techniques. The results of the filtered images are compared with image evaluation metrics value by PSNR and MSE. The results of the evaluation are represented in the table below.

Table 4. 1 Comparison of image  quality after noise reduction

| Evaluation metrics | Gaussian | mean | median | Gaussian with median |
|---|---|---|---|---|
| PSNR | 25.36 | 22.06 | 23.57 | 24.16 |
| MSE | 189.15 | 404.82 | 285.80 | 213.73 |

As shown in the above table the noise removal technique image quality we get good results on Gaussian and median filter but we have selected Gaussian with median filter because of the nature of our CT image.

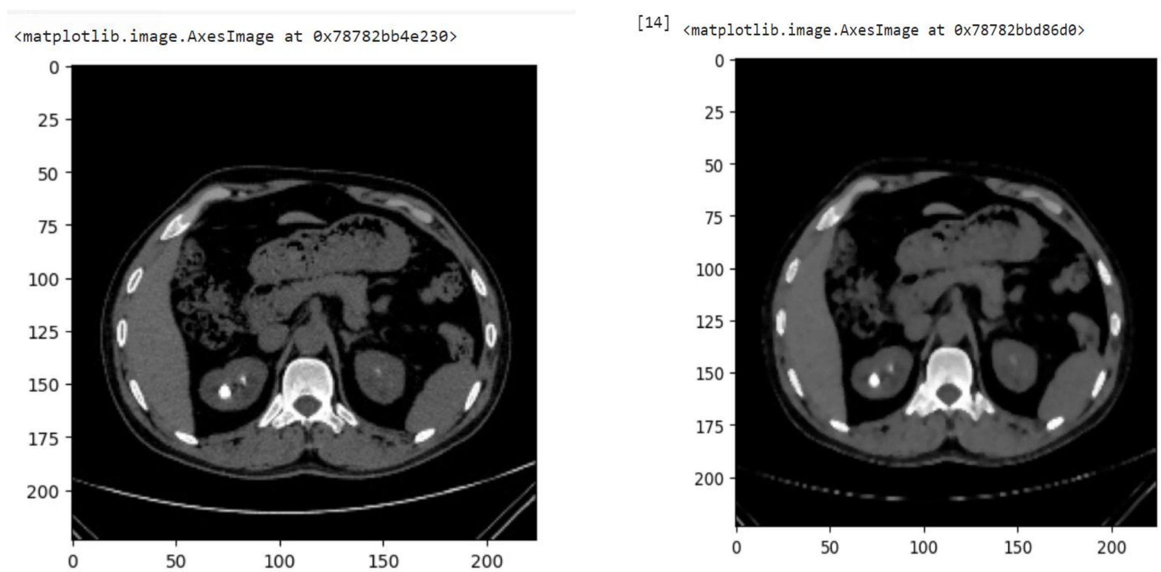



Figure 4. 1 Comparison of the original and image filtered by the median with Gaussian filter.

**Image Resizing**

In our study collected images have different sizes ranging from the length of 391 to 512 pixels and width of 320 to 512 pixels. Since the pre-trained model requires a fixed image size of 224 by 224 images are resized to 224 by 224 image width and height of matrix pixels. We have applied four interpolation techniques to resize the given collected image. These techniques are the Bicubic, Bilinear, nearest neighbor, and Lanczos interpolation techniques with an aspect ratio of 0.87. The result of the resized images is selected based on image quality evaluation metrics value. we have selected bilinear interpolation techniques according to the similarity of the resized image to the original image.

**Augmentation**

To overcome the problem of class imbalance and limited dataset we use an augmentation technique. Augmentation is applied to the mild, moderate, severe, and kidney stones with no hydronephrosis class of the training dataset with the following augmentation parameters and their corresponding value.

Table 4. 2 Image augmentation parameters and values

| **Augmentation Parameter** | **Value** |
|---|---|
| vertical_flip | True |
| horizontal_flip | True |
| zoom_range | 0.2 |
| rotation_range | 45 |
| Height_shift_range | 0.2 |
| Width_shift_range | 0.2 |

During the augmentation process, a total of 2585 images are augmented from a four-class image dataset using the above list of augmentation parameter techniques in Table(4.2). After augmentation total of 7400 image datasets are stored for the classification model development. Also, the collected images from online and offline have different image extensions(PNG, JPEG). we change the image types of all images to JPEG image types to remove incompatibility.

Table 4. 3 Number of Original images and images generated by augmentation

| Class name | Number of original images | Number of augmented images | Image extension |
|---|---|---|---|
| Mild | 1210 | 640 | JPEG |
| Moderate | 1158 | 692 | JPEG |
| Sever | 927 | 923 | JPEG |
| No-hydronephrosis | 1520 | 330 | JPEG |
| Total image | 4815 | 2585 | |

**Image Segmentation**

In our study, for segmenting the images u-net-based image segmentation is applied after a comparison of different segmentation techniques such as threshold-based image, and watershed segmentation techniques. From the three segmentation techniques in this classification model development u-net-based segmentation performed best according to the characteristics of the CT kidney stone medical images.

## 4.3. Image Dataset Split Ratio

From the total collected 7400 images, we have trained the model using three split ratios. The splitting techniques are 10%-10%-80%, 10%-20%-70%, and 15%-15%-70% of the total image data for testing, validation, and training respectively. The number of instances in each ratio is represented in the table(4.4) below. The number of instances in each class differs according to the ratio listed in the illustrated table below. The model was developed based on the split ratio to select the best one from the list depending on the experimental result which tells the number of correct classifications.

Table 4. 4 Dataset Partition

| Split-Ratio(test, validation, train) | Number of instances (test, validation, train) | Total Number of images |
|---|---|---|
| 10%-10%-80%, | 740,740,5920 | 7400 |
| 10%-20%-70%, | 740,1480, 5180 | |
| 15%-15%-70% | 1110,1110,5180 | |

## 4.4. Hyper-parameter Setup

Hyperparameters are a parameter in deep learning whose value is set before the learning process starts. There are no standard rules to choose the best hyperparameters for the given problems. The hyperparameters are chosen by trial and error by conducting a different experiment on the Proposed CNN pre-trained model that gives the best performance of the models. To optimize the model's performance, a hyper-parameter setup process was conducted. The techniques compare and set the optimal parameter values as shown in Table (4.5) below.

Table 4. 5 layers and parameters added to the pre-trained model.

| No. | Hyperparameter | Value |
|---|---|---|
| 1 | Epoch | 20,30,50 |
| 2 | Batch Size | 16, 32,64 |
| 3 | Activation Function | Relu for feature extraction, SoftMax for classification |
| 4 | Unit | 256,512,1024 |
| 5 | Learning rate | 0.001,0.01 |
| 6 | Dropout | 0.3, 0.5, 0.8 |
| 7 | Optimizer | Adam |

## 4.5. Experiment and Results of the Proposed Pre-trained Models

In this study, different experiments are performed by freezing and fine-tuning different layers of the three pre-trained models. Freezing a pre-trained model prevents the weights of its layers from being updated during training. The pre-trained layers retain their learned features and are used as fixed feature extractors. To retain the extracted features layers are frozen. The following figure shows the general architecture of Freezing and fine-tuning layers in our pre-trained models.

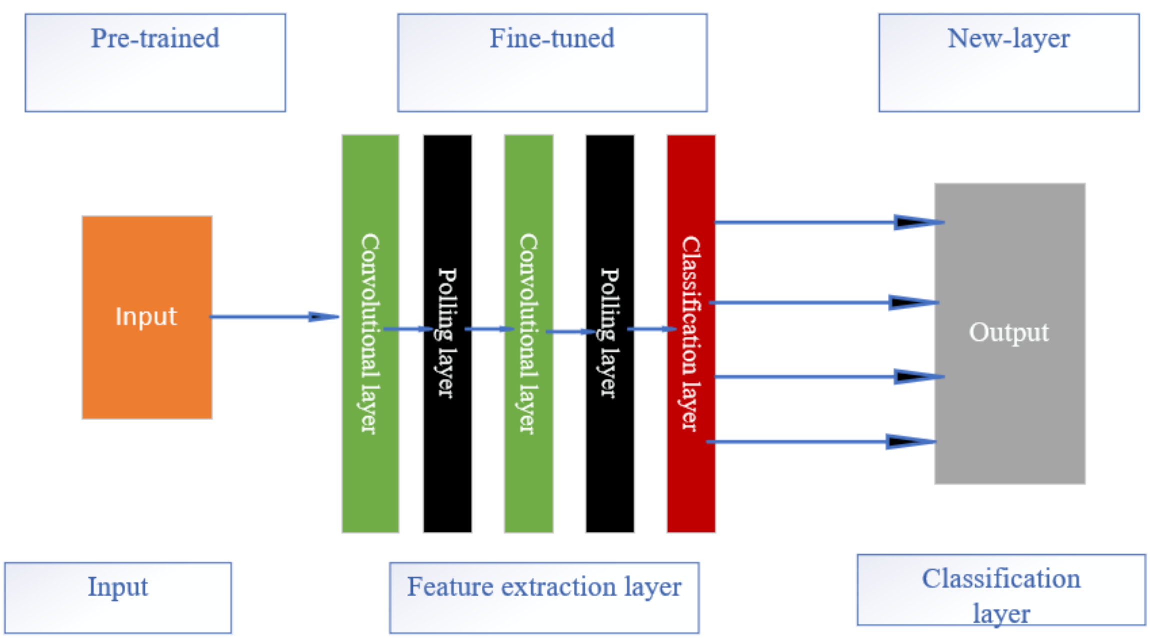



Figure 4. 2 Fine-tuned architecture

Then to make classification on the kidney CT scanned input image dataset new layers are added.

### 4.5.1. VGG16

The pre-trained VGG16 model has 13 convolutional layers, 5 pooling layers, and 3 fully connected layers, and also has 138 million parameters. The model is modified because it has 138 million parameters that train on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) and requires a lot of image datasets and 1000 classes, requiring large memory storage and computational powers. Images are converted to RGB image format because VGG16 pre-trained models require RGB image format with dimensions of $224 \times 224 \times 3$. However, in this study, the number of image datasets is limited to four classes, so, it needs the output layer reshaped to four layers for classification with a softmax activation function.

In our thesis, the VGG16 model without the fully connected layers has a feature map size of 7x7x512 after the final convolutional layers. The GlobalAveragePooling2D layer then computes the average of each feature map across its spatial dimensions, resulting in a feature vector with a dimension of 512. The feature vector dimension for the VGG16 model in our model is 512, representing the output of the GlobalAveragePooling2D layer applied to the feature maps produced by the VGG16 convolutional layers.

The following figure(4.3) shows a snapshot of the experiment conducted using the optimizer parameter ADAM, freezing all layers and making classification on the last block.

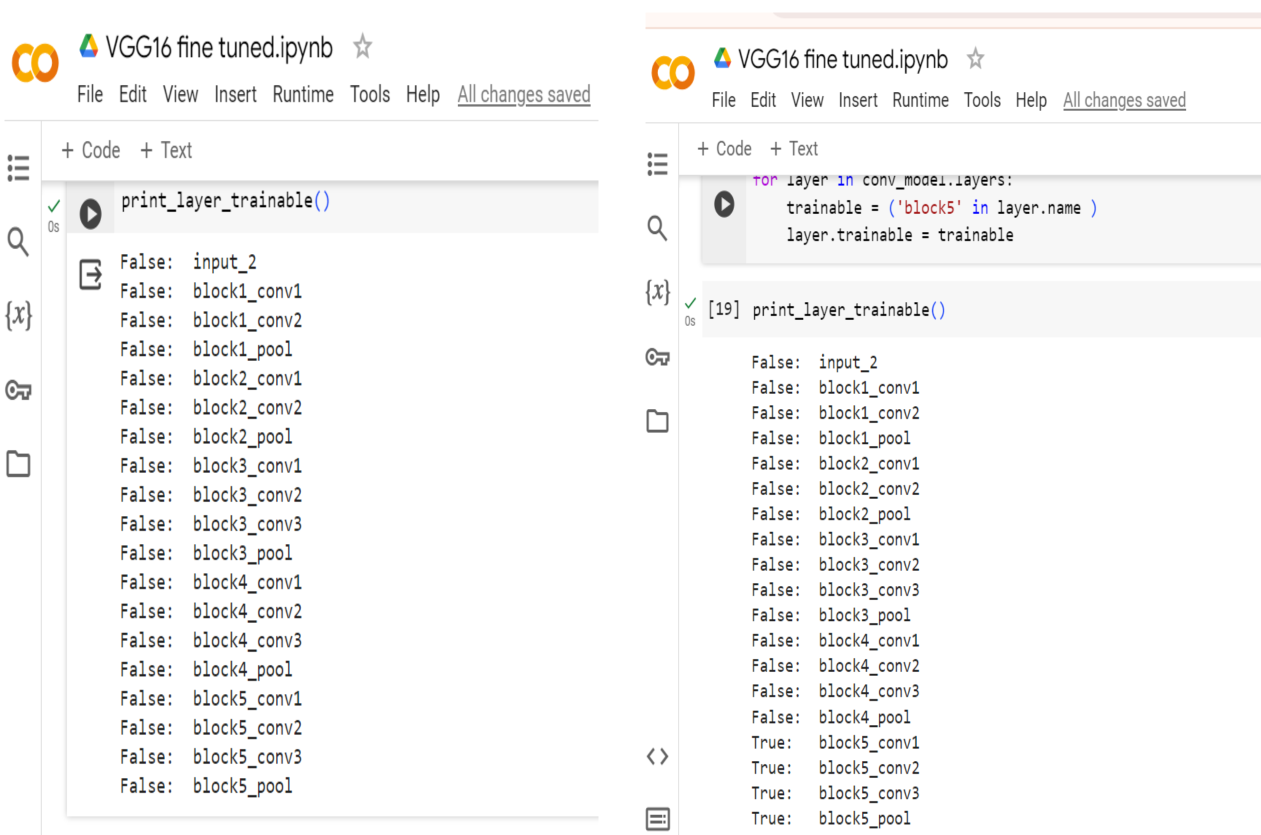



Figure 4. 3 Freezing all layers of VGG16 and making classification on the last block.

The model is trained with four epoch values(20, 30, 40) with bach size (16, 32, 64) from the trained iteration 40 epochs with 32 batch size has a better result. On such bach and iteration, the model trained with Adam optimizer and learning rate (0.01, 0.001, 0.0001) with a split ratio of 10%-10%-80%, 10%-20%-70%, and 15%-15%-70%. The model achieved good classification results on a 0.001 learning rate of 10%-10%-80% dataset split ratio.

The classification model achieved 91% train accuracy, 83% validation accuracy, and 86% test accuracy with a loss value of 0.6. Figure (4.4)below shows the train and validation accuracy-loss graph.

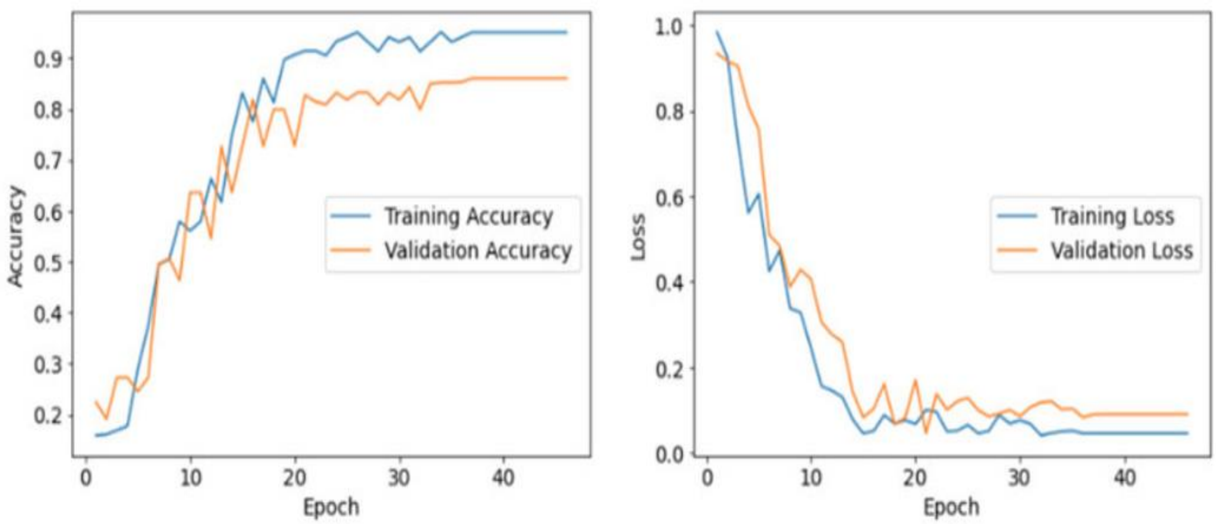



Figure 4. 4 Train and validation accuracy -loss graph for VGG16.

The train and validation accuracy increases when the number of epochs increases. The validation accuracy is smaller than train accuracy after 30 iterations but till this epoch, the two tend to same values as indicated in the graph and overlap each other. The loss graph also represents the same report because it the complementary to the accuracy graph. The accuracy and validation are far away when the iteration increases.

The model classification report on the test dataset is represented on the following confusion matrix as shown in (4.5) below.

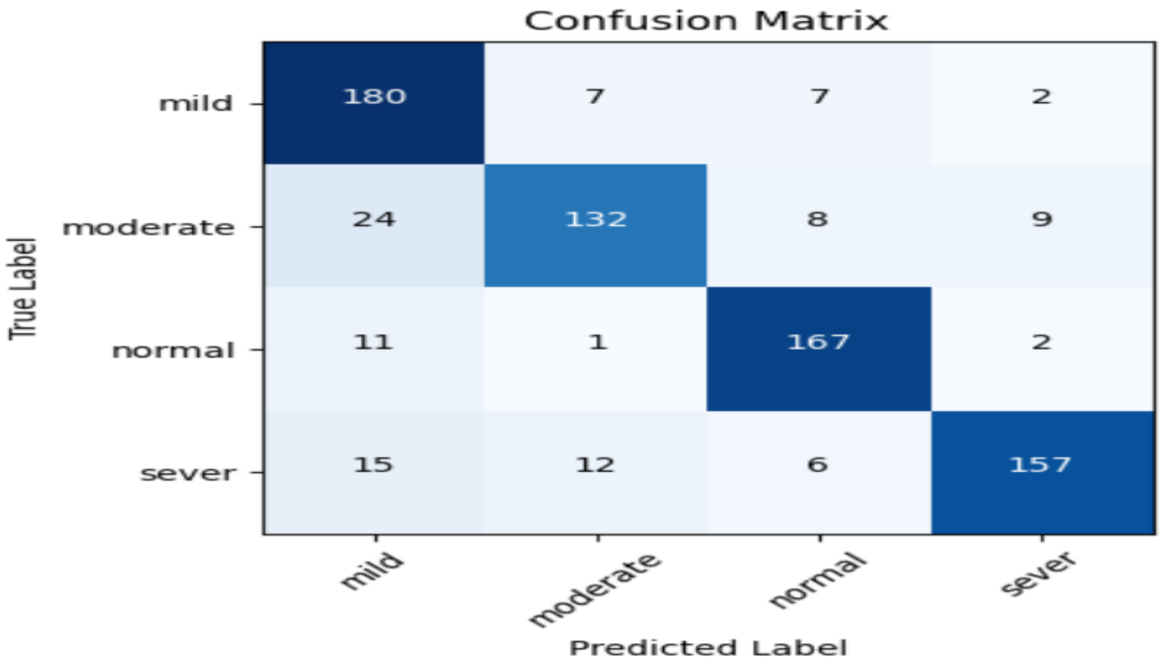



Figure 4. 5 Confusion matrix for VGG16.

As the figure above illustrates from the total number of 740 test sets 636 instances were correctly classified on the correct classes while the remaining 104 instances were misclassified. In the test set the number of instances correctly classified in each class are different. The model most probably did not correctly learn specifically the severe and mild classes.

### 4.5.2. ResNet50

In this experiment, the ResNet50 base model is obtained using the Keras API, and new layers are added to the base model. The model is trained with four epoch values(10,30, 35) with bach size (16, 32, 64) from the trained iteration 35 epochs with 32 batch size has a better result. On such bach and iteration, the model trained with Adam optimizer and learning rate (0.01, 0.001, 0.0001) with a split ratio of 10%-10%-80%, 10%-20%-70%, and 15%-15%-70%. The model achieved good classification results on a 0.001 learning rate of 10%-10%-80% dataset split ratio.

The ResNet50 model without the fully connected layers has a feature map size that depends on the input image size and the architecture of the ResNet50 model. The dimension of the feature vector in our ResNet50 model is the number of channels in the last convolutional layer of ResNet50, which is typically 2048. This means the feature vector would have a dimension of 2048, capturing high-level features extracted by the ResNet50 mode

The classification model achieved 79% train accuracy, 73% validation accuracy, and 81% test accuracy with a loss value of 0.58. Figure (4.6)below shows the train and validation accuracy-loss graph.

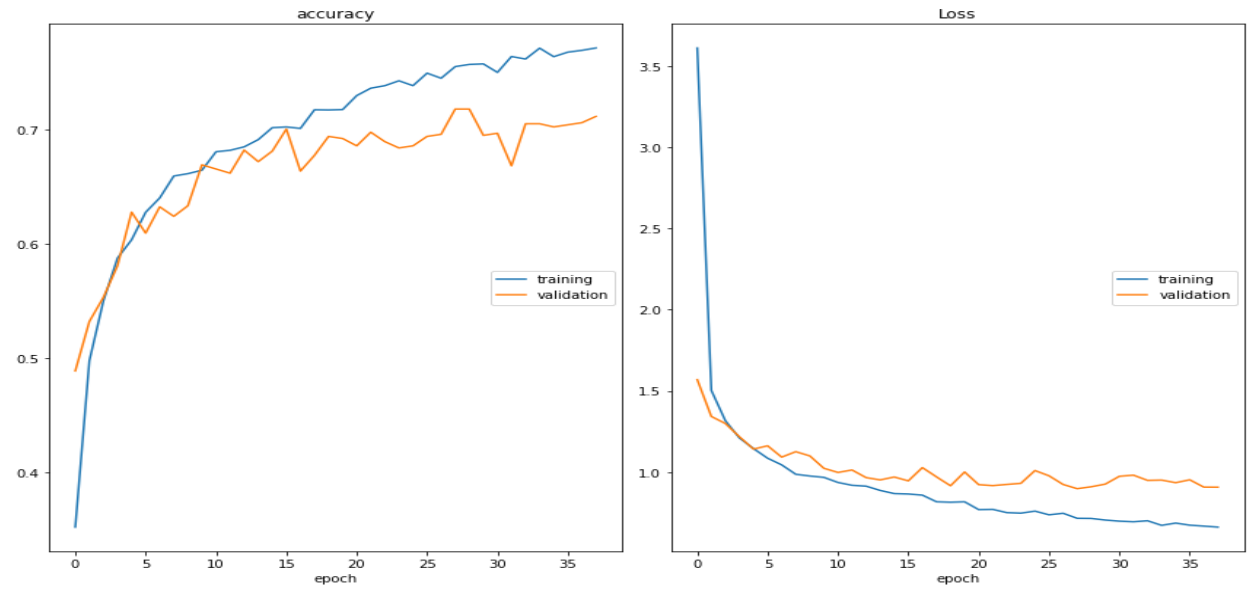



Figure 4. 6 Train and validation accuracy -loss graph  for ResNet 50

As shown in the above figure (4.6) the gap between validation and train accuracy is higher in the iteration from  10 to 35 compared to the other iteration

The model classification report on the test dataset is represented on the following confusion matrix as shown in Figure (4.7) below.

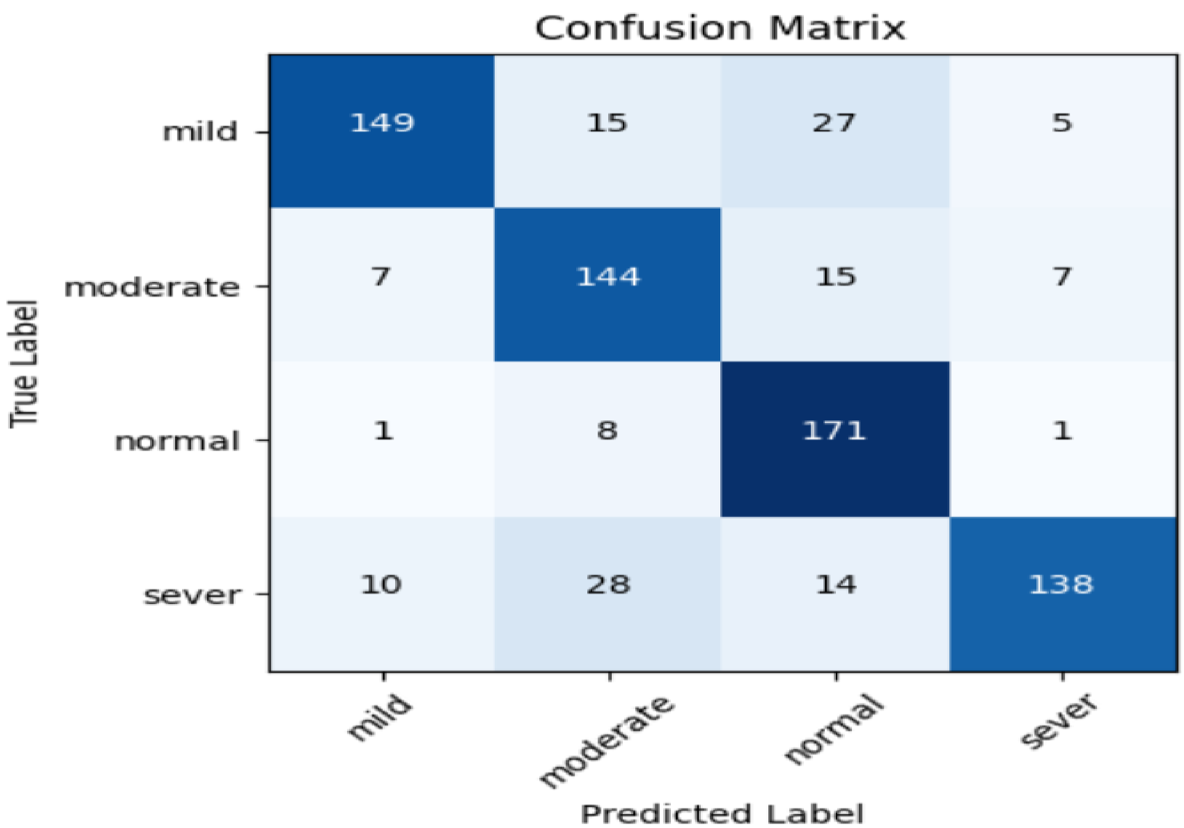



Figure 4. 7 Confusion matrif for ResNet50

As the figure above illustrates from the total number of 740 test sets 602 instances were correctly classified on the correct classes while the remaining 138 instances were misclassified.

### 4.5.3. InceptionV3

In this experiment, the InceptionV3 base model is obtained using the Keras API, and new layers are added to the base model. This model typically has a GlobalAveragePooling2D layer with 2048 channels before the classification layers. Therefore, the feature vector dimension of 2048. The model is trained with four epoch values(20, 30, 50)  with bach size (16, 32, 64) from the trained iteration 100 epochs with 32 batch size have a better result. On such bach and iteration, the model trained with Adam optimizer and learning rate (0.01, 0.001, 0.0001) with a split ratio of 10%-10%-80%, 10%-20%-70%, and 15%-15%-70%. The model achieved good classification results on a 0.001 learning rate of 10%-10%-80% dataset split ratio.

In the InceptionV3 model, the feature vector dimension depends on the architecture of the InceptionV3 model and the layer at which the feature extraction is performed. In Our InceptionV3 model, the last layer before the classification layers is the GlobalAveragePooling2D layer, which outputs a feature vector with a dimensionality equal to the number of channels in that layer.

Our InceptionV3 model typically has a GlobalAveragePooling2D layer with 2048 channels before the classification layers. Therefore, the feature vector dimension in this case would be 2048

The classification model achieved 63.46% train accuracy, 63.65% validation accuracy, and 65.54% test accuracy with a loss value of 0. 8429. Figure (4.8)below shows the train and validation accuracy-loss graph.

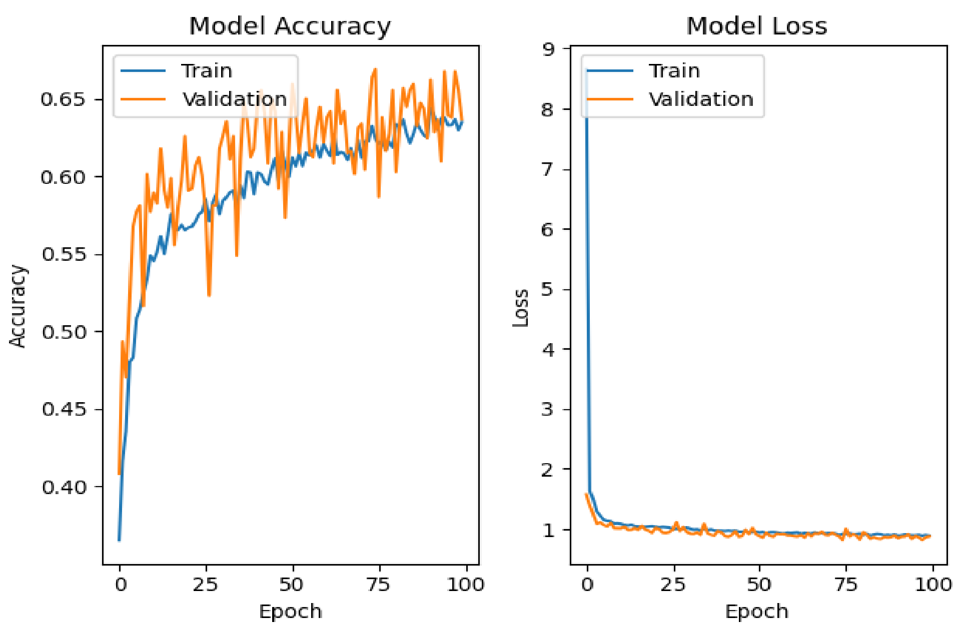



Figure 4. 8 Train and validation accuracy -loss graph for InceptionV3

As shown in the above figure (4.8) the validation accuracy is higher than the train accuracy in all iterations.

The model classification report on the test dataset is represented in the following confusion matrix as shown in Figure (4.7) below

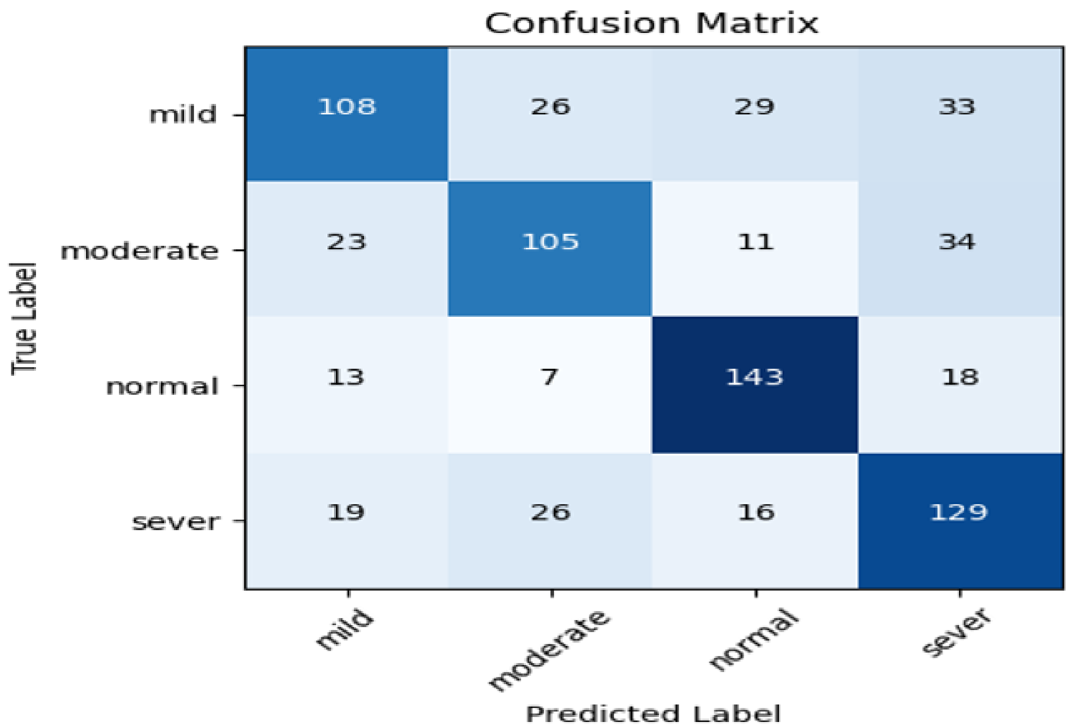



Figure 4. 9 Confusion matrix for InceptionV3 model

As the figure above illustrates from the total number of 740 test sets 485 instances were correctly classified on the correct classes while the remaining 255 instances were misclassified.

## 4.6. Error Analysis

1. Which preprocessing techniques are suitable for kidney CT hydronephrosis images?
2. Which Deep feature extraction (transfer learning) method performs best for kidney hydronephrosis image classification?
3. Which Deep learning technique model achieves the best result for kidney hydronephrosis stage classification?
4. To what extent does the developed model classify the kidney hydronephrosis stage?

**<u>Research questions are answered as described below in this study</u>**

1. Which preprocessing techniques are suitable for kidney CT hydronephrosis images?

Image datasets are enhanced with different preprocessing techniques for the model performance. Image resizing by bilinear interpolation and noise reduction by a median with Gaussian achieve good results with an experimental comparison of various techniques using image quality evaluation metrics values before and after the experiment applied to the image. The number of collected images is balanced by augmentation.

2. Which feature extraction techniques are suitable for kidney hydronephrosis stage classification?

The feature extraction technique used for the model is features extracted using pre-trained deep-based features extracted using CNN architectures (VGG16). The selection of these feature extraction techniques was based on the comparison of experiments on how deep learning pre-trained classification models are developed for kidney hydronephrosis stage classification.

3. Which deep learning pre-trained model achieves the best result for kidney hydronephrosis stage classification?

The Deep learning pre-trained model was developed.VGG16 pre-trained models achieved good result compared to other to classifiy kidney hydronephrosis CT images to.

4. To what extent does the developed model classify the kidney hydronephrosis stage?

   We evaluate the performance of the developed model using several metrics. These metrics provide a comprehensive understanding of the model's effectiveness in classifying kidney hydronephrosis stages.

**Error analysis** is an essential tool in identifying areas where the deep-learning pre-trained model may be making mistakes, thereby aiding in the improvement of overall model performance. By analyzing errors in pre-trained learning models, several advantages can be gained, including a deeper understanding of model performance, the ability to assess data quality, identification of weak base learners, guidance for model updates, and the creation of a feedback loop for iterative model improvements. The proposed deep learning model, which was developed using both VGG16, ResNet50, and InceptionV3 pre-trained models, exhibited promising performance. The evaluation results for the pre-trained models model, using specific hyper-parameters such as a learning rate of 0.001, the Adam optimizer, ReLu as the activation function in the hidden layers, softmax as the output layer activation function, a dropout value of 0.5, and 32 batch size on 50 epochs, are provided below.

Table 4. 6 Performance of each model

| Performance of VGG16 model | |
|---|---|
| Traning accuracy:-91% | Traning loss:-22% |
| Validation accuracy:-86% | Validation loss:-48% |
| Test accuracy:-83% | Test loss:-57% |
| **Performace of ResNet50 model** | |
| Traning accuracy:-79% | Traning loss:-28% |
| Validation accuracy:-83% | Validation loss:-32% |
| Test accuracy:-81% | Test loss:-44% |
| **Performance of InceptionV3 model** | |
| Training accuracy:-63.38% | Training loss:-88.91% |
| Validation accuracy:-64.86% | Validation loss:-84.22% |
| Test accuracy:-64.20% | Test loss:-86.79% |

The VGG16 pre-trained classification model achieved high accuracy, correctly classifying 5458 instances of training images, 643 validation images, and 648 testing images. However, the model misclassifies 462 images in the training dataset, 97 images in the validation dataset, and 92 images in the testing dataset.

The majority of errors and misclassifications in the training, validation, and test data were due to mild classes as shown in Figure 4.5. The model often mistook mild images as moderate and normal images. Another error occurred with severe classes, where severe images were learned as mild, and moderate. To improve the model's accuracy and robustness, the researcher suggested correcting images that look like other classes and finding additional feature extraction techniques that can differentiate them. Additionally, the InceptionV3 pre-trained model exhibited weaker classification ability compared to VGG16 and ResNet50.

## 4.7. Discussion

This kidney hydronephrosis stage classification study uses three CNN pre-trained models which are VGG16, InceptionV3, and ResNet50. For all models, we have achieved better results on a

10%-10%-80% ratio with 32 bach size and 50 itration. The image dataset suffers from salt and paper noise and they are different sizes during collection and further preprocessing is needed. To reduce the noise, we have used four filtering techniques are used. From these due to the nature of the image and image quality evaluation metrics results in median with Gaussian and bicubic interpolation techniques are used for filtering and resizing respectively. features were extracted using pre-trained models( VGG16, ResNet50, and InceptionV3).In the developed model VGG16 learns some of the severe and mild image instances as moderate and normal images but not as a class of them. The model achieved 91%, 83%, and 86% accuracy for training, validation, and testing respectively. The classification report represented in Figure (4.4) validation and accuracy graph separate speed as the iteration increases which suggests that the model is not generalizing as effectively to new data. The second pre-trained model(ResNet50) learns some mild images as moderate and normal, and severe as moderate. ResNet50 achieved 79%, 73%, and 81% accuracy for training, validation, and testing respectively. The third model(InceptionV3) pre-trained model learns some of the mild hydronephrosis images as severe, moderate, and normal. InceptionV3 achieved 63.46% train accuracy, 63.65% validation accuracy, and 65.54% test accuracy. According to the classification report, the VGG16 model is the best one selected for our data set. According to the classification report, the VGG16 pre-trained model is the best one and selected for our data set.

# CHAPTER FIVE

# CONCLUSION AND RECOMMENDATION

## 5.1. Conclusion

In conclusion, this study aimed to develop a model for classifying kidney hydronephrosis stages using deep learning and image processing techniques. For classification model datasets were collected from Dilla University General Hospital, Selam General Hospital, and a free online dataset repository(Kaggle.com). Dataset labeling was performed by radiologist domain experts. Basic preliminary image processing techniques were applied to the CT image datasets. Each preprocessing techniques were evaluated based on evaluation metrics and the best technique was selected for model development.

The models employed in this research included VGG16, ResNet50, and InceptionV3, which are widely recognized for their capabilities in image classification tasks. Through extensive experimentation and evaluation, it was observed that VGG16 consistently outperformed ResNet50 and InceptionV3 in terms of accuracy This superior performance can be attributed to VGG16's deep architecture with sequential connection pattern is common in traditional convolutional neural network (CNN) architectures, where each layer processes the input data step by step, gradually extracting higher-level features within the kidney images associated with different hydronephrosis stages.

Moreover, the integration of image processing techniques such as preprocessing, feature extraction, and segmentation contributed significantly to enhancing the models' performance and interpretability.

The results obtained in this study demonstrate the potential of deep learning and image processing approaches in medical image analysis, particularly in the context of kidney hydronephrosis stage classification. Future research directions may involve exploring ensemble methods and incorporating advanced deep-learning architectures to further improve classification accuracy and generalization across diverse datasets.

## 5.2. Contribution

The study has made significant contributions to the field of kidney stone analysis, specifically in the areas of hydronephrosis stage classifications and data preprocessing techniques. The contributions of this study are summarized as follows:

1. We have contributed a total of 7400 CT kidney hydronephrosis images with varying degrees of severity. This dataset, consisting of annotated images provides researchers with a valuable resource for training and validating their detection algorithms.
2. We have contributed preprocessing techniques that have proven to be highly effective in improving the quality and reliability of kidney hydronephrosis data. In the preprocessing phase, techniques are chosen based on image quality evaluation metrics value to assess the impact of preprocessing techniques.
3. By sharing our dataset and expertise in preprocessing techniques, we have facilitated the development of more accurate and comprehensive classification algorithms and have contributed to the broader understanding of kidney hydronephrosis. The potential implications of our work extend to healthcare professionals, enabling them to make more informed decisions in classification and treating kidney hydronephrosis patients. We remain committed to furthering research and are eager to collaborate with fellow researchers in advancing our collective understanding of this prevalent condition.

## 5.3. Recommendation

The proposed kidney stone hydronephrosis stage classification holds potential for stage classification using a CT image dataset. However, several aspects require further investigation and improvement of the developed model. This section highlights key points that need further investigation. According to the additional investigation, we have suggested the following key points.

Firstly it is recommended to use an additional dataset as the model is a deep learning-based model that needs a large dataset for model robustness. In a second way, we recommended that future researchers ensemble different pre-trained models for accurate classification. In the third recommendation, future investigators can build a hydronephrosis diagnosis model using the collected CT images. In the last, we suggest that develop a machine learning or deep learning model for recommending health care and medical information to the patient.

## REFERENCES


[1] et al Yanping Li, “"The Impact of Healthy Lifestyle Factors on Life Expectancies in the Population,” 2018.

[2] E. G. Onal and H. Tekgul, “Assessing kidney stone composition using smartphone microscopy and deep neural networks,” *BJUI Compass*, vol. 3, no. 4, pp. 310–315, 2022, doi: 10.1002/bco2.137.

[3] F. Lopez *et al.*, “Assessing deep learning methods for the identification of kidney stones in endoscopic images,” Mar. 2021, doi: 10.1109/EMBC46164.2021.9630211.

[4] H. Sibai, J. L. Pippi Salle, A. M. Houle, and R. Lambert, “Hydronephrosis with diffuse or segmental cortical thinning: Impact on renal function,” *J. Urol.*, vol. 165, no. 6 II SUPPL., pp. 2293–2295, 2001, doi: 10.1016/s0022-5347(05)66187-3.

[5] M. Genemo, “Kidney Stone Detection and Classification Based On Deep Learning Approach,” *Int. J. Adv. Nat. Sci. Eng. Res.*, vol. 4, no. 4, pp. 38–42, 2023, doi: 10.59287/ijanser.2023.7.4.545.

[6] A. Khan, R. Das, and M. C. Parameshwara, “Detection of kidney stone using digital image processing: a holistic approach,” *Eng. Res. Express*, vol. 4, no. 3, Sep. 2022, doi: 10.1088/2631-8695/ac8b65.

[7] B. L. Iverson and P. B. Dervan, “kidney-ureteral-stones.” [Online]. Available: https://www.beaumont.org/conditions/kidney-ureteral-stones-causes#:~:text=People with a family history,increase risk of stone disease.

[8] H. M. School., “Artificial Intelligence in Medicine,” *Harvard Heal. Publ.*, 2020, [Online]. Available: https://hms.harvard.edu/news/artificial-intelligence-medicine

[9] P. Joseph, R., & Mukherjee, “AI in Chronic Kidney Disease,” *J. Med. Internet Res.*, vol. 22, 2020, doi: 10.2196/11279.

[10] B. L. Iverson and P. B. Dervan, “hydronephrosis.” [Online]. Available: https://www.kidney.org/atoz/content/hydronephrosis#:~:text=This swelling typically

happens when,leading to possible kidney damage.

[11] “structure of kidney.” [Online]. Available: https://www.google.com/url?sa=i&url=https%3A%2F%2Fcourses.lumenlearning.com%2Fwm-biology2%2Fchapter%2Fkidneys%2F&psig=AOvVaw0U9IDIdHZhr1qHYi1of6Ro&ust=1710311685545000&source=images&cd=vfe&opi=89978449&ved=0CBUQjhxqFwoTCPiB3Y-N7oQDFQAAAAAdAAAAABAD

[12] K. M. Black, H. Law, A. Aldoukhi, J. Deng, and K. R. Ghani, “Deep learning computer vision algorithm for detecting kidney stone composition,” pp. 920–924, 2020, doi: 10.1111/bju.15035.

[13] A. Soni, “Kidney Stone Recognition and Extraction using Directional Emboss & SVM from Computed Tomography Images,” pp. 57–62, 2020.

[14] “National library of medcine.” [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8226024/

[15] D. A. Debal and T. M. Sitote, “Chronic kidney disease prediction using machine learning techniques,” *J. Big Data*, vol. 9, no. 1, pp. 137–140, 2022, doi: 10.1186/s40537-022-00657-5.

[16] S. Navaratnam, S. Fazilah, V. Raman, and S. Perumal, “Automated Kidney Stone Segmentation by Seed Pixel Region Growing Approach : Initial Implementation and Results,” vol. 7, pp. 43–48, 2018.

[17] B. L. Iverson and P. B. Dervan, “kidney and bladder,” pp. 7823–7830, [Online]. Available: https://www.hopkinsmedicine.org/health/wellness-and-prevention/anatomy-of-the-urinary-system#:~:text=Two ureters.,a kidney infection can develop.

[18] A. Mart, S. Member, D. Trinh, J. El Beze, and J. Hubert, “Towards an automated classification method for ureteroscopic kidney stone images using ensemble learning *,” pp. 1936–1939, 2020.

[19] B. L. Iverson and P. B. Dervan, “watershed segmentation.” [Online]. Available:

https://docs.opencv.org/3.4/d3/db4/tutorial_py_watershed.html

[20] K. K. Patro *et al.*, “Application of Kronecker convolutions in deep learning technique for automated detection of kidney stones with coronal CT images,” *Inf. Sci. (Ny).*, vol. 640, no. April, p. 119005, 2023, doi: 10.1016/j.ins.2023.119005.

[21] Karthikrajan Senthilnathan and A. K. Iyswarya, “Arti fi cial Neural Network Control Strategy for Multi-converter Uni fi ed Power Quality Conditioner for Power Quality Improvements in 3-Feeder System,” *Adv. Intell. Syst. Comput.*, vol. 394, pp. 1105–1111, 2016, doi: 10.1007/978-81-322-2656-7.

[22] M. Mitterberger, F. Aigner, L. Pallwein, G. Pinggera, P. Rehder, and F. Frauscher, “Sonographic Detection of Renal and Ureteral Stones . Value of the Twinkling Sign,” vol. 35, no. 5, pp. 532–541, 2009.

[23] B. L. Iverson and P. B. Dervan, “hhydronephrosis stage.” [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK563217/

[24] D. E. C. Na and C. Hipertensiva, “national library of medcine on large stone kidney disease,” *Natl. Libr. medcine*, [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK560887/#:~:text=Renal extracorporeal lithotripsy is a,that is truly non-invasive.

[25] C. D. Scales, A. C. Smith, J. M. Hanley, and C. S. Saigal, “scales prevalence kidney stones US 2012,” *Eur. Urol.*, vol. 62, pp. 160–165, 2012.

[26] I. Technology, “Brain Tumor Detection and Classification Using Symmetrical Side Analysis and Thresholding Technique,” 2019.

[27] M. Council, B. X-rays, R. This, and F. This, “X-Rays – Benefits and Risks Techniques that use x-rays,” *America (NY).*, no. May, 2007.

[28] B. L. Iverson and P. B. Dervan, “ultrasound,” pp. 7823–7830, [Online]. Available: https://www.google.com/search?q=Ultrasound+scan%2C+also+known+as+sonography%2C+is+a+non-invasive+medical+imaging+technique+that+utilizes+high-frequency+sound+waves+to+create+real-

time+images+of+internal+body+structures&rlz=1C1BNSD_enET1063ET1064&oq=Ultras

[29] A. Qaseem, P. Dallas, M. A. Forciea, M. Starkey, and T. D. Denberg, “Dietary and pharmacologic management to prevent recurrent nephrolithiasis in adults: A clinical practice guideline from the American College of Physicians,” *Ann. Intern. Med.*, vol. 161, no. 9, pp. 659–667, 2014, doi: 10.7326/M13-2908.

[30] B. L. Iverson and P. B. Dervan, “What-is-image-acquisition-.” [Online]. Available: https://www.quora.com/What-is-image-acquisition-and-its-techniques#:~:text=Image acquisition is like taking,image using software and devices.

[31] N. B. Bahadure, A. K. Ray, and H. P. Thethi, “Image Analysis for MRI Based Brain Tumor Detection and Feature Extraction Using Biologically Inspired BWT and SVM,” *Int. J. Biomed. Imaging*, vol. 2017, 2017, doi: 10.1155/2017/9749108.

[32] N. Khandelwal, R. Kamble, S. Rani, and S. Mamidwar, “KIDNEY STONE DETECTION USING IMAGE PROCESSING AND DEEP LEARNING,” JETIR, 2023. [Online]. Available: www.jetir.org

[33] “Image denoising techniques.” [Online]. Available: https://vciba.springeropen.com/articles/10.1186/s42492-019-0016-7

[34] Y. J. Dhanachandra, N., & Chanu, “An image segmentation approach based on fuzzy c-means and dynamic particle swarm optimization algorithm. Multimedia Tools and Applications,” pp. 79(25–26), 2020, doi: https://doi.org/10.1007/s11042-020-08699-8.

[35] " Pratt, W.K.John Wiley & Sons, “[Digital Image Processing." .],” 2007.

[36] R. E. P. Gonzalez, R.C., and Woods, .“[ ‘Digital Image Processing.,’” 2017.

[37] Lucchese, L., and Mitra, S.K. “.” Cengage Learning, “Understanding Digital Signal Processing,” 2019.

[38] J. M. A., Coll, B., and Morel, “A Non-Local Algorithm for Image Denoising,” *IEEE Comput. Soc.*, 2005.

[39] D. E. C. Na and C. Hipertensiva, “CNN requires fixed size of data”, [Online]. Available: https://datascience.stackexchange.com/questions/64022/why-must-a-cnn-have-a-fixed-input-size

[40] T. M. Shorten, C., & Khoshgoftaar, “Data Augmentation for Deep Learning,” *J. Big Data,* 2019.

[41] C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *J. Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0197-0.

[42] B. L. Iverson and P. B. Dervan, “image-segmentation.” [Online]. Available: https://www.mathworks.com/discovery/image-segmentation.html

[43] “No Title.” [Online]. Available: https://encord.com/blog/image-segmentation-for-computer-vision-best-practice-guide/

[44] T. (). Ronneberger, O., Fischer, P., and Brox, “Convolutional Networks for Biomedical Image Segmentation,” . *Med. Image Comput. Comput. Interv. (MICCAI*, 2015.

[45] “u-net image segmentation architecture.” [Online]. Available: https://www.google.com/url?sa=i&url=https%3A%2F%2Flmb.informatik.uni-freiburg.de%2Fpeople%2Fronneber%2Fu-net%2F&psig=AOvVaw3_AueQIJ5IBSmUt-1AeU3S&ust=1714469363449000&source=images&cd=vfe&opi=89978449&ved=0CBQQjhxqFwoTCPCJ6LiN54UDFQAAAAAdAAAAABAE

[46] R. E. Gonzalez, R. C., & Woods, “Digital Image Processing].,” vol. (4th ed, 2018.

[47] A. A. Irudayaraj and C. Hava Muntean, “Kidney Stone Detection using Deep Learning Methodologies MSc Research Project MSc in Data Analytics.”

[48] S. Khan, H. Rahmani, S. A. A. Shah, and M. Bennamoun, “A Guide to Convolutional Neural Networks for Computer Vision,” *Synth. Lect. Comput. Vis.*, vol. 8, no. 1, pp. 1–207, 2018, doi: 10.2200/s00822ed1v01y201712cov015.

[49]: Efrem Yohannes, “Draft_manuscript_Applied Sciences Format_November_18.”

[50] A. Mekonnen, “Addis Ababa Science and Technology University Brain Tumor

Classification and Segmentation Model Using Deep Learning Approach,” 2020.

[51] “Relu activation function.” [Online]. Available: https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.nomidl.com%2Fdeep-learning%2Fwhat-is-relu-and-sigmoid-activation-function%2F&psig=AOvVaw1i1TfJizUD2GSMI8NAwdtV&ust=1714467941829000&source=images&cd=vfe&opi=89978449&ved=0CBQQjhxqFwoTCMC5ppSI54UDFQAAAA

[52] “Sigmoid activation function.” [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/sigmoid-function

[53] “sigmoid.”

[54] “poling layer.” [Online]. Available: https://www.quora.com/What-is-max-pooling-in-convolutional-neural-networks

[55] “flattening.” [Online]. Available: https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.superdatascience.com%2Fblogs%2Fconvolutional-neural-networks-cnn-step-3-flattening&psig=AOvVaw3cFNx-hxZP2c-PGLjyWa0d&ust=1709796374321000&source=images&cd=vfe&opi=89978449&ved=0CBUQjhxqFwoTCMCX9LaN34QD

[56] B. L. Iverson and P. B. Dervan, “Feature selection.” [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S0925231218302911#:~:text=Feature selection is referred to,and irrelevant features are removed.

[57] B. D. Vijay Kotu, “Deep learning 简介 一 、 什么是 Deep Learning ？,” *Nature*, vol. 29, no. 7553, pp. 1–73, 2019, [Online]. Available: http://deeplearning.net/

[58] A. Caglayan, M. O. Horsanali, K. Kocadurdu, E. Ismailoglu, and S. Guneyli, “Deep learning model-assisted detection of kidney stones on computed tomography,” *Int. Braz J Urol*, vol. 48, no. 5, pp. 830–839, 2022, doi: 10.1590/S1677-5538.IBJU.2022.0132.

[59] J. S. Kaiming He, Xiangyu Zhang, Shaoqing Ren, “Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).,” *IEEE*, 2016.

[60] “Resnet50.” [Online]. Available: https://www.google.com/url?sa=i&url=https%3A%2F%2Ftowardsdatascience.com%2Fthe-annotated-resnet-50-a6c536034758&psig=AOvVaw26EfR5yLCxps_dC0rGJxn-&ust=1710253848418000&source=images&cd=vfe&opi=89978449&ved=0CBUQjhxqFwoTCIjRr9G17IQDFQAAAAAdAAAAABAD

[61] M. Längkvist, J. Jendeberg, P. Thunberg, A. Loutfi, and M. Lidén, “Computer aided detection of ureteral stones in thin slice computed tomography volumes using Convolutional Neural Networks,” *Comput. Biol. Med.*, vol. 97, no. April, pp. 153–160, 2018, doi: 10.1016/j.compbiomed.2018.04.021.

[62] K. Yildirim, P. G. Bozdag, M. Talo, O. Yildirim, M. Karabatak, and U. R. Acharya, “Deep learning model for automated kidney stone detection using coronal CT images,” *Comput. Biol. Med.*, vol. 135, no. June, p. 104569, 2021, doi: 10.1016/j.compbiomed.2021.104569.

[63] A. Nithya, A. Appathurai, N. Venkatadri, D. R. Ramji, and C. Anna Palagan, “Kidney disease detection and segmentation using artificial neural network and multi-kernel k-means clustering for ultrasound images,” *Meas. J. Int. Meas. Confed.*, vol. 149, p. 106952, 2020, doi: 10.1016/j.measurement.2019.106952.

[64] B. L. Iverson and P. B. Dervan, “median-filte.” [Online]. Available: https://www.sciencedirect.com/topics/computer-science/median-filter#:~:text=Median filters are useful in,a window over the image.

[65] K. Q. Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, “Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,” *IEEE*, pp. 4700–4708, 2017.

[66] A. Goodfellow, I., Bengio, Y., & Courville, “Deep Learning,” *IEEE*, p. CHAPTER 9, 2016.

[67] A. Simonyan, K., & Zisserman, “Very deep convolutional networks for large-scale image

recognition.,” 2015.

[68] D. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., & Anguelov, “Going deeper with convolutions,” *IEE*, pp. 1–9, 2015.

[69] A. Simonyan, K., & Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014.

[70] Z. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, “Rethinking the Inception Architecture for Computer Vision,” *IEEE*, pp. 2818–2826, 2016.

[71] Q. Pan, S. J., & Yang, “A Survey on Transfer Learning,” *IEEE*, 2010, doi: doi: 10.1109/TKDE.2009.191.

[72] D. E. C. Na and C. Hipertensiva, “tensorflow.” [Online]. Available: https://www.tensorflow.org/guide/basics

[73] “Confusion matrix.” [Online]. Available: https://www.google.com/url?sa=i&url=https%3A%2F%2Ftowardsdatascience.com%2Funderstanding-confusion-matrix-a9ad42dcfd62&psig=AOvVaw16TJtm0izHR-4KOt2ytCuR&ust=1713509384973000&source=images&cd=vfe&opi=89978449&ved=0CBQQjhxqFwoTCOCa16uVy4UDFQAAAAAdAAAAABAE

**APPENDIX I** Customized VGG16 model Python code for hydronephrosis stage classification

```
import os
import cv2
import numpy as np
from sklearn.model_selection import train_test_split
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
from keras.preprocessing.image import ImageDataGenerator
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, classification_report
# Constants
image_directory = "/content/drive/MyDrive/80 by 20 final dataset"
classes = ["mild", "moderate", "normal", "sever"]
num_classes = len(classes)
img_size = 224  # VGG16 input size
# Load and preprocess the images
def load_images():
    images = []
    labels = []
    for class_label, class_name in enumerate(classes):
        path = os.path.join(image_directory, class_name)
        for image_name in os.listdir(path):
            image_path = os.path.join(path, image_name)
            image = cv2.imread(image_path)
            image = cv2.resize(image, (img_size, img_size))
            images.append(image)
```

```
        labels.append(class_label)
    return np.array(images), np.array(labels)


# Split the data into training, validation, and testing sets
def split_data(images, labels):
    X_train, X_temp, y_train, y_temp = train_test_split(images, labels, test_size=0.2, random_state=42)
    X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5, random_state=42)
    return X_train, X_val, X_test, y_train, y_val, y_test
# Build the VGG16 model
def build_model():
    model = Sequential()
    model.add(Conv2D(64, (3, 3), activation='relu', padding='same', input_shape=(img_size, img_size, 3)))
    model.add(Conv2D(64, (3, 3), activation='relu', padding='same'))
    model.add(MaxPooling2D((2, 2)))
    model.add(Conv2D(128, (3, 3), activation='relu', padding='same'))
    model.add(Conv2D(128, (3, 3), activation='relu', padding='same'))
    model.add(MaxPooling2D((2, 2)))
    model.add(Conv2D(256, (3, 3), activation='relu', padding='same'))
    model.add(Conv2D(256, (3, 3), activation='relu', padding='same'))
    model.add(Conv2D(256, (3, 3), activation='relu', padding='same'))
    model.add(MaxPooling2D((2, 2)))
    model.add(Conv2D(512, (3, 3), activation='relu', padding='same'))
    model.add(Conv2D(512, (3, 3), activation='relu', padding='same'))
    model.add(Conv2D(512, (3, 3), activation='relu', padding='same'))
    model.add(MaxPooling2D((2, 2)))
```

```
    model.add(Conv2D(512, (3, 3), activation='relu', padding='same'))
    model.add(Conv2D(512, (3, 3), activation='relu', padding='same'))
    model.add(Conv2D(512, (3, 3), activation='relu', padding='same'))
    model.add(MaxPooling2D((2, 2)))
    model.add(Flatten())
    model.add(Dense(1024, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(1024, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(num_classes, activation='softmax'))
    model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
    # Display model summary
    model.summary()
    return model
# Train the model with data augmentation
def train_model(model, X_train, X_val, y_train, y_val):
    train_datagen = ImageDataGenerator(
        rotation_range=20,
        width_shift_range=0.2,
        height_shift_range=0.2,
        shear_range=0.2,
        zoom_range=0.2,
        horizontal_flip=True,
        fill_mode='nearest'
    )
    val_datagen = ImageDataGenerator()  # Validation data generator without augmentation
    train_generator = train_datagen.flow(X_train, y_train, batch_size=32)
```

```
    val_generator = val_datagen.flow(X_val, y_val, batch_size=32)
    history = model.fit(train_generator, epochs=100, validation_data=val_generator, verbose=1)
    return history
# Evaluate the model
def evaluate_model(model, X_test, y_test):
    loss, accuracy = model.evaluate(X_test, y_test, verbose=1)
    print("Test Loss:", loss)
    print("Test Accuracy:", accuracy)
    y_pred = np.argmax(model.predict(X_test), axis=-1)
    # Plot confusion matrix
    plot_confusion_matrix(y_test, y_pred)
    report = classification_report(y_test, y_pred, target_names=classes)
    print("Classification Report:")
    print(report)
# Plot the confusion matrix
def plot_confusion_matrix(y_true, y_pred):
    cm = confusion_matrix(y_true, y_pred)
    plt.imshow(cm, interpolation='nearest', cmap=plt.cm.Blues)
    plt.title('Confusion Matrix')
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)
    plt.xlabel('Predicted Label')
    plt.ylabel('True Label')
    thresh = cm.max() / 2.0
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
```

```
        plt.text(j, i, format(cm[i, j], 'd'), horizontalalignment="center", color="white" if cm[i, j] > thresh else "black")

    plt.tight_layout()

    plt.show()

# Main function

def main():

    # Load and preprocess the images

    images, labels = load_images()

    # Split the data into training, validation, and testing sets

    X_train, X_val, X_test, y_train, y_val, y_test = split_data(images, labels)

    # Build the model

    model = build_model()

    # Train the model

    history = train_model(model, X_train, X_val, y_train, y_val)

    # Evaluate the model

    evaluate_model(model, X_test, y_test)

if __name__ == '__main__':

    main()
```

**APPENDIX vv Customized ResNet50 model Python code for hydronephrosis stage classification.**

```
import os

import cv2

import numpy as np

from sklearn.model_selection import train_test_split

from keras.models import Model

from keras.layers import Input, Conv2D, MaxPooling2D, GlobalAveragePooling2D, BatchNormalization, Activation, Dense

from keras.optimizers import Adam
```

```
from keras.callbacks import ModelCheckpoint, EarlyStopping
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, classification_report
# Constants
image_directory = "/path/to/dataset"
classes = ["mild", "moderate", "normal", "sever"]
num_classes = len(classes)
img_size = 224  # ResNet50 input size
# Load and preprocess the images
def load_images():
    images = []
    labels = []
    for class_label, class_name in enumerate(classes):
        path = os.path.join(image_directory, class_name)
        for image_name in os.listdir(path):
            image_path = os.path.join(path, image_name)
            image = cv2.imread(image_path)
            image = cv2.resize(image, (img_size, img_size))
            images.append(image)
            labels.append(class_label)
    return np.array(images), np.array(labels)
# Split the data into training, validation, and testing sets
def split_data(images, labels):
    X_train, X_temp, y_train, y_temp = train_test_split(images, labels, test_size=0.2,
random_state=42)
    X_val, X_test, y_val, y_test = train_test_split(X_temp, y_temp, test_size=0.5,
random_state=42)
    return X_train, X_val, X_test, y_train, y_val, y_test
```

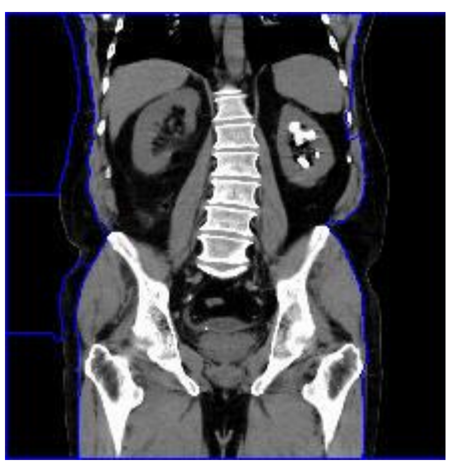
```
# Build the customized ResNet50 model
def build_model():
    input_shape = (img_size, img_size, 3)
    input_tensor = Input(shape=input_shape)
    # Initial Convolutional Layer
    x = Conv2D(64, (7, 7), strides=(2, 2), padding='same')(input_tensor)
    x = BatchNormalization()(x)
    x = Activation('relu')(x)
    x = MaxPooling2D((3, 3), strides=(2, 2), padding='same')(x)
    # Residual Blocks
    x = residual_block(x, 64)
    x = residual_block(x, 128)
    x = residual_block(x, 256)
    # Global Average Pooling
    x = GlobalAveragePooling2D()(x)
    # Dense Layers
    x = Dense(512, activation='relu')(x)
    x = BatchNormalization()(x)
    x = Dense(num_classes, activation='softmax')(x)
    model = Model(inputs=input_tensor, outputs=x)
    model.compile(optimizer=Adam(lr=0.001),                  loss='sparse_categorical_crossentropy',
metrics=['accuracy'])
    model.summary()
    return model
# Residual Block
def residual_block(input_layer, filters, strides=(1, 1)):
    x = Conv2D(filters, (3, 3), strides=strides, padding='same')(input_layer)
    x = BatchNormalization()(x)
```

```
    x = Activation('relu')(x)
    x = Conv2D(filters, (3, 3), padding='same')(x)
    x = BatchNormalization()(x)
    # Shortcut Connection
    shortcut = Conv2D(filters, (1, 1), strides=strides, padding='same')(input_layer)
    shortcut = BatchNormalization()(shortcut)
    x = keras.layers.add([x, shortcut])
    x = Activation('relu')(x)
    return x
# Train the model
def train_model(model, X_train, X_val, y_train, y_val):
    checkpoint = ModelCheckpoint('resnet50_custom.h5', monitor='val_loss', save_best_only=True, verbose=1)
    early_stop = EarlyStopping(monitor='val_loss', patience=10, restore_best_weights=True)
    history = model.fit(X_train, y_train, epochs=100, batch_size=32, validation_data=(X_val, y_val), callbacks=[checkpoint, early_stop])
    return history
# Evaluate the model
def evaluate_model(model, X_test, y_test):
    loss, accuracy = model.evaluate(X_test, y_test, verbose=1)
    print("Test Loss:", loss)
    print("Test Accuracy:", accuracy)
    y_pred = np.argmax(model.predict(X_test), axis=-1)
    # Plot confusion matrix
    plot_confusion_matrix(y_test, y_pred)
    report = classification_report(y_test, y_pred, target_names=classes)
    print("Classification Report:")
    print(report)
```

```
# Plot the confusion matrix
def plot_confusion_matrix(y_true, y_pred):
    cm = confusion_matrix(y_true, y_pred)
    plt.imshow(cm, interpolation='nearest', cmap=plt.cm.Blues)
    plt.title('Confusion Matrix')
    plt.colorbar()
    tick_marks = np.arange(len(classes))
    plt.xticks(tick_marks, classes, rotation=45)
    plt.yticks(tick_marks, classes)
    plt.xlabel('Predicted Label')
    plt.ylabel('True Label')
    thresh = cm.max() / 2.0
    for i, j in itertools.product(range(cm.shape[0]), range(cm.shape[1])):
        plt.text(j, i, format(cm[i, j], 'd'), horizontalalignment="center", color="white" if cm[i, j] >
thresh else "black")
    plt.tight_layout()
    plt.show()
# Main function
def main():
    # Load and preprocess the images
    images, labels = load_images()
    # Split the data into training, validation, and testing sets
    X_train, X_val, X_test, y_train, y_val, y_test = split_data(images, labels)
    # Build the model
    model = build_model()
    # Train the model
    history = train_model(model, X_train, X_val, y_train, y_val)
    # Evaluate the model
```

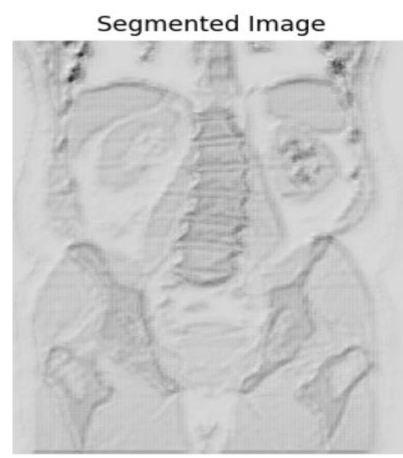
```
    evaluate_model(model, X_test, y_test)
if __name__ == '__main__':
    main()
```

**APPENDIX III** Comparison of the watershed,u-net, and threshold-based image segmentation

Segmented by watershed


Segmented by threshold




Segmented by u-net